EPOCHE & ESPRI

a DEKRA company

MS-W10-1809-I-003

Microsoft Windows Common Criteria Evaluation
Microsoft Windows 10 and Windows Server 2019
Version 1809 (October 2018 Update)
Assurance Activity Report

31-05-2019
Version 1.0

© 2019 Microsoft Corporation

© 2019 Microsoft Corporation. All rights reserved.

Prepared by:

Epoche and Espri, S.L.U.
NIF B84623016

Avenida de los Pirineos, 7
Nave 9A
28703, San Sebastián de los Reyes (Madrid)

Telephone: +34 916588314
+34 916238772

Document approved by the E&E technical manager: José Emilio Rico Martínez

© 2019 Microsoft Corporation

© 2019 Microsoft Corporation. All rights reserved.

Contents

1 Version History **4**

2 Background **5**

3 Overview **7**
3.1 Executive summary . . . 8
3.2 Referenced Documents . . . 10

4 FAU_GEN.1.1 **11**
4.1 Documentation Review activity . . . 11
4.1.1 Findings . . . 11
4.1.2 Verdict . . . 12
4.2 Test Activity . . . 13
4.2.1 Test 1 - Startup, shutdown and authentication events . . . 13
4.2.2 Test 2 - Privileges or role escalation and audit and log data access events . 19
4.2.3 Test 3 - User and group management events . . . 22
4.2.4 Test 4 - Use of privileged rights and File and object events . . . 25
4.2.5 Test 5 - Cryptographic verification of software events . . . 29
4.2.6 Test 6 - Program initiation events . . . 33
4.2.7 Test 7 - Kernel module loading events . . . 50
4.2.8 Test 8 - Lock and unlock user accounts . . . 54
4.3 Final Verdict . . . 56

5 FAU_GEN.1.2 **57**
5.1 Documentation Review activity . . . 57
5.1.1 Findings . . . 57
5.1.2 Verdict . . . 58
5.2 Test Activity . . . 58
5.2.1 Test . . . 58
5.3 Final Verdict . . . 59

6 FCS_CKM.1.1(1) **60**
6.1 Documentation Review activity . . . 62
6.1.1 Findings . . . 62
6.1.2 Verdict . . . 65
6.2 Test Activity . . . 65
6.3 Final Verdict . . . 65

7 FCS_CKM.2.1(1) **66**
7.1 Documentation Review activity . . . 68
7.1.1 Findings . . . 68
7.1.2 Verdict . . . 72

© 2019 Microsoft Corporation. All rights reserved.

7.2 Test Activity . . . 72
7.2.1 Test 1 . . . 73
7.3 Final Verdict . . . 84

8 FCS_CKM.4.1 85
8.1 Documentation Review activity . . . 87
8.1.1 Findings . . . 87
8.1.2 Verdict . . . 88
8.2 Test Activity . . . 89
8.2.1 Test 1 . . . 89
8.3 Final Verdict . . . 133

9 FCS_COP.1.1(1) 134
9.1 Documentation Review activity . . . 139
9.1.1 Findings . . . 139
9.1.2 Verdict . . . 143
9.2 Test Activity . . . 143
9.3 Final Verdict . . . 144

10 FCS_COP.1.1(2) 145
10.1 Documentation Review activity . . . 146
10.1.1 Findings . . . 146
10.1.2 Verdict . . . 149
10.2 Test Activity . . . 149
10.3 Final Verdict . . . 149

11 FCS_COP.1.1(3) 150
11.1 Documentation Review activity . . . 150
11.1.1 Findings . . . 150
11.1.2 Verdict . . . 160
11.2 Test Activity . . . 160
11.3 Final Verdict . . . 160

12 FCS_COP.1.1(4) 161
12.1 Documentation Review activity . . . 161
12.1.1 Findings . . . 161
12.1.2 Verdict . . . 164
12.2 Test Activity . . . 164
12.3 Final Verdict . . . 164

13 FCS_DTLS_EXT.1.1 165
13.1 Documentation Review activity . . . 165
13.1.1 Findings . . . 165
13.1.2 Verdict . . . 165

© 2019 Microsoft Corporation. All rights reserved.

13.2 Test Activity . . . 165
13.2.1 Test 1 . . . 169
13.3 Final Verdict . . . 174

14 FCS_DTLS_EXT.1.2 175
14.1 Documentation Review activity . . . 175
14.1.1 Findings . . . 175
14.1.2 Verdict . . . 175
14.2 Test Activity . . . 175
14.2.1 Test 1 - FCS_TLSC_EXT.1.1 for DTLS . . . 175
14.2.2 Test 2 - FCS_TLSC_EXT.1.1 for DTLS . . . 181
14.2.3 Test 3 - FCS_TLSC_EXT.1.1 for DTLS . . . 184
14.2.4 Test 4 - FCS_TLSC_EXT.1.1 for DTLS . . . 188
14.2.5 Test 5.1 - FCS_TLSC_EXT.1.1 for DTLS . . . 191
14.2.6 Test 5.2 - FCS_TLSC_EXT.1.1 for DTLS . . . 194
14.2.7 Test 5.3 - FCS_TLSC_EXT.1.1 for DTLS . . . 198
14.2.8 Test 5.4 - FCS_TLSC_EXT.1.1 for DTLS . . . 202
14.2.9 Test 5.5 - FCS_TLSC_EXT.1.1 for DTLS . . . 206
14.2.10 Test 5.6 - FCS_TLSC_EXT.1.1 for DTLS . . . 209
14.2.11 Test 1 - FCS_TLSC_EXT.1.2 for DTLS . . . 213
14.2.12 Test 2 - FCS_TLSC_EXT.1.2 for DTLS . . . 216
14.2.13 Test 3 - FCS_TLSC_EXT.1.2 for DTLS . . . 219
14.2.14 Test 4 - FCS_TLSC_EXT.1.2 for DTLS . . . 222
14.2.15 Test 5.1 - FCS_TLSC_EXT.1.2 for DTLS . . . 225
14.2.16 Test 5.2 - FCS_TLSC_EXT.1.2 for DTLS . . . 227
14.2.17 Test 5.3 - FCS_TLSC_EXT.1.2 for DTLS . . . 231
14.2.18 Test 1 - FCS_TLSC_EXT.1.3 for DTLS . . . 234
14.2.19 Test 2 - FCS_TLSC_EXT.1.3 for DTLS . . . 239
14.2.20 Test 3 - FCS_TLSC_EXT.1.3 for DTLS . . . 240
14.2.21 Test 4 - FCS_TLSC_EXT.1.3 for DTLS . . . 243
14.3 Final Verdict . . . 245

15 FCS_RBG.1.1 246
15.1 Documentation Review activity . . . 247
15.1.1 Findings . . . 247
15.1.2 Verdict . . . 249
15.2 Test Activity . . . 250
15.3 Final Verdict . . . 250

16 FCS_RBG.1.2 251
16.1 Documentation Review activity . . . 251
16.1.1 Findings . . . 251
16.1.2 Verdict . . . 253
16.2 Test Activity . . . 253

© 2019 Microsoft Corporation. All rights reserved.

16.3 Final Verdict 254

17 FCS_STO_EXT.1.1 255

17.1 Documentation Review activity 255
17.1.1 Findings 255
17.1.2 Verdict 257
17.2 Test Activity 258
17.2.1 Test 1 258
17.3 Final Verdict 262

18 FCS_TLSC_EXT.1.1 263

18.1 Documentation Review activity 264
18.1.1 Findings 264
18.1.2 Verdict 265
18.2 Test Activity 265
18.2.1 Test 1 271
18.2.2 Test 2 274
18.2.3 Test 3 278
18.2.4 Test 4 282
18.2.5 Test 5.1 285
18.2.6 Test 5.2 288
18.2.7 Test 5.3 291
18.2.8 Test 5.4 294
18.2.9 Test 5.5 297
18.2.10 Test 5.6 300
18.3 Final Verdict 303

19 FCS_TLSC_EXT.1.2 304

19.1 Documentation Review activity 305
19.1.1 Findings 305
19.1.2 Verdict 306
19.2 Test Activity 306
19.2.1 Test 1 306
19.2.2 Test 2 309
19.2.3 Test 3 312
19.2.4 Test 4 314
19.2.5 Test 5.1 317
19.2.6 Test 5.2 320
19.2.7 Test 5.3 325
19.2.8 Test 6 329
19.2.9 Test 7 329
19.3 Final Verdict 330

© 2019 Microsoft Corporation. All rights reserved.

20 FCS_TLSC_EXT.1.3 **331**
20.1 Documentation Review activity . . . 331
20.1.1 Findings . . . 331
20.1.2 Verdict . . . 331
20.2 Test Activity . . . 331
20.2.1 Test 1 . . . 332
20.2.2 Test 2 . . . 338
20.2.3 Test 3 . . . 340
20.2.4 Test 4 . . . 343
20.3 Final Verdict . . . 345

21 FCS_TLSC_EXT.2.1 **346**
21.1 Documentation Review activity . . . 346
21.1.1 Findings . . . 346
21.1.2 Verdict . . . 346
21.2 Test Activity . . . 346
21.2.1 Test 1 . . . 347
21.3 Final Verdict . . . 352

22 FCS_TLSC_EXT.3.1 **353**
22.1 Documentation Review activity . . . 353
22.1.1 Findings . . . 353
22.1.2 Verdict . . . 353
22.2 Test Activity . . . 354
22.2.1 Test 1 . . . 354
22.3 Final Verdict . . . 358

23 FCS_TLSC_EXT.4.1 **359**
23.1 Documentation Review activity . . . 359
23.1.1 Findings . . . 359
23.1.2 Verdict . . . 359
23.2 Test Activity . . . 359
23.2.1 Test 1 . . . 360
23.2.2 Test 2 . . . 363
23.3 Final Verdict . . . 367

24 FDP_ACF_EXT.1.1 **368**
24.1 Documentation Review activity . . . 369
24.1.1 Findings . . . 369
24.1.2 Verdict . . . 369
24.2 Test Activity . . . 370
24.2.1 Test 1, Test 2, Test 3, Test 4, Test 5 and Test 6 . . . 370
24.3 Final Verdict . . . 375

© 2019 Microsoft Corporation. All rights reserved.

25 FDP_IFC_EXT.1.1 **376**
25.1 Documentation Review activity . . . 376
25.1.1 Findings . . . 376
25.1.2 Verdict . . . 377
25.2 Test Activity . . . 378
25.2.1 Test 1 - Information Flow Control . . . 378
25.2.2 Verdict . . . 384
25.3 Final Verdict . . . 384

26 FIA_AFL.1.1 **385**
26.1 Documentation Review activity . . . 385
26.1.1 Findings . . . 385
26.1.2 Verdict . . . 385
26.2 Test Activity . . . 385
26.2.1 Test . . . 385
26.3 Final Verdict . . . 386

27 FIA_AFL.1.2 **387**
27.1 Documentation Review activity . . . 387
27.1.1 Findings . . . 387
27.1.2 Verdict . . . 387
27.2 Test Activity . . . 387
27.2.1 Test 1 . . . 388
27.3 Final Verdict . . . 397

28 FIA_UAU.5.1 **398**
28.1 Documentation Review activity . . . 398
28.1.1 Findings . . . 398
28.1.2 Verdict . . . 399
28.2 Test Activity . . . 399
28.2.1 Test 1 & Test 2 - Username and password . . . 399
28.2.2 Test 1 & Test 2 - Username and a PIN that releases an asymmetric key stored in OE-protected storage (TPM) . . . 403
28.2.3 Test 1 & Test 2 - Username and a PIN that releases an asymmetric key stored in OE-protected storage (Smart Card) . . . 409
28.2.4 Test 1 & Test 2 - X.509 Certificates . . . 422
28.3 Final Verdict . . . 423

29 FIA_UAU.5.2 **424**
29.1 Documentation Review activity . . . 424
29.1.1 Findings . . . 424
29.1.2 Verdict . . . 425
29.2 Test Activity . . . 425
29.2.1 Test 1 . . . 425
29.2.2 Test 2 . . . 425

© 2019 Microsoft Corporation. All rights reserved.

29.3 Final Verdict 432

30 FIA_X509_EXT.1.1 433

30.1 Documentation Review activity 434
30.1.1 Findings 434
30.1.2 Verdict 434
30.2 Test Activity 434
30.2.1 Test 1 435
30.2.2 Test 2 440
30.2.3 Test 3 442
30.2.4 Test 4 459
30.2.5 Test 5 465
30.2.6 Test 6 468
30.2.7 Test 7 471
30.3 Final Verdict 475

31 FIA_X509_EXT.1.2 476

31.1 Documentation Review activity 476
31.1.1 Findings 476
31.1.2 Verdict 476
31.2 Test Activity 476
31.2.1 Test 1 477
31.2.2 Test 2 480
31.2.3 Test 3 483
31.3 Final Verdict 486

32 FIA_X509_EXT.2.1 487

32.1 Documentation Review activity 487
32.1.1 Findings 487
32.1.2 Verdict 487
32.2 Test Activity 487
32.2.1 Test 1 488
32.3 Final Verdict 492

33 FMT_MOF_EXT.1.1 & FMT_SMF_EXT.1.1 493

33.1 Documentation Review activity 493
33.1.1 Findings 493
33.1.2 Verdict 496
33.2 Test Activity 496
33.2.1 Test 1 - Enable/Disable Screen Lock 496
33.2.2 Test 2 - Configure Screen Lock Inactivity Timeout 500
33.2.3 Test 3 - Configure Local Audit Storage Capacity 502
33.2.4 Test 4 - Configure Minimum Password Length 505
33.2.5 Test 10 - Configure Lockout Policy for Unsuccessful Authentication Attempts508
33.2.6 Test 11 - Configure Host-based Firewall 511

© 2019 Microsoft Corporation. All rights reserved.

33.2.7 Test 12 - Configure name/address of directory server to bind with 514
33.2.8 Test 14 - Configure name/address of remote management server from which to receive management settings 529
33.2.9 Test 15 - Configure Audit Rules 544
33.2.10 Test 16 - Configure Name/Address of Network Time Server 550
33.2.11 Test 17 - Enable/Disable Automatic Software Update 555
33.2.12 Test 18 - Configure Wi-Fi interface 560
33.2.13 Test 19 - Enable/Disable Bluetooth interface 562
33.2.14 Test 20 - Configure USB interfaces 564
33.2.15 Test 21 - Enable/Disable Local Area Network interface 568
33.2.16 Test 22 - Manage Windows Diagnostics settings 571
33.2.17 Test 23 - Configure Remote Connection Inactivity Timeout 575
33.3 Final Verdict 579

34 FPT_ACF_EXT.1.1 580
34.1 Documentation Review activity 580
34.1.1 Findings 580
34.1.2 Verdict 580
34.2 Test Activity 581
34.2.1 Test 1 581
34.2.2 Test 2 583
34.2.3 Test 3 585
34.2.4 Test 4 587
34.2.5 Test 5 590
34.2.6 Test 6 591
34.3 Final Verdict 592

35 FPT_ACF_EXT.1.2 593
35.1 Documentation Review activity 593
35.1.1 Findings 593
35.1.2 Verdict 593
35.2 Test Activity 593
35.2.1 Test 1 593
35.2.2 Test 2 595
35.2.3 Test 3 597
35.3 Final Verdict 597

36 FPT_ASLR_EXT.1.1 598
36.1 Documentation Review activity 598
36.1.1 Findings 598
36.1.2 Verdict 598
36.2 Test Activity 598
36.2.1 Test 1 598
36.3 Final Verdict 639

© 2019 Microsoft Corporation. All rights reserved.

37 FPT_SBOP_EXT.1.1 **640**
37.1 Documentation Review activity . . . 640
37.1.1 Findings . . . 640
37.1.2 Verdict . . . 641
37.2 Test Activity . . . 641
37.2.1 Test 1 . . . 641
37.3 Final Verdict . . . 655

38 FPT_SRP_EXT.1.1 **656**
38.1 Documentation Review activity . . . 657
38.1.1 Findings . . . 657
38.1.2 Verdict . . . 657
38.2 Test Activity . . . 657
38.2.1 Test 1 & Test 2 . . . 657
38.2.2 Test 3 & Test 4 . . . 661
38.2.3 Test 5 & Test 6 . . . 670
38.2.4 Test 7 & Test 8 . . . 688
38.3 Final Verdict . . . 692

39 FPT_TST_EXT.1.1 **693**
39.1 Documentation Review activity . . . 693
39.1.1 Findings . . . 693
39.1.2 Verdict . . . 695
39.2 Test Activity . . . 695
39.2.1 Test 1 . . . 695
39.2.2 Test 2 . . . 698
39.2.3 Test 3 . . . 705
39.3 Final Verdict . . . 714

40 FPT_TUD_EXT.1.1 **715**
40.1 Documentation Review activity . . . 715
40.1.1 Findings . . . 715
40.1.2 Verdict . . . 716
40.2 Test Activity . . . 716
40.2.1 Test 1 . . . 716
40.3 Final Verdict . . . 719

41 FPT_TUD_EXT.1.2 **720**
41.1 Documentation Review activity . . . 720
41.1.1 Findings . . . 720
41.1.2 Verdict . . . 721
41.2 Test Activity . . . 721
41.2.1 Test 1 . . . 721
41.2.2 Test 2 . . . 730
41.3 Final Verdict . . . 732

© 2019 Microsoft Corporation. All rights reserved.

42 FPT_TUD_EXT.2.1 **733**
42.1 Documentation Review activity . . . 733
42.1.1 Findings . . . 733
42.1.2 Verdict . . . 735
42.2 Test Activity . . . 735
42.2.1 Test 1 . . . 735
42.3 Final Verdict . . . 740

43 FPT_TUD_EXT.2.2 **741**
43.1 Documentation Review activity . . . 741
43.1.1 Findings . . . 741
43.1.2 Verdict . . . 742
43.2 Test Activity . . . 742
43.2.1 Test 1 . . . 742
43.2.2 Test 2 . . . 748
43.3 Final Verdict . . . 756

44 FTA_TAB.1.1 **757**
44.1 Documentation Review activity . . . 757
44.1.1 Findings . . . 757
44.1.2 Verdict . . . 759
44.2 Test Activity . . . 759
44.2.1 Test 1 . . . 759
44.3 Final Verdict . . . 762

45 FTP_ITC_EXT.1.1 (DTLS) **763**
45.1 Documentation Review activity . . . 763
45.1.1 Findings . . . 763
45.1.2 Verdict . . . 763
45.2 Test Activity . . . 763
45.2.1 Test 1 . . . 763
45.3 Final Verdict . . . 764

46 FTP_ITC_EXT.1.1 (TLS) **765**
46.1 Documentation Review activity . . . 765
46.1.1 Findings . . . 765
46.1.2 Verdict . . . 765
46.2 Test Activity . . . 765
46.2.1 Test 1 . . . 765
46.3 Final Verdict . . . 783

47 FTP_TRP.1.1 **784**
47.1 Documentation Review activity . . . 784
47.1.1 Findings . . . 784
47.1.2 Verdict . . . 785

© 2019 Microsoft Corporation. All rights reserved.

47.2 Test Activity . . . 785
47.2.1 Test 1 & Test 3 . . . 785
47.2.2 Test 2 . . . 796
47.2.3 Test 4 . . . 798
47.3 Final Verdict . . . 806

48 FAU_GEN.1/WLAN 807
48.1 Documentation Review activity . . . 808
48.1.1 Findings . . . 808
48.1.2 Verdict . . . 809
48.2 Test Activity . . . 809
48.2.1 Test 1 . . . 809
48.3 Final Verdict . . . 824

49 FCS_CKM.1/WLAN 825
49.1 Documentation Review activity . . . 826
49.1.1 Findings . . . 826
49.1.2 Verdict . . . 826
49.2 Test Activity . . . 827
49.2.1 Test 1 . . . 827
49.2.2 Test 2 . . . 829
49.3 Final Verdict . . . 838

50 FCS_CKM.2/WLAN 839
50.1 Documentation Review activity . . . 840
50.1.1 Findings . . . 840
50.1.2 Verdict . . . 840
50.2 Test Activity . . . 840
50.2.1 Test 1 . . . 840
50.3 Final Verdict . . . 843

51 FCS_TLSC_EXT.1/WLAN 844
51.1 Documentation Review activity . . . 845
51.1.1 Findings . . . 845
51.1.2 Verdict . . . 846
51.2 Test Activity . . . 846
51.2.1 Test 1 . . . 846
51.2.2 Test 2 . . . 849
51.2.3 Test 3 . . . 853
51.2.4 Test 4 . . . 856
51.2.5 Test 5 . . . 857
51.3 Final Verdict . . . 863

© 2019 Microsoft Corporation. All rights reserved.

52 FCS_TLSC_EXT.2/WLAN **864**
52.1 Documentation Review activity . . . 864
52.1.1 Findings . . . 864
52.1.2 Verdict . . . 864
52.2 Test Activity . . . 865
52.2.1 Test 1 . . . 865
52.3 Final Verdict . . . 869

53 FIA_PAE_EXT.1/WLAN **870**
53.1 Documentation Review activity . . . 870
53.1.1 Findings . . . 870
53.1.2 Verdict . . . 870
53.2 Test Activity . . . 870
53.2.1 Test 1 . . . 870
53.2.2 Test 2 . . . 876
53.2.3 Test 3 . . . 879
53.3 Final Verdict . . . 882

54 FIA_X509_EXT.2/WLAN **883**
54.1 Documentation Review activity . . . 883
54.1.1 Findings . . . 883
54.1.2 Verdict . . . 884
54.2 Test Activity . . . 884
54.2.1 Test 1 . . . 884
54.3 Final Verdict . . . 887

55 FIA_X509_EXT.4/WLAN **888**
55.1 Documentation Review activity . . . 888
55.1.1 Findings . . . 888
55.1.2 Verdict . . . 888
55.2 Test Activity . . . 888
55.2.1 Test 1 . . . 888
55.3 Final Verdict . . . 889

56 FMT_SMF_EXT.1/WLAN **890**
56.1 Documentation Review activity . . . 890
56.1.1 Findings . . . 890
56.1.2 Verdict . . . 890
56.2 Test Activity . . . 890
56.2.1 Test 1 . . . 890
56.3 Final Verdict . . . 910

57 FPT_TST_EXT.1/WLAN **911**
57.1 Documentation Review activity . . . 911
57.1.1 Findings . . . 911

© 2019 Microsoft Corporation. All rights reserved.

57.1.2 Verdict . . . 911
57.2 Test Activity . . . 912
57.2.1 Test 1 . . . 912
57.2.2 Test 2 . . . 913
57.3 Final Verdict . . . 918

58 FTA_WSE_EXT.1/WLAN 919
58.1 Documentation Review activity . . . 919
58.1.1 Findings . . . 919
58.1.2 Verdict . . . 920
58.2 Test Activity . . . 920
58.2.1 Test 1 . . . 920
58.2.2 Test 2 . . . 925
58.3 Final Verdict . . . 929

59 FTP_ITC_EXT.1/WLAN 930
59.1 Documentation Review activity . . . 930
59.1.1 Findings . . . 930
59.1.2 Verdict . . . 931
59.2 Test Activity . . . 931
59.2.1 Test 1 . . . 931
59.2.2 Test 2 . . . 932
59.2.3 Test 3 . . . 933
59.2.4 Test 4 . . . 933
59.3 Final Verdict . . . 935

60 Evaluation Tools 936

61 Acronyms 937

62 Glossary 938
62.1 Common Criteria Terms . . . 938
62.2 Technology Terms . . . 938

© 2019 Microsoft Corporation. All rights reserved.

1 Version History

Version	Date	Summary of changes
1.0	May 31, 2019	Initial version

© 2019 Microsoft Corporation. All rights reserved.

2 Background

Evaluation information:

Product	Windows Operating Systems (OS): • Microsoft Windows 10 Home Edition (October 2018 Update) (32-bit and 64-bit versions) • Microsoft Windows 10 Pro Edition (October 2018 Update) (64-bit version) • Microsoft Windows 10 Enterprise Edition (October 2018 Update) (64-bit version • Microsoft Windows Server 2019 Standard Edition, version 1809 • Microsoft Windows Server 2019 Datacenter Edition, version 1809 TOE Build: Windows 10: build 10.0.17763 (a.k.a. 1809) Windows Server 2019: build 10.0.17763 (a.k.a. 1809)
Certification Body	Centro Criptológico Nacional - CNI
Developer	1 Microsoft Way, Redmond, WA 98052, United States of America
Sponsor	1 Microsoft Way, Redmond, WA 98052, United States of America
Laboratory	Epoche & Espri S.L.U Avenida de los Pirineos, 7 Nave 9A 28703, San Sebastián de los Reyes (Madrid)
CC Version	CC v 3.1 R5
CEM Version	CEM v 3.1 R5
Protection Profile	NIAP - Protection Profile for General Purpose Operating Systems, Version: 4.2.1, 20190422, application notes as defined in [MS-W10-1903-I-000] and NIAP Technical Decisions (TDs).
EP for WLAN Client	NIAP - Extended Package Wireless Local Area Network (WLAN) Clients, Version 1.0, February 08, 2016. [EPWLAN10]
Evaluation Level	In conformance with [GPOSPP421]

© 2019 Microsoft Corporation. All rights reserved.

Security Target	Microsoft Windows 10, Windows Server 2019 (October 2018 Update) Security Target version 0.04, May 6, 2019

© 2019 Microsoft Corporation. All rights reserved.

3 Overview

This document includes the results of the evaluation activities performed by Epoche and Espri in the Common Criteria evaluation of the following operating systems:

- Microsoft Windows 10 Home Edition (October 2018 Update) (32-bit and 64-bit versions)
- Microsoft Windows 10 Pro Edition (October 2018 Update) (64-bit version)
- Microsoft Windows 10 Enterpise Edition (October 2018 Update) (64-bit version)
- Microsoft Windows Server 2019 Standard Edition, version 1809
- Microsoft Windows Server 2019 Datacenter Edition, version 1809

The build tested for all Windows 10 and Windows Server 2019 editions has been [version 10.0.17763] (also known as 1809)

The hardware platforms used during the evaluation are listed below:

- Microsoft Surface Go
- Microsoft Surface Book 2
- Microsoft Surface Pro LTE
- Microsoft Surface Laptop
- Microsoft Windows Server 2019 Hyper-V
- Microsoft Windows Server 2016 Hyper-V
- Dell Latitude 5290
- Dell Latitude 12 Rugged Tablet
- Dell PowerEdge R740
- Samsung Galaxy Book 10.6”
- Samsung Galaxy Book 12”
- HP Elite x2 1013 G3 Tablet
- HP EliteBook x360 1030 G2

Note: For the sake of a better readability the following platforms:

- Microsoft Windows Server 2016 Hyper-V
- Microsoft Windows Server 2019 Hyper-V

has been referenced to:

- Microsoft Windows Hyper-V Server 2016
- Microsoft Windows Hyper-V Server 2019

This naming convention has been used hereinafter.

The following list summarizes the combination between hardware platforms and operating system editions used for the testing:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Go with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has chosen an appropriate strategy for testing all Windows editions included in the TOE. Analysis of the differences between these OS editions shows that not all the security requirements need to be tested in all OS editions, given that the functionality is identical. Therefore, all test cases have been performed in each of Windows 10 x86, Windows 10 x64 and Windows Server 2019.

The following information is included for each SFR:

1. Description of the assurance activity, considering any modification issued by NIAP Technical Decisions.
2. Documentation review activity for both TSS and guidance including the corresponding verdict if applicable.
3. Testing activity. Composed of setup, procedure for its execution, obtained results (expected results are specified in the assurance activity description) and verdict. All this information is provided for all test cases.

3.1 Executive summary

The following table summarizes the results obtained for each security requirement defined in the ***Security Target*** document, taking into account the information provided in the ***Operational Guidance*** document:

Security Funcional Requirement - Coming from [GPOSPP421]:

SFRs	TSS Analysis	AGD Analysis	Testing Activity
FCS_CKM.1.1	PASS	PASS	Covered by FIPS certification
FCS_CKM.2.1	PASS	PASS	Covered by FIPS certification (SP 800-56A) PASS (SP 800-56B)
FCS_CKM.4.1	PASS	PASS	PASS
FCS_COP.1.1(SYM)	PASS	PASS	Covered by FIPS certification
FCS_COP.1.1(HASH)	PASS	PASS	Covered by FIPS certification
FCS_COP.1.1(SIGN)	PASS	PASS	Covered by FIPS certification

© 2019 Microsoft Corporation. All rights reserved.

SFRs	TSS Analysis	AGD Analysis	Testing Activity
FCS_COP.1.1(HMAC)	PASS	PASS	Covered by FIPS certification
FCS_RBG_EXT.1.1	PASS	PASS	Covered by FIPS certification
FCS_RBG_EXT.1.2	PASS	PASS	PASS
FCS_STO_EXT.1.1	PASS	PASS	PASS
FCS_TLSC_EXT.1.1	PASS	PASS	PASS
FCS_TLSC_EXT.1.2	PASS	PASS	PASS
FCS_TLSC_EXT.1.3	PASS	PASS	PASS
FCS_TLSC_EXT.2.1	PASS	PASS	PASS
FCS_TLSC_EXT.3.1	PASS	PASS	PASS
FCS_TLSC_EXT.4.1	PASS	PASS	PASS
FCS_DTLS_EXT.1.1	PASS	PASS	PASS
FCS_DTLS_EXT.1.2	PASS	PASS	PASS
FDP_ACF_EXT.1.1	PASS	PASS	PASS
FDP_IFC_EXT.1.1	PASS	PASS	PASS
FAU_GEN.1.1	PASS	PASS	PASS
FAU_GEN.1.2	PASS	PASS	PASS
FIA_AFL.1.1	PASS	PASS	PASS
FIA_AFL.1.2	PASS	PASS	PASS
FIA_UAU.5.1	PASS	PASS	PASS
FIA_UAU.5.2	PASS	PASS	PASS
FIA_X509_EXT.1.1	PASS	PASS	PASS
FIA_X509_EXT.1.2	PASS	PASS	PASS
FIA_X509_EXT.2.1	PASS	PASS	PASS
FMT_MOF_EXT.1.1	PASS	PASS	PASS
FMT_SMF_EXT.1.1	PASS	PASS	PASS
FPT_ACF_EXT.1.1	PASS	PASS	PASS
FPT_ACF_EXT.1.2	PASS	PASS	PASS
FPT_ASLR_EXT.1.1	PASS	PASS	PASS
FPT_SBOP_EXT.1.1	PASS	PASS	PASS
FPT_SRP_EXT.1.1	PASS	PASS	PASS
FPT_TST_EXT.1.1	PASS	PASS	PASS
FPT_TUD_EXT.1.1	PASS	PASS	PASS
FPT_TUD_EXT.1.2	PASS	PASS	PASS
FPT_TUD_EXT.2.1	PASS	PASS	PASS
FPT_TUD_EXT.2.2	PASS	PASS	PASS
FTA_TAB.1.1	PASS	PASS	PASS
FTP_ITC_EXT.1.1(TLS)	PASS	PASS	PASS
FTP_ITC_EXT.1.1(DTLS)	PASS	PASS	PASS
FTP_TRP.1.1	PASS	PASS	PASS
FTP_TRP.1.2	PASS	PASS	PASS

© 2019 Microsoft Corporation. All rights reserved.

SFRs	TSS Analysis	AGD Analysis	Testing Activity
FTP_TRP.1.3	PASS	PASS	PASS

Security Funcional Requirement - Coming from [EPWLAN10]:

SFRs	TSS Analysis	AGD Analysis	Testing Activity
FAU_GEN.1 (WLAN)	PASS	PASS	PASS
FCS_CKM.1 (WLAN)	PASS	PASS	PASS
FCS_CKM.2 (WLAN)	PASS	PASS	PASS
FCS_TLSC_EXT.1 (WLAN)	PASS	PASS	PASS
FCS_TLSC_EXT.2 (WLAN)	PASS	PASS	PASS
FIA_PAE_EXT.1	PASS	PASS	PASS
FIA_X509_EXT.2 (WLAN)	PASS	PASS	PASS
FIA_X509_EXT.4	PASS	PASS	PASS
FMT_SMF_EXT.1 (WLAN)	PASS	PASS	PASS
FPT_TST_EXT.1 (WLAN)	PASS	PASS	PASS
FTA_WSE_EXT.1	PASS	PASS	PASS
FPT_ITC_EXT.1 (WLAN)	PASS	PASS	PASS

3.2 Referenced Documents

Note: For the sake of a better readability the following documents will be referenced with their *Document Identifier* name instead of their *Document* Name:

Document Identifier	Document
Security Target	Microsoft Windows 10 and Microsoft Windows Server 2019 Security Target, version 0.04, May 6, 2019
Operational Guidance	Operational and Administrative Guidance Microsoft Windows 10 and Windows Server 2019, Common Criteria Evaluation for Microsoft Windows 10 and Windows Server 2019 Version 1809 (October 2018 Update), version 3.0. 21 December, 2018

© 2019 Microsoft Corporation. All rights reserved.

4 FAU_GEN.1.1

The assurance activity for the **FAU_GEN.1.1** requirement is stated as follows:

> The evaluator will check the administrative guide and ensure that it lists all of the auditable events. The evaluator will check to make sure that every audit event type selected in the ST is included.
>
> The evaluator will test the OS's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the ST. This should include all instance types of an event specified. When verifying the test results, the evaluator will ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

4.1 Documentation Review activity

4.1.1 Findings

The ***Security Target*** document, defines in its section **5.1.1.1 Audit Data Generation(FAU_GEN.1) and FAU_GEN.1(WLAN)**, the following auditable events:

© 2019 Microsoft Corporation. All rights reserved.

5.1.1.1 Audit Data Generation (FAU_GEN.1) and FAU_GEN.1(WLAN)

FAU_GEN.1.1 The **OS** shall be able to generate an audit record of the following auditable events:

a. Start-up and shutdown of the audit functions;
b. All auditable events for the **not specified** level of audit; and
c.
- **Authentication events (Success/Failure);**
- **Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);**
- **Privilege or role escalation events (Success/Failure);**

[
- ***File and object events (Successful and unsuccessful attempts to create, access, delete, modify, modify permissions),***
- ***User and Group management events (Successful and unsuccessful add, delete, modify, disable),***
- ***Audit and log data access events (Success/Failure),***
- ***Cryptographic verification of software (Success/Failure),***
- ***Attempted application invocation with arguments (Success/Failure e.g. due to software restriction policy),***
- ***System reboot, restart, and shutdown events (Success/Failure),***
- ***Kernel module loading and unloading events (Success/Failure),***
- ***Administrator or root-level access events (Success/Failure),***
- ***[Lock and unlock a user account, audit events from the WLAN EP listed in Table 12].***

]

In addition, the ***Operational Guidance*** document, includes in its section **7.1 Audit Events - GP OS protection profile** a table with all the auditable events generated by the TOE. The content of this table matches with the selection performed by the vendor in the ***Security Target*** document as it can be seen in the image above.

4.1.2 Verdict

The evaluator has reviewed the ***Security Target*** document and has ensured that every auditable event type selected in the ***Security Target*** document is included in the ***Operational Guidance*** document.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to this activity.

© 2019 Microsoft Corporation. All rights reserved.

4.2 Test Activity

There are quite a few auditable events. To make a better coverage, the evaluator has divided this test activity into different subtests. For each of these subtests, the evaluator has developed a script to obtain the selected audit events.

The auditable events defined in the ***Security Target*** document are the following, including the *eventID* defined in the ***Operational Guidance*** document:

1. Start-up and shutdown of the audit functions (1100, 4608)
2. Authentication events (4624, 4625)
3. Use of privileged/special rights events (4656, 4670)
4. Privilege or role escalation events (4673, 4674)
5. File and object events (4656)
6. User and Group management events (4720, 4725, 4726, 4731, 4732, 4733, 4734, 4735, 4738)
7. Audit and log data access events (4673)
8. Cryptographic verification of software (2, 3)
9. Program initiations (3077, 3038, 8020, 8022)
10. System reboot, restart and shutdown events (1100, 4608)
11. Kernel module loading and unloading events (3004, 3038)
12. Administrator or root-level access events (4624, 4625)
13. Lock and unlock a user account (4740,4767)

The table below, shows the coverage between the auditable events and the subtests division performed by the evaluator. All the auditable events selected in the ***Security Target*** document are covered:

Scripts/Events	1	2	3	4	5	6	7	8	9	10	11	12	13
Test 1 - Startup, shutdown and authentication events	✓	✓								✓		✓	
Test 2 - Privileges or role escalation and audit and log data access events				✓			✓						
Test 3 - User and Group management events						✓							
Test 4 - Use of privileged rights and File and object events			✓		✓								
Test 5 - Cryptographic verification of software events								✓					
Test 6 - Program initiation events									✓				
Test 7 - Kernel module loading events											✓		
Test 8 - Lock and unlock a user account													✓

4.2.1 Test 1 - Startup, shutdown and authentication events

Generation of the audit startup, shutdown and logon events will be tested in this test case.

4.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts. To do it, execute in a *Powershell* terminal with administrator rights the command:

© 2019 Microsoft Corporation. All rights reserved.

```
Set-ExecutionPolicy Unrestricted -Force
```

- Scripts *Test1 - Setup.ps1* and *Test1 - Startup&Shutdown&Logon.ps1* shall be available.

4.2.1.2 Procedure

The evaluator has automated this subtest by developing a script. This script shows the audit log events related to:

- Start-up and shutdown of the audit functions.
- System restart and shutdown.
- Success and failure authentication attempts.

The evaluator shall carry out the following steps to check the correct generation of the audit events:

- Run as administrator the script *Test1 - Setup.ps1*. To do this, type the following command in a *PowerShell* terminal:

  > .\Test1 - Setup.ps1

- The computer will be restarted. After that, try to log in as an administrator using a known bad password.
- Try to log in again, but this time using the correct password.
- Finally, run as administrator the script *Test1 - Startup&Shutdown&Logon.ps1*. To do this, type the following command in a *PowerShell* terminal:

  > .\Test1 - Startup&Shutdown&Logon.ps1

- Observe the results.

4.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

As it can be observed in the following images, the audit events related to the startup and shutdown of the audit functions and successful and failed authentication events have been correctly generated. In addition, the generated audit log entries include all the fields defined in the ***Security Target*** document.

- Startup and shutdown of the audit functions, and system events like shutdown, reboot or restart:

```
Message              : The event logging service has shut down.
Id                   : 1100
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 103
Opcode               : 0
Keywords             : 4620693217682128896
RecordId             : 19084
ProviderName         : Microsoft-Windows-Eventlog
ProviderId           : fc65ddd8-d6ef-4962-83d5-6e5cfe9ce148
LogName              : Security
ProcessId            : 1816
ThreadId             : 8380
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 12:57:07 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Service shutdown
KeywordsDisplayNames : {Audit Success}
Properties           : {}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Windows is starting up.

                       This event is logged when LSASS.EXE starts and the auditing subsystem is initialized.
Id                   : 4608
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12288
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 19099
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 840
ThreadId             : 844
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 12:57:28 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Security State Change
KeywordsDisplayNames : {Audit Success}
Properties           : {}
```

- Authentication events (including successful and failed authentication attempts):

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : An account failed to log on.

                       Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3E7

                       Logon Type:                      2

                       Account For Which Logon Failed:
                        Security ID:            S-1-0-0
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2

                       Failure Information:
                        Failure Reason:         Unknown user name or bad password.
                        Status:                 0xC000006D
                        Sub Status:             0xC000006A

                       Process Information:
                        Caller Process ID:      0x848
                        Caller Process Name:    C:\Windows\System32\svchost.exe

                       Network Information:
                        Workstation Name:       SURFACEBOOK2
                        Source Network Address: 127.0.0.1
                        Source Port:            0

                       Detailed Authentication Information:
                        Logon Process:          User32
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                       This event is generated when a logon request fails. It is generated on the computer where
                       access was attempted.

                       The Subject fields indicate the account on the local system which requested the logon. This is
                       most commonly a service such as the Server service, or a local process such as Winlogon.exe or
                       Services.exe.

                       The Logon Type field indicates the kind of logon that was requested. The most common types are
                       2 (interactive) and 3 (network).

                       The Process Information fields indicate which account and process on the system requested the
                       logon.

                       The Network Information fields indicate where a remote logon request originated. Workstation
                       name is not always available and may be left blank in some cases.

                       The authentication information fields provide detailed information about this specific logon
                       request.
                        - Transited services indicate which intermediate services have participated in this logon
                       request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session
                       key was requested.
Id                   : 4625
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12544
Opcode               : 0
Keywords             : -9218868437227405312
RecordId             : 19268
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 840
ThreadId             : 956
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 12:57:34 PM
ActivityId           : f0a82cfa-8c13-0005-722d-a8f0138cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Logon
KeywordsDisplayNames : {Audit Failure}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : An account was successfully logged on.

                       Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3E7

                       Logon Information:
                        Logon Type:             5
                        Restricted Admin Mode:  -
                        Virtual Account:                No
                        Elevated Token:         Yes

                       Impersonation Level:             Impersonation

                       New Logon:
                        Security ID:            S-1-5-18
                        Account Name:           SYSTEM
                        Account Domain:         NT AUTHORITY
                        Logon ID:               0x3E7
                        Linked Logon ID:                0x0
                        Network Account Name:   -
                        Network Account Domain: -
                        Logon GUID:             {00000000-0000-0000-0000-000000000000}

                       Process Information:
                        Process ID:             0x33c
                        Process Name:           C:\Windows\System32\services.exe

                       Network Information:
                        Workstation Name:       -
                        Source Network Address: -
                        Source Port:            -

                       Detailed Authentication Information:
                        Logon Process:          Advapi
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                       This event is generated when a logon session is created. It is generated on the computer that
                       was accessed.

                       The subject fields indicate the account on the local system which requested the logon. This is
                       most commonly a service such as the Server service, or a local process such as Winlogon.exe or
                       Services.exe.

                       The logon type field indicates the kind of logon that occurred. The most common types are 2
                       (interactive) and 3 (network).

                       The New Logon fields indicate the account for whom the new logon was created, i.e. the account
                       that was logged on.

                       The network fields indicate where a remote logon request originated. Workstation name is not
                       always available and may be left blank in some cases.

                       The impersonation level field indicates the extent to which a process in the logon session can
                       impersonate.

                       The authentication information fields provide detailed information about this specific logon
                       request.
                        - Logon GUID is a unique identifier that can be used to correlate this event with a KDC event.
                        - Transited services indicate which intermediate services have participated in this logon
                       request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session
                       key was requested.
Id                   : 4624
Version              : 2
Qualifiers           :
Level                : 0
Task                 : 12544
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 19355
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 840
ThreadId             : 892
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 12:57:51 PM
ActivityId           : f0a82cfa-8c13-0005-722d-a8f0138cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Logon
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

4.2.1.4 Verdict

As the result above states, the audit events related to the startup and shutdown of the audit functions and successful and failed authentication events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

4.2.2 Test 2 - Privileges or role escalation and audit and log data access events

Generation of log data access events will be tested in this test case.

4.2.2.1 Setup

The applicable setup for this test is the same as the one defined in the previous test case.

4.2.2.2 Procedure

The evaluator has automated this subtest by developing a script. This script, performs the following operations:

- First, the audit events related to the: *"Privileges or role escalation and audit and log data access events"* are enabled.
- After that, two attempts to access the audit data are carried out:
  - The first one, using a user who does not have administrator permissions.
  - The second one, using a user who has administrator permissions.

The evaluator shall carry out the following steps to check the correct generation of the audit events:

- Run as administrator the script *Test2 - AuditData&Priveleges.ps1*. To do this, type the following command in a *PowerShell* terminal:

```
.\Test2 - AuditData&Priveleges.ps1
```

- Observe the results.

4.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

During the execution of this test case, an attempt to access audit data was performed using a user who does not have administrator permissions. The result of this action is the following:

After that, the same action was repeated but this time using a user with administrator rights.

The following audit log entries show the successful and the failed attempt to access the audit data. In addition, the generated audit log entries include all the fields defined in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : An operation was attempted on a privileged object.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x5CEBE

                       Object:
                        Object Server:  LSA
                        Object Type:    -
                        Object Name:    -
                        Object Handle:  0x0

                       Process Information:
                        Process ID:     0x348
                        Process Name:   C:\Windows\System32\lsass.exe

                       Requested Operation:
                        Desired Access: 16777216
                        Privileges:             SeSecurityPrivilege
Id                   : 4674
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13056
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 19539
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 4
ThreadId             : 200
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 1:35:30 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Sensitive Privilege Use
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

```
Message              : A privileged service was called.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1006
                        Account Name:           userFAU
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x15EF16

                       Service:
                        Server: Security
                        Service Name:   -

                       Process:
                        Process ID:     0xc9c
                        Process Name:   C:\Windows\System32\svchost.exe

                       Service Request Information:
                        Privileges:             SeTcbPrivilege
Id                   : 4673
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13056
Opcode               : 0
Keywords             : -9218868437227405312
RecordId             : 19554
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 4
ThreadId             : 1916
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 1:35:36 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Sensitive Privilege Use
KeywordsDisplayNames : {Audit Failure}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

4.2.2.4 Verdict

As the result above states, the audit events related to the privileged events and audit data access events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2.**

4.2.3 Test 3 - User and group management events

Generation of user and group management audit events will be tested in this test case.

4.2.3.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A user account with user name *userFAU* shall not exist.
- A local group with name *groupFAU* shall not exist.
- Script *FAU_GEN.1 - Users&Groups.ps1* shall be available.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.3.2 Procedure

The evaluator has automated this subtest by developing a script. This script, performs the following operations:

- First, the script enables the audit function related to the accounts management events.
- After that, the following operations are executed:
  - Create a new local user account (with name *userFAU*).
  - Modify the previous user account description.
  - Disable that user account and then enable it.
  - Create a new local group (with name *groupFAU*).
  - Modify the previous group description.
  - Add the user created above to the group.
  - Remove the user from the group.
  - Delete the group *groupFAU*.
  - Delete user *userFAU*.

The evaluator shall carry out the following steps to check the generation of the audit events:

- Run as administrator the script *Test3 - Users&Groups.ps1*. To do this, type the following command in a *PowerShell* terminal:

© 2019 Microsoft Corporation. All rights reserved.

```
.\FAU_GEN.1 - Users&Groups.ps1
```

- Observe the results.

4.2.3.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The following operations have been performed during the test case execution:

- Add user
- Add group
- Add user to group
- Delete user
- Delete group
- Delete user from group
- Modify user
- Modify group
- Enable user
- Disable user

All operations have generated their related audit events. Since there are many audit log entries, the following images will show only the creation of a new user account and a new user group:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A user account was created.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x5CEBE

                       New Account:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1007
                        Account Name:           userFAU
                        Account Domain:         SURFACEBOOK2

                       Attributes:
                        SAM Account Name:       userFAU
                        Display Name:           <value not set>
                        User Principal Name:    -
                        Home Directory:         <value not set>
                        Home Drive:             <value not set>
                        Script Path:            <value not set>
                        Profile Path:           <value not set>
                        User Workstations:      <value not set>
                        Password Last Set:      <never>
                        Account Expires:                <never>
                        Primary Group ID:       513
                        Allowed To Delegate To: -
                        Old UAC Value:          0x0
                        New UAC Value:          0x15
                        User Account Control:
                                Account Disabled
                                'Password Not Required' - Enabled
                                'Normal Account' - Enabled
                        User Parameters:        <value not set>
                        SID History:            -
                        Logon Hours:            All

                       Additional Information:
                        Privileges              -
Id                   : 4720
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13824
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 19567
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 840
ThreadId             : 3216
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 1:40:45 PM
ActivityId           : f0a82cfa-8c13-0005-722d-a8f0138cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : User Account Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A security-enabled local group was created.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x5CEBE

                       New Group:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1008
                        Group Name:             groupFAU
                        Group Domain:           SURFACEBOOK2

                       Attributes:
                        SAM Account Name:       groupFAU
                        SID History:            -

                       Additional Information:
                        Privileges:             -
Id                   : 4731
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13826
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 19584
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 840
ThreadId             : 3216
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 1:40:52 PM
ActivityId           : f0a82cfa-8c13-0005-722d-a8f0138cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Security Group Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

As it can be observed in the previous images, the audit events related to the user and group management events have been correctly generated. In addition, the generated audit log entries include the fields defined in the ***Security Target*** document.

4.2.3.4 Verdict

As the result above states, the audit events related to user and group management events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3.**

4.2.4 Test 4 - Use of privileged rights and File and object events

Generation of privileged rights and File and object audit events will be tested in this test case.

4.2.4.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A user account with user name *userFAU* shall exist. This user shall belong to the default *Users* group.

© 2019 Microsoft Corporation. All rights reserved.

- A file in path *C:\TEMP\file.txt* shall not exist.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.4.2 Procedure

The evaluator has automated this subtest by developing a script. The script shows the audit functions related to the use of privileged rights and File and object events. This script, performs the following operations:

- First, a new empty file is created in a temporary folder (*C:\TEMP\file.txt*). After that, the following operations are carried out over the created file:
  - The inherited permissions are disabled.
  - Full permissions over the file are assigned to the *Administrators* group.
  - An audit rule is added to the created file to audit all access event (success and failure) performed by the *Users* group.
- After that, the audit functions related to the file system events are enabled.
- Finally, two access attempts to the file are performed. The first one, using a user who does not belong to the *Administrators* group. The second one, using a user who belongs to the *Administrators* group.

The evaluator shall carry out the following steps to check the generation of the audit events:

- Run as administrator the script *Test4 - Fileobject&AdminRoot.ps1*. To do this, type the following command in a *PowerShell* terminal:

  ```
  .\FAU_GEN.1 - Fileobject&AdminRoot.ps1
  ```

- Observe the results.

4.2.4.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

During the test execution, an access attempt to the created file is performed using a user who does not have permissions to access it. Therefore, its access is denied:

```
Windows PowerShell
WARNING: Access to the path 'C:\TEMP\file.txt' is denied.
Press Enter to continue...:
```

After that, the file was accessed by a user with the proper permissions. The following audit log entries that show both, the failed and the successful attempts to access the file, were generated:

```
EventID             : 4656
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 19622
Category            : (12800)
CategoryNumber      : 12800
EntryType           : FailureAudit
Message             : A handle to an object was requested.

                      Subject:
                         Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1009
                         Account Name:           userFAU
                         Account Domain:         SURFACEBOOK2
                         Logon ID:               0x218e56

                      Object:
                         Object Server:          Security
                         Object Type:            File
                         Object Name:            C:\TEMP\file.txt
                         Handle ID:              0x0
                         Resource Attributes:    -

                      Process Information:
                         Process ID:             0x1740
                         Process Name:           C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

                      Access Request Information:
                         Transaction ID:         {00000000-0000-0000-0000-000000000000}
                         Accesses:               %%1538
                                                 %%1541
                                                 %%4416
                                                 %%4419
                                                 %%4423

                         Access Reasons:         %%1538: %%1805
                                                 %%1541: %%1805
                                                 %%4416: %%1805
                                                 %%4419: %%1805
                                                 %%4423: %%1811  D:(A;OICIID;0x1200a9;;;BU)

                         Access Mask:            0x120089
                         Privileges Used for Access Check:       -
                         Restricted SID Count:   0
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {S-1-5-21-1598544264-1886666874-1118049692-1009, userFAU, SURFACEBOOK2, 0x218e56...}
InstanceId          : 4656
TimeGenerated       : 12/4/2018 1:45:28 PM
TimeWritten         : 12/4/2018 1:45:28 PM
UserName            :
Site                :
Container           :
```

© 2019 Microsoft Corporation. All rights reserved.

```
EventID             : 4656
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 19625
Category            : (12800)
CategoryNumber      : 12800
EntryType           : SuccessAudit
Message             : A handle to an object was requested.

                      Subject:
                         Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                         Account Name:           Administrator
                         Account Domain:         SURFACEBOOK2
                         Logon ID:               0x5cebe

                      Object:
                         Object Server:          Security
                         Object Type:            File
                         Object Name:            C:\TEMP\file.txt
                         Handle ID:              0xb70
                         Resource Attributes:    -

                      Process Information:
                         Process ID:             0x4a8
                         Process Name:           C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

                      Access Request Information:
                         Transaction ID:         {00000000-0000-0000-0000-000000000000}
                         Accesses:               %%1538
                                                 %%1541
                                                 %%4416
                                                 %%4419
                                                 %%4423

                         Access Reasons:         %%1538: %%1804
                                                 %%1541: %%1801  D:(A;;FA;;;BA)
                                                 %%4416: %%1801  D:(A;;FA;;;BA)
                                                 %%4419: %%1801  D:(A;;FA;;;BA)
                                                 %%4423: %%1801  D:(A;;FA;;;BA)

                         Access Mask:            0x120089
                         Privileges Used for Access Check:       -
                         Restricted SID Count:   0
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {S-1-5-21-1598544264-1886666874-1118049692-500, Administrator, SURFACEBOOK2, 0x5cebe...}
InstanceId          : 4656
TimeGenerated       : 12/4/2018 1:45:41 PM
TimeWritten         : 12/4/2018 1:45:41 PM
UserName            :
Site                :
Container           :
```

As it can be observed in the previous images, the audit events related to the use of privileged rights and file and object events have been correctly generated. Additionally, the generated audit log entries include all the fields defined in the ***Security Target*** document.

Moreover, the evaluator has reviewed that the event, with ID *4670*, is also generated. This event, is related to permission modifications on an object.

© 2019 Microsoft Corporation. All rights reserved.

```
EventID            : 4670
MachineName        : SURFACEBOOK2
Data               : {}
Index              : 19612
Category           : (13570)
CategoryNumber     : 13570
EntryType          : SuccessAudit
Message            : Permissions on an object were changed.

                     Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x5cebe

                     Object:
                        Object Server:  Security
                        Object Type:    File
                        Object Name:    C:\TEMP\file.txt
                        Handle ID:      0xb98

                     Process:
                        Process ID:     0x4a8
                        Process Name:   C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

                     Permissions Change:
                        Original Security Descriptor:   D:AI(A;ID;FA;;;BA)(A;ID;FA;;;SY)(A;ID;0x1200a9;;;BU)(A;ID;0x1301bf;;;AU)
                        New Security Descriptor:        D:PARAI(A;;FA;;;BA)
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-21-1598544264-1886666874-1118049692-500, Administrator, SURFACEBOOK2, 0x5cebe...}
InstanceId         : 4670
TimeGenerated      : 12/4/2018 1:45:27 PM
TimeWritten        : 12/4/2018 1:45:27 PM
UserName           :
Site               :
Container          :
```

4.2.4.4 Verdict

As the result above states, the audit events related to file and object events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4.**

4.2.5 Test 5 - Cryptographic verification of software events

Generation of audit events related to cryptographic verification of software will be tested in this test case.

4.2.5.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- *SignTool*, a command line tool that provides the ability to sign files and verify signatures in files. It is distributed with the *Windows 10 Software Development Kit (SDK)*.
- A hexadecimal editor (e.g. *HxD*).
- A valid update file must be downloaded in the tested platform. To do this, the evaluator shall carry out the following steps:
  - Open *Microsoft Edge* and go to the following URL: Windows Update Catalog
- In the search box type: *Windows 10 1809* for Windows 10 and *Windows Server 2019* for Windows Server. A list of available updates will be shown.
  - Choose one update from the list, and ensure that the selected update is valid to the architecture of the platform which is being tested. Once the operating system and the architecture of the update file have been checked.

© 2019 Microsoft Corporation. All rights reserved.

- Finally, click the *Download* button. Choose the folder where the update will be stored and wait until the download has finished. The downloaded file shall have the *.msu* extension (*Microsoft Update Standalone Package*).
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.5.2 Procedure

The evaluator has automated this subtest by developing a script. First, the following operation shall be performed:

- Run as administrator the script *FPT_TUD_EXT.1 - Test 1.ps1*. To do this, type the following command in a PowerShell terminal:

  ```
  .\FPT_TUD_EXT.1 - Test 1.ps1
  ```

- The script modifies the original update files for all the architectures (x86, x64 and Server). The update files must be copied to the folder directory that has been recently created, located on *Input/Original*.
- Then, an attempt to install the modified update file (located in *Input/Invalid*) should be performed. The operating system shall reject the operation.
- After that, attempt to install the original update file. The update shall be successfully installed.

Finally, the evaluator shall carry out the following steps to check the proper generation of the audit events:

- Run as administrator the script *Test5 - Cryptographic verification events.ps1*. To do this, type the following command in a *PowerShell* terminal:

  ```
  .\Test5 - Cryptographic verification events.ps1
  ```

- Observe the results.

4.2.5.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to install a modified update and the OS has rejected the operation:

The following event audit was generated:

```
Message              : Windows update "Security Update for Windows (KB4467708)" could not be installed because of
                       error 2149867522 "" (Command line: ""C:\Windows\system32\wusa.exe" "D:\Updates -
                       Modified\Invalid\X64\Invalidx64.msu" ")
Id                   : 3
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 0
Opcode               : 0
Keywords             : -9223372036854775808
RecordId             : 2
ProviderName         : Microsoft-Windows-WUSA
ProviderId           : 09608c12-c1da-4104-a6fe-b959cf57560a
LogName              : Setup
ProcessId            : 10844
ThreadId             : 9580
MachineName          : SURFACEBOOK2
UserId               : S-1-5-21-1598544264-1886666874-1118049692-500
TimeCreated          : 12/5/2018 9:26:43 AM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Setup
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty}
```

After that, the evaluator has attempted to install the original update. The OS has successfully installed the update:

© 2019 Microsoft Corporation. All rights reserved.

And the following audit event was created:

```
Message              : Windows update "Security Update for Windows (KB4467708)" was successfully installed. (Command
                       line: ""C:\Windows\system32\wusa.exe" "D:\Updates - Modified\Modified\X64\Modifiedx64.msu" ")
Id                   : 2
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 0
Opcode               : 0
Keywords             : -9223372036854775808
RecordId             : 6
ProviderName         : Microsoft-Windows-WUSA
ProviderId           : 09608c12-c1da-4104-a6fe-b959cf57560a
LogName              : Setup
ProcessId            : 10204
ThreadId             : 7624
MachineName          : SURFACEBOOK2
UserId               : S-1-5-21-1598544264-1886666874-1118049692-500
TimeCreated          : 12/5/2018 9:27:58 AM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Setup
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty}
```

4.2.5.4 Verdict

As the result above states, the audit events related to the cryptographic verification of software have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 5** requirements established in the assurance activity section.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to **Test 5.**

4.2.6 Test 6 - Program initiation events

Generation of audit events related to the success or failure initiation programs due to software restriction policies will be tested in this test case.

4.2.6.1 Setup

The applicable setup is different depending on the operating system and the edition installed in the evaluated platform.

4.2.6.1.1 For Windows 10 Enterprise and Windows Server 2019 - Applocker

The evaluator shall configure *Applocker*. The following conditions must be fulfilled to perform this part of the test:

- *Application Identity Service* must be running. The evaluator has developed a script to automate the start of the service and set it to start on Windows boot. The content of the script is the following:

```
#
# Test 6. Program initiation events - Applocker
#

#Check service status
Write-Host "[I]: Checking Application Identity Service status..."
$currentstatus = Get-Service -Name 'AppIDSvc'

if ($currentStatus.Status -eq "Stopped") {
    Write-Host "[E]: Application Identity Service is stopped"
    Write-Host "[I]: Starting the service..."
    Start-Service AppIDSvc
    Set-ItemProperty -Path
    'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
    Services\AppIDSvc'
     -Name Start -Value 2
    Start-Process 'gpupdate.exe' -ArgumentList '/force' -WorkingDirectory
    "C:\Windows\System32"
} else {
    Write-Host "[I]: Application Identity Service is running"
    Write-Host "[I]: Setting the service to automatic start..."
    Set-ItemProperty -Path
    'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
    Services\AppIDSvc'
     -Name Start -Value 2
    Start-Process 'gpupdate.exe' -ArgumentList '/force' -WorkingDirectory
    "C:\Windows\System32"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    }
```

- *AppLocker* shall not have any defined rules in section *Packaged app Rules*.

The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.6.1.2 For Windows 10 Home and Pro Edition - Device Guard

The evaluator shall configure Device Guard to obtain the same behaviour in the platforms which have Windows 10 Home or Pro Edition installed. The following conditions must be fulfilled to perform this part of the test:

- *cipolicy.bin* file shall be available in the tested platform. This file contains the program restriction polices that will be applied. This policy blocks the execution of the version 1.0 of the *TestConsole* executable and allows the execution of the version 1.1.
- *TestConsolev10.exe* and *TestConsolev11.exe* shall be available on the tested platform. These files, which have been provided by the OS vendor, are executable files signed by an untrusted certification authority.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.6.2 Procedure

AppLocker policies are not applicable to all the TOE editions (they are only applicable for Windows 10 Enterprise and Windows Server 2019). The evaluator has divided this test in two different parts:

- On the first part, an *AppLocker* policy is configured in order to restrict the application (*.appx* files on Windows and *.exe* files on Windows Server) execution. This method only applies to Windows 10 Enterprise edition and Windows Server 2019.
- On the second part, the evaluator shall copy a pre-configured policy to restrict the execution of executable files which have not been signed by a trusted certification authority. This method applies to Windows Home and Pro editions.

The evaluator shall carry out the following steps for the tested platform to check that the TOE audits the analyzed events:

4.2.6.2.1 For Windows 10 Enterprise and Windows Server 2019 - Applocker

- The evaluator shall configure *AppLocker*. To do this the evaluator shall open the Local Security Policy tool (*WIN+R* and then type *secpol.msc*) and go to *Application Control Policies -> AppLocker -> Packaged app Rules*.

© 2019 Microsoft Corporation. All rights reserved.

- Right-click in the *Packaged app Rules* node and select *Create New Rule…* menu item.

© 2019 Microsoft Corporation. All rights reserved.

- A new wizard shall be shown, in which, the evaluator can configure the software restriction policy that is going to be applied. The evaluator shall configure the wizard as shown the next images:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

- Once the software restriction policy has been configured. It has to be enforced to be applied. To enforce the new rule go to *Applocker* option and right click on it. Then select the *Properties* option.

© 2019 Microsoft Corporation. All rights reserved.

- After that, select the checkbox *Package app Rules* and the option *Enforce rules* in the list.

© 2019 Microsoft Corporation. All rights reserved.

- To check that the new rule is successfully applied, run the *Settings* application (*WIN+I*). The operating system shall reject the execution and an error message shall be shown. Additionally, a new audit event shall be generated indicating that the application has not been executed successfully.
- After that, modify the previous created rule to allow the execution of the application. To do that run the *Local Security Policy* tool (*WIN+R, then type secpol.msc*) and go to *Application Control Policies -> AppLocker -> Packaged app Rules* section. Double click in the created rule and replace the action value *Deny* with *Allow*.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator shall carry out the following steps to check the generation of the audit events:

- Run as administrator the script *Test6 - Program Initiation Events-Applocker-GetLog.ps1*. To do this, type the following command in a *PowerShell* terminal:

  ```
  .\Test6 - Program Initiation events-Applocker-GetLog.ps1
  ```

- Observe the results.

For Windows 10 Home and Pro Edition - Device Guard

The evaluator has automated this subtest by developing a script. First, the following operation shall be performed:

- Run as administrator the script *Test6 - Program Initiation Events - DeviceGuard.ps1*. To do this, type the following command in a *PowerShell* terminal:

  ```
  .\Test6 - Program Initiation Events - DeviceGuard.ps1
  ```

  The TOE must be restarted so the new policy can be applied.

  After that, the previous script shall be executed again. This time, the script attempts to execute the *TestConsolev10.exe* application, which has not been signed by a trusted certification authority. The file shall not be executed properly and an error message shall be shown. Additionally, a new audit event shall be generated, indicating that there was an

© 2019 Microsoft Corporation. All rights reserved.

error during the program execution. Then, an attempt to execute the *TestConsolev11.exe* application is performed. The execution of this program shall be successful. The audit logs related to the successful and unsuccessful execution attempts are shown.

- Observe the results.

4.2.6.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

4.2.6.3.1 For Windows 10 Enterprise and Windows Server 2019 - Applocker

- The evaluator has executed the *Settings* application successfully when the *AppLocker* rule allowed the application execution.

© 2019 Microsoft Corporation. All rights reserved.

- Additionally, the following audit event was generated, indicating that the application execution was allowed.

```
Message              : WINDOWS.IMMERSIVECONTROLPANEL was allowed to run.
Id                   : 8020
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 0
Opcode               : 0
Keywords             : 2305843009213693952
RecordId             : 251
ProviderName         : Microsoft-Windows-AppLocker
ProviderId           : cbda4dbf-8d5d-4f69-9578-be14aa540d22
LogName              : Microsoft-Windows-AppLocker/Packaged app-Execution
ProcessId            : 1020
ThreadId             : 4820
MachineName          : SURFACEBOOK2
UserId               : S-1-5-21-1598544264-1886666874-1118049692-500
TimeCreated          : 12/5/2018 10:03:28 AM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-AppLocker/Packaged app-Execution
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

- The evaluator has executed the *Settings* application when the *AppLocker* rule denied the application execution. The obtained result is as follows:

© 2019 Microsoft Corporation. All rights reserved.

- And the following audit event was generated, indicating that the application execution was not allowed:

```
Message               : MICROSOFT.WINDOWS.CORTANA was prevented from running.
Id                    : 8022
Version               : 0
Qualifiers            :
Level                 : 2
Task                  : 0
Opcode                : 0
Keywords              : 2305843009213693952
RecordId              : 250
ProviderName          : Microsoft-Windows-AppLocker
ProviderId            : cbda4dbf-8d5d-4f69-9578-be14aa540d22
LogName               : Microsoft-Windows-AppLocker/Packaged app-Execution
ProcessId             : 1020
ThreadId              : 3584
MachineName           : SURFACEBOOK2
UserId                : S-1-5-21-1598544264-1886666874-1118049692-500
TimeCreated           : 12/5/2018 10:03:26 AM
ActivityId            :
RelatedActivityId     :
ContainerLog          : Microsoft-Windows-AppLocker/Packaged app-Execution
MatchedQueryIds       : {}
Bookmark              : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName      : Error
OpcodeDisplayName     : Info
TaskDisplayName       :
KeywordsDisplayNames  : {}
Properties            : {System.Diagnostics.Eventing.Reader.EventProperty,
                        System.Diagnostics.Eventing.Reader.EventProperty,
                        System.Diagnostics.Eventing.Reader.EventProperty,
                        System.Diagnostics.Eventing.Reader.EventProperty...}
```

4.2.6.3.2 For Windows 10 Home and ProEdition - Device Guard

After executing the script the first time, the pre-configure policy file is copied to the *CodeIntegrity* folder (%WINDIR%\system32\CodeIntegrity). Then the system is restarted so the new policy can be applied. The second time the script is executed, an attempt to run the *TestConsolev10.exe* executable file is performed. The operating system has rejected the execution showing the following message:

© 2019 Microsoft Corporation. All rights reserved.

- Additionally, the following audit event has been generated, indicating that there was an error during the program execution.

```
Message              : Code Integrity determined that a process (\Device\HarddiskVolume4\Windows\System32\WindowsPowerShell\v1.0\powershell.exe) attempted to
                       load \Device\HarddiskVolume4\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FAU\common\test6tools\FPT_SRP_EXT\TestConsolev10.exe that did
                       not meet the Enterprise signing level requirements or violated code integrity policy.
Id                   : 3077
Version              : 3
Qualifiers           :
Level                : 2
Task                 : 18
Opcode               : 111
Keywords             : -9223372036854775808
RecordId             : 53
ProviderName         : Microsoft-Windows-CodeIntegrity
ProviderId           : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName              : Microsoft-Windows-CodeIntegrity/Operational
ProcessId            : 8992
ThreadId             : 9140
MachineName          : SURFACELAPTOP2
UserId               : S-1-5-21-2217501322-3974158852-995746943-500
TimeCreated          : 12/6/2018 3:58:23 PM
ActivityId           : 33407269-8dbf-0001-6580-4033bf8dd401
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-CodeIntegrity/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    :
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

- Finally, an attempt to execute the *TestConsolev11.exe* executable file is performed. The operating system has successfully executed the program file.

© 2019 Microsoft Corporation. All rights reserved.

- The executable used (*TestConsolev11.exe*) is an unsigned binary (*based on a signed catalog of file hashes*). Therefore, it creates several audit events with IDs: 3038, 3015, 3008 and 3016. The following image shows the first event generated when the application is executed.

4.2.6.4 Verdict

As the result above states, the audit events related to the program initiation events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 6** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 6.**

© 2019 Microsoft Corporation. All rights reserved.

4.2.7 Test 7 - Kernel module loading events

Generation of kernel module loading events will be tested in this test case.

4.2.7.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A hexadecimal editor (e.g. *HxD*)
- A *WinPE* USB for both architectures (x64 and x86) must be available.
- The following script developed by the evaluator shall be available in the tested platforms: *Test7 - Kernel module loading events - Step 1.ps1*, which enable the *CodeIntegrity* audit log, and *Test7 - Kernel module loading events - Step 2.ps1*, which generates two text files with the CodeIntegrity audit logs.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

4.2.7.2 Procedure

The evaluator shall carry out the following steps to check that the TOE audits the analyzed events:

- First, the evaluator shall modify a file which is loaded during the boot process. In this case, the modified file is *win32k.sys*, which is stored in %windir%\system32. The evaluator shall create a copy of this file and modify any byte of it using a hexadecimal editor.

- Once the copied file has been modified, the evaluator shall enable the Code Integrity audit log. To do this, the evaluator shall execute as administrator the script *Test7 - Kernel module loading events - Step 1.ps1*. The content of this script is as follows:

```
#
# Test 7. Kernel module loading events – Step 1
#

auditpol /set /category:* /success:disable /failure:disable
```

© 2019 Microsoft Corporation. All rights reserved.

```
auditpol /set /category:System /success:enable /failure:enable
wevtutil cl security
wevtutil sl /e:false Microsoft-Windows-CodeIntegrity/Verbose
wevtutil sl /e:false Microsoft-Windows-CodeIntegrity/Operational
wevtutil cl Microsoft-Windows-CodeIntegrity/Verbose
wevtutil cl Microsoft-Windows-CodeIntegrity/Operational
wevtutil sl /e:true Microsoft-Windows-CodeIntegrity/Verbose
wevtutil sl /e:true Microsoft-Windows-CodeIntegrity/Operational
```

- Then, boot the TOE into the *Windows Pre Environment* using the *WinPE* USB.
- On the *Windows Pre Environment*, create a copy of the original file, which shall be used in the below steps to restore the corrupted file. Then, replace the original file with the one modified in the first step and restart the TOE.
- Observe that the TOE does not boot properly.
- Boot the TOE into the *Windows Pre Environment* again, and restore the original file using the backup created in the previous steps.
- Restart the TOE and observe that this time the TOE boots successfully.

The evaluator shall carry out the following steps to check the generation of the audit events:

- Run as administrator the script *Test7 - Kernel module loading events - Step 2.ps1*. To do this, type the following command in a *PowerShell* terminal:

  > .\Test6 - Program Initiation Events.ps1

- The script collects the *Windows-CodeIntegrity/Verbose* and *Windows-CodeIntegrity/Operational* audit logs of the TOE.
- The content of this script is as follows:

```
#
# Test 7. Kernel module loading events - Step 2
#

If (-NOT ([Security.Principal.WindowsPrincipal][Security.Principal.
WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.
WindowsBuiltInRole] "Administrator"))
{
$arguments = "& '" + $myinvocation.mycommand.definition + "'"
Start-Process powershell -Verb runAs -ArgumentList $arguments
Break
}

$path = "$env:USERPROFILE\Desktop"
$events = Get-WinEvent -logname Microsoft-Windows-CodeIntegrity/Verbose -Oldest
$events | Format-List * *> "$path\CodeIntegrityOutputVerbose.txt"
$events = Get-WinEvent -logname Microsoft-Windows-CodeIntegrity/Operational
    -Oldest
$events | Format-List * *> "$path\CodeIntegrityOutputOperational.txt"
```

© 2019 Microsoft Corporation. All rights reserved.

```
auditpol /set /category:* /success:disable /failure:disable
```

- Analyze the audit logs obtained in the above step.

4.2.7.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

When the boot process fails, an event log is generated in *Windows CodeIntegrity/Operational*, indicating the name of the corrupted file:

```
Message                 : Windows is unable to verify the image integrity of the
    file \Device\HarddiskVolume4\Windows\System32\win32k.sys because file hash
    could not be found on the system. A recent hardware or software change might
    have installed a file that is signed incorrectly or damaged, or that might
    be malicious software from an unknown source.
Id                      : 3004
Version                 : 1
Qualifiers              :
Level                   : 2
Task                    : 1
Opcode                  : 104
Keywords                : -9223372036854775808
RecordId                : 17
ProviderName            : Microsoft-Windows-CodeIntegrity
ProviderId              : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName                 : Microsoft-Windows-CodeIntegrity/Operational
ProcessId               : 540
ThreadId                : 544
MachineName             : SURFACEBOOK2
UserId                  : S-1-5-18
TimeCreated             : 12/3/2018 2:48:28 PM
```

© 2019 Microsoft Corporation. All rights reserved.

```
ActivityId              : 47408b39-8b5a-0006-938b-40475a8bd401
RelatedActivityId       :
ContainerLog            : Microsoft-Windows-CodeIntegrity/Operational
MatchedQueryIds         : {}
Bookmark                : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName        : Error
OpcodeDisplayName       :
TaskDisplayName         :
KeywordsDisplayNames    : {}
Properties              : {System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty...}
```

When the TOE boots properly, many audit events are generated indicating that the integrity of files loaded during the boot process are successfully checked. An example of this event type is as follows:

```
Message                 : Code Integrity started validating image header of
    \Windows\System32\win32k.sys file.
Id                      : 3038
Version                 : 2
Qualifiers              :
Level                   : 5
Task                    : 13
Opcode                  : 1
Keywords                : 4611686018427387904
RecordId                : 407
ProviderName            : Microsoft-Windows-CodeIntegrity
ProviderId              : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName                 : Microsoft-Windows-CodeIntegrity/Verbose
ProcessId               : 548
ThreadId                : 552
MachineName             : SURFACEBOOK2
UserId                  : S-1-5-18
TimeCreated             : 12/5/2018 10:19:52 AM
ActivityId              : 1633ec08-8cc7-0006-52ec-3316c78cd401
RelatedActivityId       :
ContainerLog            :
    C:\Windows\System32\Winevt\Logs\Microsoft-Windows-CodeIntegrity%4Verbose.etl
MatchedQueryIds         : {}
Bookmark                : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName        : Verbose
OpcodeDisplayName       : Start
TaskDisplayName         :
KeywordsDisplayNames    : {}
Properties              : {System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty,
                          System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

4.2.7.4 Verdict

As the result above states, the audit events related to the kernel module loading and unloading events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 7** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 7.**

4.2.8 Test 8 - Lock and unlock user accounts

Generation of events related to lock and unlock user account will be tested in this test case.

4.2.8.1 Setup

The applicable setup for this test is the same as the one defined for *FIA_AFL.1*.

4.2.8.2 Procedure

The evaluator shall carry out the same procedure as one defined for *FIA_AFL.1*.

4.2.8.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

During the test execution of *FIA_AFL.1*, the following results were obtained. Those results state that the events related to lock and unlock the user account are generated successfully.

© 2019 Microsoft Corporation. All rights reserved.

After reaching the configured threshold of failure authentication attempts, the user account is locked and the following audit event is generated.

```
Message              : A user account was locked out.

                       Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3E7

                       Account That Was Locked Out:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1003
                        Account Name:           userFIA

                       Additional Information:
                        Caller Computer Name:   SURFACEBOOK2
Id                   : 4740
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13824
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 14055
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 852
ThreadId             : 952
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 10:40:14 AM
ActivityId           : 6141e5e7-8bfe-0007-0dee-4161fe8bd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : User Account Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

Once the user account has been locked, it has to be unlocked using an administrator account or wait to the configured auto unlock time. The following audit event has been generated:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A user account was unlocked.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x1C218D

                       Target Account:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1003
                        Account Name:           userFIA
                        Account Domain:         SURFACEBOOK2
Id                   : 4767
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13824
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 14302
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 852
ThreadId             : 8396
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 10:42:41 AM
ActivityId           : 6141e5e7-8bfe-0000-6de6-4161fe8bd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : User Account Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

As it can be observed in the previous images, the audit events related to the lock and unlock of a user account events have been correctly generated. In addition, the generated audit log entries include all the fields defined in the ***Security Target*** document.

4.2.8.4 Verdict

As the result above states, the audit events related to the lock and unlock user account events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 8** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 8.**

4.3 Final Verdict

Due to documentation review activity and all subtests have assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FAU_GEN.1.1.

© 2019 Microsoft Corporation. All rights reserved.

5 FAU_GEN.1.2

The assurance activity for the **FAU_GEN.1.2** requirement is stated as follows:

> The evaluator will check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must becovered, along with a brief description of each field. The evaluator will ensure that the fields contains the information required.
>
> The evaluator shall test the OS's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the ST.The evaluator will ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record provide the required information.

5.1 Documentation Review activity

5.1.1 Findings

The evaluator has reviewed the ***Operational Guidance*** document. This document includes in its section **7.1 Audit Events - GP OS protection profile** a table with all the auditable events. The content of this table matches with the selection performed by the vendor in the ***Security Target*** document.

For example, the following image shows some of the auditable events defined in the Operational Guidance document:

Description	**Context: Event ID (Detail)** **Note – *all* events are in the Security log.**
Start-up and shut-down of the audit functions	Start-up: **4608** Shut-down: **1100**
Authentication events (Success/Failure)	Success: **4624** Failure: **4625**
Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes)	WRITE_DAC : **4670** All other object access writes : **4656**
Privilege or role escalation events (Success/Failure)	Success: **4673** Failure: **4674**

The ***Security Target*** document also states the minimum information that each audit record should include. These fields are the following:

- Date and time of the event.
- Type of the event.
- Subject identity
- Outcome (success or failure) of the event.

© 2019 Microsoft Corporation. All rights reserved.

In addition, the ***Operational Guidance*** document also provides information related the main fields for each auditable event. This information includes the name of these fields and a brief description for each one. For example, the following image shows the main required fields for the auditable event 2 (*Package was successfully changed to the Installed state*).

Event ID	Location in Log	Message	Fields
2		Package was successfully changed to the Installed state	**System->TimeCreated[SystemTime]**: <Date and time of event> **System->Provider[Name]**: <Type of event> **System->Security[UserID]**: <Subject identifier > **System->Level**: <Outcome as Success or Failure>

5.1.2 Verdict

The evaluator has reviewed the ***Security Target*** document and has ensured that the format of every auditable event is described, including at least the fields defined in the ***Security Target*** document (*date and time, type of the event, subject identity and outcome*).

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

5.2 Test Activity

This test is done at the same time as *FAU_GEN.1.1*.

5.2.1 Test

5.2.1.1 Setup

The applicable setup for this test is the same as the one defined for *FAU_GEN.1.1*.

5.2.1.2 Procedure

This test is done at the same time as *FAU_GEN.1.1*.

5.2.1.3 Results

Results for this test are included in *FAU_GEN.1.1*.

5.2.1.4 Verdict

As it is can be appreciated in the obtained results section of *FAU_GEN.1.1* requirement, all the audit events generated by the TOE match with the format described in the operational guidance and include, at least, the fields listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to the test activity.

5.3 Final Verdict

Since this requirement has been tested alongside the *FAU_GEN.1.1* requirement, where all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FAU_GEN.1.2 requirement.

© 2019 Microsoft Corporation. All rights reserved.

6 FCS_CKM.1.1(1)

The assurance activity for the **FCS_CKM.1.1(1)** requirement is stated as follows:

> The evaluator will ensure that the TSS identifies the key sizes supported by the OS. If the ST specifies more than one scheme, the evaluator will examine the TSS to verify that it identifies the usage for each scheme.
>
> The evaluator will verify that the AGD guidance instructs the administrator how to configure the OS to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.
>
> Assurance Activity Note: The following tests may require the vendor to furnish a developer environment and developer tools that are typically not available to end-users of the OS.
>
> **Key Generation for FIPS PUB 186-4 RSA Schemes**
>
> The evaluator will verify the implementation of RSA Key Generation by the OS using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include:
>
> 1. Random Primes:
> - Provable primes
> - Probable primes
> 2. Primes with Conditions:
> - Primes p1, p2, q1,q2, p and q shall all be provable primes
> - Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes
> - Primes p1, p2, q1,q2, p and q shall all be probable primes
>
> To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator will verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.
>
> If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator will have the TSF generate 10 keys pairs for each supported key length nlen and verify:
>
> - n = p*q,
> - p and q are probably prime according to Miller-Rabin tests,
> - GCD(p-1,e) = 1,

© 2019 Microsoft Corporation. All rights reserved.

- GCD(q-1,e) = 1,
- $2^{16} \leq e \leq 2^{256}$ and e is an odd integer,
- $|p-q| > 2^{nlen/2-100}$,
- $p \geq 2^{nlen/2-1/2}$,
- $q \geq 2^{nlen/2-1/2}$,
- $2^{(nlen/2)} < d < LCM(p-1,q-1)$,
- e*d = 1 mod LCM(p-1,q-1).

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator will require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator will submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator will generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator will obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.

The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

- Cryptographic and Field Primes:
  - Primes q and p shall both be provable primes
  - Primes q and field prime p shall both be probable primes

and two ways to generate the cryptographic group generator g:

- Cryptographic Group Generator:
  - Generator g constructed through a verifiable process
  - Generator g constructed through an unverifiable process

The Key generation specifies 2 ways to generate the private key x:

- Private Key:
  - len(q) bit output of RBG where $1 \leq x \leq q-1$

© 2019 Microsoft Corporation. All rights reserved.

- len(q) + 64 bit output of RBG, followed by a mod q-1 operation where $1 \leq x \leq q-1$

The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm:

- g != 0,1
- q divides p-1
- g^q mod p = 1
- g^x mod p = y

for each FFC parameter set and key pair.

6.1 Documentation Review activity

6.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[1]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the NIST CAVP certificates which documents the key sizes supported by the OS versions for every supported algorithm. Two key generation schemes for digital signature purposes are used and both of them conforming to the *FIPS 186-4* standard:

- *Certificates number #C 211*[2] *and #C 348*[3] *for RSA and,*
- *Certificates number #C 211, #C 348 and #C 350*[4] *for ECDSA.*

[1] The test results are described in the evaluation and Assurance Activity Reports

[2] Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

[3] Validation for #C 348:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30744

[4] Validation for #C 350:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

RSA KeyGen

- Public Exponent Mode: Fixed
- Fixed Public Exponent: 10001
- Private Key Format: standard
- Capabilities:
  - Key Generation Mode: B.3.6
  - Properties:
    - Modulo: 2048
    - Primality Tests: C.3
  - Properties:
    - Modulo: 3072
    - Primality Tests: C.3
- Prerequisites:
  - C 211 (DRBG)
  - C 211 (SHS)

ECDSA SigGen

- Capabilities:
  - Curve: P-256
  - Hash Algorithm: SHA2-256
- Capabilities:
  - Curve: P-384
  - Hash Algorithm: SHA2-384
- Capabilities:
  - Curve: P-521
  - Hash Algorithm: SHA2-512
- Prerequisites:
  - C 211 (DRBG)
  - C 211 (SHS)

The ***Operational Guidance*** document, states that for the evaluated version the following security policy needs to be applied (Section 5.2.5.2):

© 2019 Microsoft Corporation. All rights reserved.

> - Local Policies\Security Options\System cryptography: Use FIPS 140 compliant cryptographic algorithms, including encryption, hashing and signing algorithm

After applying this policy, only FIPS certified algorithms can be used, including the key generation algorithms defined in the table above.

6.1.2 Verdict

The evaluator considers that the TSS identifies the key sizes supported by the OS for every algorithm through its NIST certificates. Moreover, all the key generation algorithms (whose purpose is to be used as part of digital signatures processes) follow the same standard (*FIPS 186-4*).

The ***Operational Guidance*** document, defines the FIPS security policy to be applied. Once this policy is applied, only the approved key generation method described above can be used. No further configuration is needed to generate keys following the *Appendix B.3* and *B.4* of the *FIPS-PUB 186-4* standard.

Hence, the **PASS** verdict is assigned to the documentation review activity.

6.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the **PASS** verdict is assigned.

6.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_CKM.1.1 requirement.

© 2019 Microsoft Corporation. All rights reserved.

7 FCS_CKM.2.1(1)

The assurance activity for the **FCS_CKM.2.1(1)** requirement is stated as follows:

> The evaluator will ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator will examine the TSS to verify that it identifies the usage for each scheme.
>
> The evaluator will verify that the AGD guidance instructs the administrator how to configure the OS to use the selected key establishment scheme(s).
>
> Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.
>
> **Key Establishment Schemes**
>
> The evaluator will verify the implementation of the key establishment schemes supported by the OS using the applicable tests below.
>
> **SP800-56A Key Establishment Schemes**
>
> The evaluator will verify the OS's implementation of SP80056A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that the OS has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the discrete logarithm cryptography (DLC) primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator will also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MAC data and the calculation of MAC tag.
>
> **Function Test**
>
> The Function test verifies the ability of the OS to implement the key agreement schemes correctly. To conduct this test the evaluator will generate or obtain test vectors from a known good implementation of the OS's supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.
>
> The evaluator will obtain the DKM, the corresponding OS's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and OS id fields.

© 2019 Microsoft Corporation. All rights reserved.

If the OS does not use a KDF defined in SP 800-56A, the evaluator will obtain only the public keys and the hashed value of the shared secret.

The evaluator will verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the OS shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the OS to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator will obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the OS should be able to recognize. The evaluator generates a set of 30 test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the OS's public/private key pairs, MAC tag, and any inputs used in the KDF, such as the other info and OS id fields.

The evaluator will inject an error in some of the test vectors to test that the OS recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MAC'd, or the generated MAC tag. If the OS contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the OS's static private key to assure the OS detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The OS shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator will compare the OS's results with the results using a known good implementation verifying that the OS detects these errors.

SP800-56B Key Establishment Schemes

The evaluator will verify that the TSS describes whether the OS acts as a sender, a recipient, or both for RSA-based key establishment schemes.

If the OS acts as a sender, the following assurance activity shall be performed to ensure the proper operation of every OS supported combination of RSA-based key establishment scheme:

To conduct this test the evaluator will generate or obtain test vectors from a known good implementation of the OS's supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall gen-

© 2019 Microsoft Corporation. All rights reserved.

erate 10 sets of test vectors. Each test vector shall include the RSA public key, the plaintext keying material, any additional input parameters if applicable, the MAC key and MAC tag if key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform a key establishment encryption operation on the OS with the same inputs (in cases where key confirmation is incorporated, the test shall use the MAC key from the test vector instead of the randomly generated MAC key used in normal operation) and ensure that the outputted ciphertext is equivalent to the ciphertext in the test vector.

If the OS acts as a receiver, the following assurance activities shall be performed to ensure the proper operation of every OS supported combination of RSA-based key establishment scheme:

To conduct this test the evaluator will generate or obtain test vectors from a known good implementation of the OS's supported schemes. For each combination of supported key establishment scheme and its options (with our without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA private key, the plaintext keying material, any additional input parameters if applicable, the MAC tag in cases where key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator will perform the key establishment decryption operation on the OS and ensure that the outputted plaintext keying material is equivalent to the plaintext keying material in the test vector. In cases where key confirmation is incorporated, the evaluator will perform the key confirmation steps and ensure that the outputted MAC tag is equivalent to the MAC tag in the test vector.

The evaluator will ensure that the TSS describes how the OS handles decryption errors. In accordance with NIST Special Publication 800-56B, the OS must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations. If KTS-OAEP is supported, the evaluator will create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each. If KTS-KEM-KWS is supported, the evaluator will create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.

7.1 Documentation Review activity

7.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[5]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the NIST CAVP certificates, where it is defined the key establishment scheme met for the fulfillment of the Special Publication *800-56A*:

- *Certificates number #C 211*[6]*and #C 350*[7].

[5]The test results are described in the evaluation and Assurance Activity Reports
[6]Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607
[7]Validation for #C 350:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

KAS-ECC

Function: Domain Parameter Generation, Domain Parameter Validation, Key Pair Generation, Partial Public Key Validation, Public Key Regeneration

Scheme:

- Ephemeral Unified:
  - KAS Role: Initiator, Responder
  - KDF without Key Confirmation:
    - KDF Option:
      - Concatenation:
    - Parameter Set:
      - EC:
        - Curve: P-256
        - Hash Algorithm: SHA2-256
        - MAC Option:
          - HMAC-SHA2-256:
            - Key Length: 256
            - MAC Length: 256
      - ED:
        - Curve: P-384
        - Hash Algorithm: SHA2-384
        - MAC Option:
          - HMAC-SHA2-384:
            - Key Length: 256
            - MAC Length: 256
      - EE:
        - Curve: P-521
        - Hash Algorithm: SHA2-512
        - MAC Option:
          - HMAC-SHA2-512:
            - Key Length: 256
            - MAC Length: 256
- One Pass DH:
  - KAS Role: Initiator, Responder
  - KDF without Key Confirmation:
    - KDF Option:

© 2019 Microsoft Corporation. All rights reserved.

As it is observed in the image above the ECC KAS (*Elliptic Curve Key Agreement Scheme*) is supported by Windows 10 October 2018 Update.

The TSS distinguishes between two key establishment schemes, the elliptic curve Diffie-Hellman which is used in TLS and the RSA-based which is used for establishing a shared secret key (the TOE can act both as a sender or recipient).

In addition, for the RSA-based key establishment scheme (*SP 800-56B*) the TSS states that, any decryption errors which occur during key establishment are presented to the user at a highly abstracted level.

The ***Operational Guidance*** document, states that for the evaluated version the following security policy needs to be applied (Section 5.2.5.2):

> - Local Policies\Security Options\System cryptography: Use FIPS 140 compliant cryptographic algorithms, including encryption, hashing and signing algorithm.

After applying this policy, only FIPS certified algorithms can be used, including the key agreement algorithms defined in the above table. In addition, the vendor has included the following wording, specifying that no further configuration is needed:

> *Windows automatically implement RSA-based key establishment schemes that meet SP-800-56B, no configuration is necessary.*
>
> *Windows automatically implement elliptic curve-based key establishment schemes that meet SP-800-56A, no configuration is necessary.*

7.1.2 Verdict

The evaluator considers that the TSS identifies all the key agreement schemes and its algorithms involved according to the *SP-800-56A* and *SP-800-56B* standards.

The AGD guidance defines the FIPS security policy to be applied. Once this policy is applied, only the key agreement schemes described above can be used. No more configuration is needed for using the supported key agreement schemes.

Hence, the **PASS** verdict is assigned to the documentation review activity.

7.2 Test Activity

The evaluator has reviewed the NIST certificates (*#C 211* and *#C 350*) and considers that for the elliptic curve-based key establishment scheme the CAVP certificates meet the *SP-800-56A*, therefore this part of the test activity is covered. In addition, given that the TOE can act as sender and recipient for the RSA-based key establishment scheme that meets SP-800-56B, the Lab considers that this test cannot be completely covered, only using the CAVP certificates (*#C 211* and *#C 348*), due to the CAVS Tool (*21.4*) only supports RSADP (recipient). So, to test the sender side (the recipient is also tested) the Laboratory has carried out the following test:

© 2019 Microsoft Corporation. All rights reserved.

7.2.1 Test 1

7.2.1.1 Setup

Based on the information provided in the *TSS*, it is assumed that neither the key confirmation nor the *KTS-OAEP* nor additional input parameters are supported. So, the testing is carried out only considering the *RSA key pair* (private and public key), plaintext and ciphertext.

Plaintext for the tests has been obtained from the NIST CAVS tool (21.4) with the following configuration:

Based-on this configuration the following input vectors are generated (30 vectors of 2048 bits length, only the first 2 are shown and only the first 10 are used for this evaluation):

```
[mod = 2048]
        COUNT = 0
        c = 1625cdda17e2e8cfa247ef5f43ac25cfff9550de66e4199d164db581800bc5b7efd17bb7c604ff1c082711b31e97e
fd2d57879eba7ef9360ee91cb992f1c3c551465b13dba920ebfae89ec1902a984b44cb7ceb4f11beacda44cbe582989e1a65fcab7
947df6de5f9c56c9ba764aab1e6d3a9762cfd0e58b5864d7e5f8b87d53d7e6d5e3979637a5d974f0be02e22eb83fe91c1cc6764d4
35fc35e692313d70384a253528edce1396dcda94c2ed9d329caada40b6075ea9cc24f9b0b3300b95f6fe7780bec2fa3c260483c8c
8397631a166c78562435843f88f2d5f4afe9e90e9f9b8e3338f646e9baccf0a78884a5322c10e1279a1fe3d3b993564b1f3530b9

        COUNT = 1
        c = 1cd94856d065cd8d1805c16f7529ea7bf5f2ee938adf65b8bba831169a6a1240f60575bcd1c28dc2143aa5418273b
52eeb4b33d2a21a2bfb319392924aebd816013ebc16392e64f45ccdade950bdc049a695e60814879d7e2540eb672f10f22e864bc5
6befef7aaa5786806ec54e33758e98ae3ec86ee0c90078e39e30218810ebf537c3db4df66c6d2fc7d8cb7037300b1b347ba5171f6
42b03223d3465c4459a04ba26e59200c2068e292f4969f40085e81fc795c980d80ca9514f2525cc54f940f89df504c2358a6d4f7c
2682917064c5182a80cd2bab0d3218d9eaa9c8c57270962ff3ed614c6192e0c74c033605107bc7ad4c6c496895651ee56b39e121

        COUNT = 2
        c = 1e4935144248b36ff815c11f9286f43d5717e0d6b3d476eb2c7a91bc98deeb478d7c950b1a8c8711b4f2f42d73bcc
2827fc27844c58999c2ee6731a391e8ad6fa771ef524023a21d7502cb8e9e57a1d1c5539f465af4d785272d89a08741f4a8d3706c
721ed5ad9a228f6fc9791f3b3411117c65dadfd93ec112c23be2aa49f7f9c034f21a6a5091a3df08666edb35b24b453828ad1047f
4a35f04fc0aec546602aa6e5a9ea93209e03abdeee4553a535b37fa183b4fc8b0b55b766c68cfbcfcd4764333115bfc0ac0eab6e8
4353f020292febbbec660479df504112621a2ac1586cda03db48559d2e361cdcf29eff209d75339c291ba45709848c76619745be

        COUNT = 3
        c = 1fd1f67427bb64e9ccfb5830d0aa5a4290c44324547e45dee6d92b75c777e2db2ead392992dbe5cbfaf8c8adf3717
3ad3334e9507b07a5b3c21fdfc58fee0cb4a73ca4a5e87927a8755262eb1382ef2ed3ebfb76f2e54b06fdd6818b38039c9741a5bf
50f140336a8330ae52f884d6acb91001007fbe3fb070259a2e8b5d9c6b4511129374d510f76dba354b692030112ecc8355eb3b879
b6335b1164da5131afba726d43be3c6397b1a7e3c30fe064478470addfd032dad1d2d4eab49820a8bab90827c0c105e16f475128e
13c55f2ddfa81d007fbf39d0e836fa54469d88a69df3aef4b004df7822f677aadb1e44b4ad1845263716b1eb0a783b7c0b9d9553

        COUNT = 4
        c = 1221e47ee956a35fb7bd6f3b9fc93b447d892c1b37dafba5f2dc9c19d9a21d57192ed07c309cacb3689df14e9e270
b15486bb9ddb7086e5d997ef83ec1e55f712bdf10c53acb3ab9453b798e52a8b2a121653ef86dd24a1b6ed27d25a27fdc1acd1ceb
7b3f4a1518557f4e2642b5f7910f5f0294e1a156867bc075f703d85cfb09cdabc6c5830579a052ba3cf7b4a38d1b414ed79cde5b4
7c73126db6d58a7bce6da96cd56ddd2842f9a03f9f70f7e3c4cf36aec25ef21055a0bd9fa676784046d2bf5b979c043e009417284
cc2c5047aa5e9ef9043e70693d38d77b796dbc7ba4a80fb1b7942236362d4f06fd75e1553d683227c231101b779e65872bd217ca
```

© 2019 Microsoft Corporation. All rights reserved.

The following environment is required to perform the test:

- Any of the platforms listed in the ST.
- A computer with:
  - Operating System: Debian 9 (64 bits)
  - The *OpenSSL FIPS* Object Module 2.0.13 (*fipscanister.o* file) and OpenSSL 1.0.2k.

This software shall be installed on the Debian 9 machine, as it can be observed in the next code snippet:

```
[evaluador@localhost Desktop]$ openssl version
OpenSSL 1.0.2k-fips  26 Jan 2017
```

For the installation of the *OpenSSL FIPS Object Module 2.0.13*, hereinafter called *OpenSSL FIPS*, all the steps defined in the *User Guide for the OpenSSL FIPS Object module v2.0* document have been followed.

The Lab has developed two different applications: RSA_ENC.cpp, which receives a set of plaintext as input and generates a set of ciphertext; and RSA_DEC.cpp, which receives a set of ciphertext as input and decrypts them to obtain the set of plaintext. These applications have been developed based on the public APIs for W10O18U to test the RSA-based key establishment scheme conforming to the special publication 800-56B testing.

The test approach is described as follows:

RSA_ENC.cpp receives the generated vectors by the CAVS tool as input (plaintext) and encrypts them using a valid key pair (public and private key). As result of this execution RSA_ENC.cpp generates a set of ciphertext. Then, the evaluator executes *OpenSSL FIPS* to generate a new set of ciphertext using the same set of plaintext and key pair.

Once the two different set of ciphertext have been generated, they are swapped in order to perform the decrypt operation. It means that RSA_DEC.cpp receives the set of ciphertext generated by *OpenSSL FIPS* and *OpenSSL FIPS* receives the set of ciphertext generated by RSA_ENC.cpp. Both of these executions generate a set of plaintext, which are equal to each other and equal to the ones generated by the CAVS tool.

The following screenshot shows the RSA_DEC.cpp source code, where the main function is *BCrypt-Decrypt*:

© 2019 Microsoft Corporation. All rights reserved.

```
// Decipher ciphertext to obtain the plaintext
if (!NT_SUCCESS(status = BCryptDecrypt(
    hPriv,
    pC,
    c_length,
    NULL,
    NULL,
    0,
    pK,
    k_length,
    &k_length,
    BCRYPT_PAD_PKCS1)))
{
    wprintf(L"**** Error 0x%x returned by BCryptDecrypt\n", status);
    goto Cleanup;
}

sprintf_s(filename, "plaintext%d_windows.bin", i);
cout << filename << endl;
PrintBytes((BYTE *)pK, k_length);
SavePlaintextToFile((BYTE *)pK, k_length, filename);
```

Finally, the following screenshot shows the main function in RSA_ENC.cpp source code, which it is called *BCryptEncrypt*:

```
// Decipher ciphertext to obtain the plaintext
if (!NT_SUCCESS(status = BCryptEncrypt(
    hPub,
    pK,
    k_length,
    NULL,
    NULL,
    0,
    pC,
    c_length,
    &c_length,
    BCRYPT_PAD_PKCS1)))
{
    wprintf(L"**** Error 0x%x returned by BCryptEncrypt\n", status);
    goto Cleanup;
}

sprintf_s(filename, "ciphertext%d_windows.bin", i);
cout << filename << endl;
PrintBytes((BYTE *)pC, c_length);
SaveCiphertextToFile((BYTE *)pC, c_length, filename);
```

© 2019 Microsoft Corporation. All rights reserved.

7.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- Execute CKM_2_SETUP.sh on Debian 9 machine to:
- Generate the input vectors as plaintext<index>_input.bin from the set of plaintext provided by the CAVS tool.

```
function GenerateInputVectors {
    i=0
    cat $TOOLS_FOLDER/RSADPComponent800_56B.req | grep -m 10 "c ="
        | cut -c5-$((5+245*2)) | while read -r line;
    do
        echo -n $line | xxd -r -p > $input_vectors/plaintext${i}_input.bin
        echo -e "\tGenerated plaintext${i}_input.bin";
        i=$((i+1))
    done
}
```

- Encrypt input vectors with *OpenSSL FIPS*.

```
function Encrypt {
    GeneratePublicPEM;

    ciphertexts=$(ls $OPENSSL_PATH_CIPHERS/ciphertext* 2> /dev/null | wc -l)
    if [ "$ciphertexts" != "0" ]; then
    rm -rf $OPENSSL_PATH_CIPHERS/ciphertext*;
    fi

    for i in {0..9}
    do
        env OPENSSL_FIPS=1 /usr/local/ssl/bin/openssl rsautl -encrypt -pkcs
            -pubin -inkey
            $TOOLS_FOLDER/pub.pem -in $input_vectors/plaintext${i}_input.bin
            -out $OPENSSL_PATH_CIPHERS/ciphertext${i}_openssl.bin
        echo -e "\tCiphertext $i:";
        hexdump -C $OPENSSL_PATH_CIPHERS/ciphertext${i}_openssl.bin
        echo;
    done
}
```

- Prepare a ZIP file called Windows_CKM_2_Test.zip, which will be used on TOE. This file contains:
  - Input vectors available in the tested platform.
  - Folder structure to generate ciphertexts and plaintexts files on Windows.
  - RSA_CRYPT_TOOL.zip: This zip file contains the Windows utilities to crypt and encrypt the previous input vectors (RSA_ENC and RSA_DEC).
  - Generated OpenSSL ciphertext files.
  - A *Powershell* script CKM_2_WIN.ps1 to run the test on Windows platforms.
- Copy and extract Windows_CKM_2_Test.zip on the evaluated platforms:

© 2019 Microsoft Corporation. All rights reserved.

ciphertextsFromOpenSSL	10/22/2018 10:05	File folder	
ciphertextsFromWin	10/22/2018 10:05	File folder	
input_vectors	10/22/2018 10:05	File folder	
Windows	10/22/2018 10:03	File folder	
CKM_2_WIN	10/22/2018 9:56 AM	Windows PowerShell Script	2 KB
RSA_CRYPT_TOOL	10/22/2018 9:56 AM	Compressed (zipped) Folder	501 KB

- Execute CKM_2_WIN.ps1 *Powershell* script. The main steps automatically performed are described below:

```
$input_vectors = ".\input_vectors"
$ciphertextsFromOpenSSL = ".\ciphertextsFromOpenSSL"
$ciphertextsFromWin = ".\ciphertextsFromWin"
$windows_results = ".\Windows\resultsPlaintext"
# Extract file
Expand-Archive ".\RSA_CRYPT_TOOL.zip" -DestinationPath "." -Force

[string]$path = if ($ENV:PROCESSOR_ARCHITECTURE -like "*64") {
    $decrypt = ".\RSA_CRYPT_TOOL\RSA_Decrypt\x64\RSA_Decrypt.exe"
    $encrypt = ".\RSA_CRYPT_TOOL\RSA_Encrypt\x64\RSA_Encrypt.exe"
} else {
    $decrypt = ".\RSA_CRYPT_TOOL\RSA_Decrypt\x86\RSA_Decrypt.exe"
    $encrypt = ".\RSA_CRYPT_TOOL\RSA_Encrypt\x86\RSA_Encrypt.exe"
}

cp $decrypt -Destination $ciphertextsFromOpenSSL
cp $encrypt -Destination $input_vectors

Start-Process $ciphertextsFromOpenSSL\RSA_Decrypt.exe -WorkingDirectory
    $ciphertextsFromOpenSSL -Wait
Start-Process $input_vectors\RSA_Encrypt.exe -WorkingDirectory $input_vectors
    -Wait

cp $input_vectors\ciphertext* -Destination $ciphertextsFromWin
cp $ciphertextsFromOpenSSL\plaintext* -Destination $windows_results

$windows_ckm = "Windows_CKM_2_Results"
mkdir "$windows_ckm"
cp -r $windows_results -Destination $windows_ckm
cp -r $ciphertextsFromWin -Destination $windows_ckm
Compress-Archive -Path ".\$windows_ckm" -DestinationPath
    ".\Windows_CKM_2_Results.zip"
```

- RSA_CRYPT_TOOL.zip is extracted.
- Processor architecture is selected.
- Ciphertexts from OpenSSL are decrypted with \RSA_CRYPT_TOOL\RSA_Decrypt\x64\RSA_Decrypt.exe.
- Input vectors are encrypted with \RSA_CRYPT_TOOL\RSA_Encrypt\x64\RSA_Encrypt.exe.

© 2019 Microsoft Corporation. All rights reserved.

- Plaintexts files and ciphertexts from windows files are compressed in *Windows_CKM_2_Results.zip*.

```
PS C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS> .\CKM_2_WIN.ps1


    Directory: C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        10/23/2018   9:03 AM                Windows_CKM_2_Results


PS C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS> dir


    Directory: C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        10/22/2018  10:05 AM                ciphertextsFromOpenSSL
d-----        10/22/2018  10:05 AM                ciphertextsFromWin
d-----        10/22/2018  10:05 AM                input_vectors
d-----        10/23/2018   9:02 AM                RSA_CRYPT_TOOL
d-----        10/22/2018  10:03 AM                Windows
d-----        10/23/2018   9:03 AM                Windows_CKM_2_Results
------        10/22/2018   9:56 AM           1271 CKM_2_WIN.ps1
------        10/22/2018   9:56 AM         512001 RSA_CRYPT_TOOL.zip
-a----        10/23/2018   9:03 AM           9152 Windows_CKM_2_Results.zip
```

- Copy *Windows_CKM_2_Results.zip* in the Linux machine:
  - Execute *CKM_2.sh* script:
    - *Windows_CKM_2_Results.zip* is extracted
- Ciphertexts from Windows are decrypted with *OpenSSL FIPS*.

```
function Decrypt {
    GeneratePrivatePEM;
    for i in {0..9}
    do
        env OPENSSL_FIPS=1 /usr/local/ssl/bin/openssl rsautl -decrypt -inkey
            $TOOLS_FOLDER/priv.pem
            -in $OPENSSL_PATH_CIPHER_WIN/ciphertext${i}_windows.bin
                > $OPENSSL_PATH_RESULTS/plaintext${i}_openssl.bin
        echo -e "\tPlaintext $i:";
        hexdump -C $OPENSSL_PATH_RESULTS/plaintext${i}_openssl.bin
        echo;
    done
}
```

- SHA256 hashes of OpenSSL plaintext files and Windows plaintext files are compared between each other.

© 2019 Microsoft Corporation. All rights reserved.

7.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

An example of the obtained results are shown. Given the following plaintext,

```
[mod = 2048]
        COUNT = 0
        c = 1625cdda17e2e8cfa247ef5f43ac25cfff9550de66e4199d164db581800bc5b7efd17bb7c604ff1c082711b31e97e
fd2d57879eba7ef9360ee91cb992f1c3c551465b13dba920ebfae89ec1902a984b44cb7ceb4f11beacda44cbe582989e1a65fcab7
947df6de5f9c56c9ba764aab1e6d3a9762cfd0e58b5864d7e5f8b87d53d7e6d5e3979637a5d974f0be02e22eb83fe91c1cc6764d4
35fc35e692313d70384a253528edce1396dcda94c2ed9d329caada40b6075ea9cc24f9b0b3300b95f6fe7780bec2fa3c260483c8c
8397631a166c78562435843f88f2d5f4afe9e90e9f9b8e3338f646e9baccf0a78884a5322c10e1279a1fe3d3b993564b1f3530b9

        COUNT = 1
        c = 1cd94856d065cd8d1805c16f7529ea7bf5f2ee938adf65b8bba831169a6a1240f60575bcd1c28dc2143aa5418273b
52eeb4b33d2a21a2bfb319392924aebd816013ebc16392e64f45ccdade950bdc049a695e60814879d7e2540eb672f10f22e864bc5
6befef7aaa5786806ec54e33758e98ae3ec86ee0c90078e39e30218810ebf537c3db4df66c6d2fc7d8cb7037300b1b347ba5171f6
42b03223d3465c4459a04ba26e59200c2068e292f4969f40085e81fc795c980d80ca9514f2525cc54f940f89df504c2358a6d4f7c
2682917064c5182a80cd2bab0d3218d9eaa9c8c57270962ff3ed614c6192e0c74c033605107bc7ad4c6c496895651ee56b39e121
```

The evaluator has obtained the following ciphertext using *OpenSSL FIPS*:

© 2019 Microsoft Corporation. All rights reserved.

```
        Ciphertext 0:
00000000  17 18 c4 0e 0a 4d 14 68  93 ba 0a 94 42 31 40 6f  |.....M.h....B1@o|
00000010  de cd b9 ef 62 c1 ab b9  d7 5f 7e c3 36 be fe 08  |....b...._~.6...|
00000020  a4 a0 b0 45 80 3e 68 a6  d7 41 03 6d 97 fb 76 4e  |...E.>h..A.m..vN|
00000030  af 1a bd 68 c5 04 6b ca  a5 1b fa e8 97 64 69 e2  |...h..k......di.|
00000040  e5 21 22 e4 fd 78 fb 9b  13 30 f6 b0 47 92 59 b8  |.!"..x...0..G.Y.|
00000050  e7 44 48 da 4d 11 50 04  98 4a b6 d7 2b 92 6e df  |.DH.M.P..J..+.n.|
00000060  b2 4c 42 89 89 de 7f d6  14 4f 5a ae 56 a5 66 cc  |.LB......OZ.V.f.|
00000070  50 4e 1d b5 61 ae 3c 52  93 13 2e 34 41 8f 94 79  |PN..a.<R...4A..y|
00000080  1f 20 67 28 aa 96 b8 9f  b7 24 85 d2 b1 e4 ba b7  |. g(.....$......|
00000090  4b 61 cf 4f b0 33 cd 09  00 fb 4a 8b 5d 14 73 a6  |Ka.O.3....J.].s.|
000000a0  c4 93 43 19 bc 79 ce 57  0b 2f b4 ff 1b 21 ee 31  |..C..y.W./...!.1|
000000b0  d6 9a 9c f7 78 40 34 c3  5c 31 1a 7b 38 ed 8c 6c  |....x@4.\1.{8..l|
000000c0  53 f9 c1 dc 8d 3f 6f 3f  b4 28 56 d2 3f 82 e8 61  |S....?o?.(V.?..a|
000000d0  71 65 ae 45 b3 28 68 b3  ef 70 ea 0c 69 6b 0e a1  |qe.E.(h..p..ik..|
000000e0  9a f0 bb 32 34 2a 75 6a  3c 1c 3d f6 92 92 d8 2c  |...24*uj<.=....,|
000000f0  e3 65 5d 4d 00 08 db f7  47 b4 e3 b8 b0 ca 51 be  |.e]M....G.....Q.|
00000100

        Ciphertext 1:
00000000  15 5c 82 a2 be f4 30 21  3f cc 41 88 34 59 cb 4d  |.\....0!?.A.4Y.M|
00000010  da 29 a6 09 df 42 a0 8f  c6 50 ec 99 75 9b 94 7a  |.)...B...P..u..z|
00000020  ca 7f bf 3c 7f d8 be 98  7b c3 f4 03 c0 48 2b 08  |...<....{....H+.|
00000030  34 11 9e ec 8e 84 34 8c  f1 5c 63 a8 a5 7a 70 ae  |4.....4..\c..zp.|
00000040  5f 5b 20 8e c1 a1 c0 fa  6f e0 8e e6 aa 3e 37 c8  |_[ .....o....>7.|
00000050  98 e9 e7 19 64 a2 5b c5  d1 8d 47 1c 2c 3b 8d d8  |....d.[...G.,;..|
00000060  4d 25 22 9b a8 bd 0b 59  4e d6 eb 31 11 f0 42 7d  |M%"....YN..1..B}|
00000070  28 0c 68 fb 0f 6c 49 eb  5d 08 47 f5 4e 67 6c c8  |(.h..lI.].G.Ngl.|
00000080  d9 04 e3 c3 3a 1b 20 74  a6 ba 48 56 67 81 04 a3  |....:. t..HVg...|
00000090  a6 d3 12 8b 81 f4 11 ae  6c 64 74 ae f3 06 b3 8b  |........ldt.....|
000000a0  c4 be 69 d9 05 0c 7c c6  6e 7c 68 79 03 59 41 ac  |..i...|.n|hy.YA.|
000000b0  78 7f 26 53 f6 59 d3 79  a3 7e 18 a1 9a 66 4a 51  |x.&S.Y.y.~...fJQ|
000000c0  f8 37 b7 00 f9 ca 07 ac  fe a9 f9 a8 1e 4d 6e 52  |.7...........MnR|
000000d0  61 3c 57 11 dc bd 29 fb  1a ee a0 0f db e2 a8 18  |a<W...).........|
000000e0  96 21 3d f3 7c 61 b4 22  42 82 7b c9 c4 f4 19 30  |.!=.|a."B.{....0|
000000f0  15 ac a3 d2 17 82 d4 4a  8f bc 6a 5c 1b e4 7b 4d  |.......J..j\..{M|
00000100
```

Using RSA_CRYPT_TOOL\RSA\Encrypt\x64\RSA_Encrypt.exe, the ciphertext generated are the following:

© 2019 Microsoft Corporation. All rights reserved.

C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS\input_vectors\RSA_Encrypt.exe

```
Starting...
ciphertext0_windows.bin
0xce, 0x90, 0x16, 0x35, 0x18, 0xaa, 0x07, 0x0e, 0xfe, 0x71,
0xf1, 0x85, 0xb8, 0xfc, 0x4d, 0xf1, 0x61, 0x16, 0x2b, 0x91,
0x51, 0x3b, 0x59, 0x83, 0xb2, 0x2b, 0xf1, 0x49, 0x3a, 0xb7,
0x8c, 0x84, 0xf0, 0x84, 0x19, 0x32, 0xab, 0x32, 0xc9, 0x7b,
0x51, 0x8e, 0xd8, 0x0d, 0x8d, 0x86, 0x8e, 0x3c, 0x4c, 0xc8,
0xff, 0x8a, 0xc8, 0x4d, 0x14, 0x0d, 0x0e, 0x63, 0x6b, 0x40,
0x65, 0xd2, 0x73, 0xd9, 0xfc, 0xf2, 0x40, 0x88, 0x9a, 0x30,
0x99, 0x11, 0xe7, 0x73, 0xb8, 0xe6, 0x7a, 0x01, 0xe3, 0x4f,
0x62, 0x44, 0xe3, 0xaa, 0xd3, 0xe4, 0x4f, 0xe2, 0xf8, 0x45,
0x36, 0x78, 0xd3, 0x7c, 0x92, 0x33, 0x47, 0x1a, 0xb5, 0x41,
0x2f, 0x60, 0x5e, 0x1c, 0xa0, 0x1b, 0xde, 0xe2, 0xa3, 0x3b,
0x5c, 0x09, 0x32, 0x46, 0xe2, 0xfe, 0xf9, 0xe6, 0x38, 0x49,
0x4d, 0xa0, 0xa0, 0x29, 0xec, 0xe7, 0x81, 0x74, 0x6c, 0xe1,
0x6a, 0xde, 0x10, 0xcb, 0xcd, 0x06, 0xf4, 0x0f, 0x18, 0x61,
0xdb, 0xb1, 0xfa, 0x10, 0xe0, 0xac, 0xdb, 0x2f, 0x69, 0x56,
0x81, 0xff, 0x0d, 0x03, 0x6e, 0x74, 0x4c, 0x04, 0x5f, 0xb9,
0x9c, 0x0b, 0x7d, 0x9b, 0x76, 0xa1, 0x74, 0x9a, 0x51, 0x2e,
0x16, 0xe1, 0x48, 0x3e, 0xd0, 0x4e, 0xd5, 0x47, 0xe8, 0x57,
0x57, 0x0d, 0xbe, 0xab, 0x5f, 0x4d, 0x14, 0x4a, 0x68, 0x64,
0x3e, 0x7f, 0x96, 0xae, 0xa6, 0xc1, 0x5d, 0x73, 0x2a, 0xcc,
0x49, 0xc2, 0x71, 0xc9, 0x2f, 0xe6, 0xcd, 0xfd, 0x16, 0xe7,
0xb4, 0x60, 0xa1, 0xcd, 0x32, 0x22, 0x40, 0x64, 0xc6, 0xc7,
0xed, 0x71, 0x5c, 0x9a, 0xa8, 0x98, 0x3d, 0x1f, 0xa7, 0x1f,
0x57, 0xfb, 0x09, 0x37, 0xe9, 0xb6, 0xb6, 0x19, 0xd0, 0x6f,
0x17, 0x9e, 0xd0, 0xdd, 0xa5, 0x8a, 0xf8, 0xeb, 0xe3, 0xe6,
0xf4, 0xcd, 0xa6, 0x2b, 0x1c, 0xa0,
ciphertext1_windows.bin
0x00, 0x88, 0x0a, 0x74, 0x39, 0x7d, 0x13, 0x03, 0x98, 0x9d,
0x9c, 0xa6, 0x5e, 0xd3, 0xf7, 0xf4, 0xa6, 0x88, 0x89, 0x8a,
0xd6, 0xe4, 0x3e, 0xae, 0x36, 0xb1, 0x5b, 0x38, 0xbc, 0x04,
0xa1, 0xa1, 0x06, 0xa2, 0xb5, 0x12, 0x40, 0x20, 0x52, 0xf4,
0xdb, 0x07, 0xab, 0x98, 0xf5, 0xe4, 0xfd, 0x34, 0x67, 0xdc,
0x83, 0x39, 0x27, 0xad, 0x95, 0x4e, 0x65, 0xbd, 0xbf, 0xe0,
0x86, 0xdd, 0x32, 0xd9, 0x64, 0x1e, 0xcf, 0x50, 0x4c, 0x82,
0x83, 0xec, 0x4b, 0x7d, 0x55, 0x7a, 0x9b, 0x75, 0xd5, 0x46,
0x7d, 0x12, 0x9e, 0x3a, 0xa1, 0x4d, 0x45, 0x31, 0x2c, 0x14,
0x8e, 0xb0, 0xda, 0x60, 0xbf, 0xab, 0x06, 0x44, 0x5f, 0x4c,
0x52, 0x7e, 0x4e, 0x27, 0x26, 0x3a, 0x0d, 0x5a, 0xa7, 0x7f,
0x12, 0x1d, 0x38, 0x52, 0x72, 0x31, 0xb0, 0x19, 0x78, 0x77,
0xf4, 0xb6, 0xed, 0x57, 0x09, 0x76, 0x2f, 0x95, 0x82, 0x00,
0xfa, 0xc4, 0xde, 0x1e, 0x6f, 0x65, 0x13, 0x36, 0xae, 0xda,
0x99, 0x35, 0x6a, 0x6d, 0x77, 0x7d, 0xd3, 0xf9, 0x5d, 0xc1,
0x24, 0xed, 0xee, 0xc4, 0x3f, 0xc3, 0xf9, 0x58, 0xf8, 0xc5,
0xe3, 0xf0, 0x25, 0xbe, 0x6b, 0xdd, 0x07, 0xff, 0xe3, 0x72,
0x4e, 0x02, 0x6a, 0x15, 0x6d, 0xf7, 0x8e, 0x33, 0xf1, 0xf5,
0xc2, 0xb0, 0xa3, 0xf7, 0x66, 0x83, 0xba, 0x05, 0x1a, 0x98,
0x9f, 0xea, 0x40, 0xae, 0xcf, 0xba, 0x15, 0x02, 0x61, 0x75,
0x8c, 0x04, 0x96, 0x3e, 0x5e, 0xf3, 0xeb, 0xb8, 0x04, 0x8c,
0x18, 0xe8, 0xab, 0x73, 0xcb, 0x42, 0x77, 0x5d, 0x0e, 0xaf,
0xd7, 0x92, 0x23, 0x40, 0x94, 0x0d, 0x33, 0xaa, 0x8b, 0x75,
0x50, 0x6b, 0xe1, 0x11, 0xdb, 0x46, 0xc6, 0x2f, 0xcc, 0x2b,
0xc0, 0xe3, 0x3e, 0x4d, 0xd8, 0x60, 0xe5, 0xf1, 0x1b, 0x06,
0x36, 0xd6, 0x2b, 0x56, 0xdc, 0x38,
```

© 2019 Microsoft Corporation. All rights reserved.

After that, sets of ciphertext are swapped, and the evaluator obtained the following plaintext by using *OpenSSL FIPS*:

```
        Plaintext 0:
00000000  16 25 cd da 17 e2 e8 cf  a2 47 ef 5f 43 ac 25 cf  |.%.......G._C.%.|
00000010  ff 95 50 de 66 e4 19 9d  16 4d b5 81 80 0b c5 b7  |..P.f....M......|
00000020  ef d1 7b b7 c6 04 ff 1c  08 27 11 b3 1e 97 ef d2  |..{......'......|
00000030  d5 78 79 eb a7 ef 93 60  ee 91 cb 99 2f 1c 3c 55  |.xy....`..../.<U|
00000040  14 65 b1 3d ba 92 0e bf  ae 89 ec 19 02 a9 84 b4  |.e.=............|
00000050  4c b7 ce b4 f1 1b ea cd  a4 4c be 58 29 89 e1 a6  |L........L.X)...|
00000060  5f ca b7 94 7d f6 de 5f  9c 56 c9 ba 76 4a ab 1e  |_...}.._.V..vJ..|
00000070  6d 3a 97 62 cf d0 e5 8b  58 64 d7 e5 f8 b8 7d 53  |m:.b....Xd....}S|
00000080  d7 e6 d5 e3 97 96 37 a5  d9 74 f0 be 02 e2 2e b8  |......7..t......|
00000090  3f e9 1c 1c c6 76 4d 43  5f c3 5e 69 23 13 d7 03  |?....vMC_.^i#...|
000000a0  84 a2 53 52 8e dc e1 39  6d cd a9 4c 2e d9 d3 29  |..SR...9m..L...)|
000000b0  ca ad a4 0b 60 75 ea 9c  c2 4f 9b 0b 33 00 b9 5f  |....`u...O..3.._|
000000c0  6f e7 78 0b ec 2f a3 c2  60 48 3c 8c 83 97 63 1a  |o.x../..`H<...c.|
000000d0  16 6c 78 56 24 35 84 3f  88 f2 d5 f4 af e9 e9 0e  |.lxV$5.?........|
000000e0  9f 9b 8e 33 38 f6 46 e9  ba cc f0 a7 88 84 a5 32  |...38.F........2|
000000f0  2c 10 e1 27 9a                                    |,..'.|
000000f5

        Plaintext 1:
00000000  1c d9 48 56 d0 65 cd 8d  18 05 c1 6f 75 29 ea 7b  |..HV.e.....ou).{|
00000010  f5 f2 ee 93 8a df 65 b8  bb a8 31 16 9a 6a 12 40  |......e...1..j.@|
00000020  f6 05 75 bc d1 c2 8d c2  14 3a a5 41 82 73 b5 2e  |..u......:.A.s..|
00000030  eb 4b 33 d2 a2 1a 2b fb  31 93 92 92 4a eb d8 16  |.K3...+.1...J...|
00000040  01 3e bc 16 39 2e 64 f4  5c cd ad e9 50 bd c0 49  |.>..9.d.\...P..I|
00000050  a6 95 e6 08 14 87 9d 7e  25 40 eb 67 2f 10 f2 2e  |.......~%@.g/...|
00000060  86 4b c5 6b ef ef 7a aa  57 86 80 6e c5 4e 33 75  |.K.k..z.W..n.N3u|
00000070  8e 98 ae 3e c8 6e e0 c9  00 78 e3 9e 30 21 88 10  |...>.n...x..0!..|
00000080  eb f5 37 c3 db 4d f6 6c  6d 2f c7 d8 cb 70 37 30  |..7..M.lm/...p70|
00000090  0b 1b 34 7b a5 17 1f 64  2b 03 22 3d 34 65 c4 45  |..4{...d+."=4e.E|
000000a0  9a 04 ba 26 e5 92 00 c2  06 8e 29 2f 49 69 f4 00  |...&......)/Ii..|
000000b0  85 e8 1f c7 95 c9 80 d8  0c a9 51 4f 25 25 cc 54  |..........QO%%.T|
000000c0  f9 40 f8 9d f5 04 c2 35  8a 6d 4f 7c 26 82 91 70  |.@.....5.mO|&..p|
000000d0  64 c5 18 2a 80 cd 2b ab  0d 32 18 d9 ea a9 c8 c5  |d..*..+..2......|
000000e0  72 70 96 2f f3 ed 61 4c  61 92 e0 c7 4c 03 36 05  |rp./..aLa...L.6.|
000000f0  10 7b c7 ad 4c                                    |.{..L|
000000f5
```

And the following plaintext have been obtained using RSA_CRYPT_TOOL\RSA\Decrypt\x64\RSA_Decrypt.exe on the evaluated platforms:

© 2019 Microsoft Corporation. All rights reserved.

C:\Users\evaluador\Desktop\Windows_CKM_2_Test\TEST_WINDOWS\ciphertextsFromOpenSSL\RSA_Decrypt.exe

```
Starting...
plaintext0_windows.bin
0x16, 0x25, 0xcd, 0xda, 0x17, 0xe2, 0xe8, 0xcf, 0xa2, 0x47,
0xef, 0x5f, 0x43, 0xac, 0x25, 0xcf, 0xff, 0x95, 0x50, 0xde,
0x66, 0xe4, 0x19, 0x9d, 0x16, 0x4d, 0xb5, 0x81, 0x80, 0x0b,
0xc5, 0xb7, 0xef, 0xd1, 0x7b, 0xb7, 0xc6, 0x04, 0xff, 0x1c,
0x08, 0x27, 0x11, 0xb3, 0x1e, 0x97, 0xef, 0xd2, 0xd5, 0x78,
0x79, 0xeb, 0xa7, 0xef, 0x93, 0x60, 0xee, 0x91, 0xcb, 0x99,
0x2f, 0x1c, 0x3c, 0x55, 0x14, 0x65, 0xb1, 0x3d, 0xba, 0x92,
0x0e, 0xbf, 0xae, 0x89, 0xec, 0x19, 0x02, 0xa9, 0x84, 0xb4,
0x4c, 0xb7, 0xce, 0xb4, 0xf1, 0x1b, 0xea, 0xcd, 0xa4, 0x4c,
0xbe, 0x58, 0x29, 0x89, 0xe1, 0xa6, 0x5f, 0xca, 0xb7, 0x94,
0x7d, 0xf6, 0xde, 0x5f, 0x9c, 0x56, 0xc9, 0xba, 0x76, 0x4a,
0xab, 0x1e, 0x6d, 0x3a, 0x97, 0x62, 0xcf, 0xd0, 0xe5, 0x8b,
0x58, 0x64, 0xd7, 0xe5, 0xf8, 0xb8, 0x7d, 0x53, 0xd7, 0xe6,
0xd5, 0xe3, 0x97, 0x96, 0x37, 0xa5, 0xd9, 0x74, 0xf0, 0xbe,
0x02, 0xe2, 0x2e, 0xb8, 0x3f, 0xe9, 0x1c, 0x1c, 0xc6, 0x76,
0x4d, 0x43, 0x5f, 0xc3, 0x5e, 0x69, 0x23, 0x13, 0xd7, 0x03,
0x84, 0xa2, 0x53, 0x52, 0x8e, 0xdc, 0xe1, 0x39, 0x6d, 0xcd,
0xa9, 0x4c, 0x2e, 0xd9, 0xd3, 0x29, 0xca, 0xad, 0xa4, 0x0b,
0x60, 0x75, 0xea, 0x9c, 0xc2, 0x4f, 0x9b, 0x0b, 0x33, 0x00,
0xb9, 0x5f, 0x6f, 0xe7, 0x78, 0x0b, 0xec, 0x2f, 0xa3, 0xc2,
0x60, 0x48, 0x3c, 0x8c, 0x83, 0x97, 0x63, 0x1a, 0x16, 0x6c,
0x78, 0x56, 0x24, 0x35, 0x84, 0x3f, 0x88, 0xf2, 0xd5, 0xf4,
0xaf, 0xe9, 0xe9, 0x0e, 0x9f, 0x9b, 0x8e, 0x33, 0x38, 0xf6,
0x46, 0xe9, 0xba, 0xcc, 0xf0, 0xa7, 0x88, 0x84, 0xa5, 0x32,
0x2c, 0x10, 0xe1, 0x27, 0x9a,
plaintext1_windows.bin
0x1c, 0xd9, 0x48, 0x56, 0xd0, 0x65, 0xcd, 0x8d, 0x18, 0x05,
0xc1, 0x6f, 0x75, 0x29, 0xea, 0x7b, 0xf5, 0xf2, 0xee, 0x93,
0x8a, 0xdf, 0x65, 0xb8, 0xbb, 0xa8, 0x31, 0x16, 0x9a, 0x6a,
0x12, 0x40, 0xf6, 0x05, 0x75, 0xbc, 0xd1, 0xc2, 0x8d, 0xc2,
0x14, 0x3a, 0xa5, 0x41, 0x82, 0x73, 0xb5, 0x2e, 0xeb, 0x4b,
0x33, 0xd2, 0xa2, 0x1a, 0x2b, 0xfb, 0x31, 0x93, 0x92, 0x92,
0x4a, 0xeb, 0xd8, 0x16, 0x01, 0x3e, 0xbc, 0x16, 0x39, 0x2e,
0x64, 0xf4, 0x5c, 0xcd, 0xad, 0xe9, 0x50, 0xbd, 0xc0, 0x49,
0xa6, 0x95, 0xe6, 0x08, 0x14, 0x87, 0x9d, 0x7e, 0x25, 0x40,
0xeb, 0x67, 0x2f, 0x10, 0xf2, 0x2e, 0x86, 0x4b, 0xc5, 0x6b,
0xef, 0xef, 0x7a, 0xaa, 0x57, 0x86, 0x80, 0x6e, 0xc5, 0x4e,
0x33, 0x75, 0x8e, 0x98, 0xae, 0x3e, 0xc8, 0x6e, 0xe0, 0xc9,
0x00, 0x78, 0xe3, 0x9e, 0x30, 0x21, 0x88, 0x10, 0xeb, 0xf5,
0x37, 0xc3, 0xdb, 0x4d, 0xf6, 0x6c, 0x6d, 0x2f, 0xc7, 0xd8,
0xcb, 0x70, 0x37, 0x30, 0x0b, 0x1b, 0x34, 0x7b, 0xa5, 0x17,
0x1f, 0x64, 0x2b, 0x03, 0x22, 0x3d, 0x34, 0x65, 0xc4, 0x45,
0x9a, 0x04, 0xba, 0x26, 0xe5, 0x92, 0x00, 0xc2, 0x06, 0x8e,
0x29, 0x2f, 0x49, 0x69, 0xf4, 0x00, 0x85, 0xe8, 0x1f, 0xc7,
0x95, 0xc9, 0x80, 0xd8, 0x0c, 0xa9, 0x51, 0x4f, 0x25, 0x25,
0xcc, 0x54, 0xf9, 0x40, 0xf8, 0x9d, 0xf5, 0x04, 0xc2, 0x35,
0x8a, 0x6d, 0x4f, 0x7c, 0x26, 0x82, 0x91, 0x70, 0x64, 0xc5,
0x18, 0x2a, 0x80, 0xcd, 0x2b, 0xab, 0x0d, 0x32, 0x18, 0xd9,
0xea, 0xa9, 0xc8, 0xc5, 0x72, 0x70, 0x96, 0x2f, 0xf3, 0xed,
0x61, 0x4c, 0x61, 0x92, 0xe0, 0xc7, 0x4c, 0x03, 0x36, 0x05,
0x10, 0x7b, 0xc7, 0xad, 0x4c,
```

© 2019 Microsoft Corporation. All rights reserved.

To check that both obtained plaintext match between them and with the one provided by the CAVS tool, SHA256 hashes are compared between plaintext files:

```
[-] Comparing SHA256 hashes between plaintext0_windows.bin and plaintext0_openssl.bin...
        plaintext0_windows.bin: ef69ec52bb770bf061fe54d7254c2e74423bd05d41182cc377788b65e614b414
        plaintext0_openssl.bin: ef69ec52bb770bf061fe54d7254c2e74423bd05d41182cc377788b65e614b414

[-] Comparing SHA256 hashes between plaintext1_windows.bin and plaintext1_openssl.bin...
        plaintext1_windows.bin: 1646b0fc1b70776be2fdf65770ae36cbf92d9a7b4e01a9d2aeebdc41f22db23b
        plaintext1_openssl.bin: 1646b0fc1b70776be2fdf65770ae36cbf92d9a7b4e01a9d2aeebdc41f22db23b
```

```
-----------------------------------------------------------------------
|                              Test CKM_2                             |
-----------------------------------------------------------------------
WINDOWS                        OPENSSL                        SAME FILE
plaintext0_windows.bin         plaintext0_openssl.bin         PASS
plaintext1_windows.bin         plaintext1_openssl.bin         PASS
```

7.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfilment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

7.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_CKM.2.1.

© 2019 Microsoft Corporation. All rights reserved.

8 FCS_CKM.4.1

The assurance activity for the **FCS_CKM.4.1** requirement is stated as follows:

> The evaluator examines the TSS to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in non-volatile memory) and how they are overwritten.
>
> The evaluator will check to ensure the TSS lists each type of key that is stored in in non-volatile memory, and identifies how the TOE interacts with the underlying platform to manage keys (e.g., store, retrieve, destroy). The description includes details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys (e.g., file system APIs, platform key store APIs).
>
> If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.
>
> The evaluator will check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.
>
> **Operational Guidance**
>
> There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator will check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information. The evaluator will check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer and how such situations can be avoided or mitigated if possible.
>
> Some examples of what is expected to be in the documentation are provided here.
>
> When the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, to mitigate this the drive should support the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.
>
> Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. To reduce this risk, the operating system and file system of the OE should support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion. If a RAID array is being used, only set-ups that support TRIM are utilized. If the drive is connected via PCI-Express, the operating system supports TRIM over that channel.

© 2019 Microsoft Corporation. All rights reserved.

The drive should be healthy and contains minimal corrupted data and should be end of lifed before a significant amount of damage to drive health occurs, this minimizes the risk that small amounts of potentially recoverable data may remain in damaged areas of the drive.

- **Test 1:** Applied to each key held as in volatile memory and subject to destruction by overwrite by the TOE (whether or not the value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator will:

  1. Record the value of the key in the TOE subject to clearing.
  2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
  3. Cause the TOE to clear the key.
  4. Cause the TOE to stop the execution but not exit.
  5. Cause the TOE to dump the entire memory of the TOE into a binary file.
  6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.

  Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

- **Test 2:** Applied to each key help in non-volatile memory and subject to destruction by the TOE. The evaluator will use special tools (as needed), provided by the TOE developer if necessary, to ensure the tests function as intended.

  1. Identify the purpose of the key and what access should fail when it is deleted. (e.g. the data encryption key being deleted would cause data decryption to fail.)
  2. Cause the TOE to clear the key.
  3. Have the TOE attempt the functionality that the cleared key would be necessary for.

  The test succeeds if step 3 fails.

- **Test 3:**

  The following tests apply only to selection a), since the TOE in thisinstance has more visibility into what is happening within theunderlying platform (e.g., a logical view of the media). In selection b),the TOE has no visibility into the inner workings and completely relies on the underlying platform, so there is no reason to test the TOE beyond test 2.

  For selection a), the following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern.

  Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator will use a tool that provides a logical view of the media (e.g., MBR file system):

© 2019 Microsoft Corporation. All rights reserved.

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.

- **Test 4:** Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator will use a tool that provides a logical view of the media:

  1. Record the logical storage location of the key in the TOE subject to clearing.
  2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
  3. Cause the TOE to clear the key.
  4. Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

  The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

8.1 Documentation Review activity

8.1.1 Findings

According to the SFR definition, only the key destruction method for volatile memory is selected. The evaluator has reviewed the information provided in TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes the zeroization method for volatile memory. It is carried out using the *RtlSecureZeroMemory* function which overwrites with zeros the memory space indicated (the source code of this function is provided below). The TSS also states that:

> *Windows overwrites each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers for the key or plaintext password which was typed by the user, that is included in the path of such data).*
>
> [...]
>
> *The following table describes the keys and secrets used for **networking and data protection**; when these ephemeral keys or secrets are no longer needed for a network session, due to either normal end of the session or abnormal termination, or after protecting sensitive data using DPAPI, they are deleted:*
>
> [...]

Key	Description
Symmetric encryption/decryption keys	Keys used for AES (FIPS 197) encryption/decryption for IPsec ESP, TLS, Wi-Fi.

© 2019 Microsoft Corporation. All rights reserved.

Key	Description
HMAC keys	Keys used for HMAC-SHA1, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512 (FIPS 198-1) as part of IPsec
Asymmetric ECDSA Public Keys	Keys used for the verification of ECDSA digital signatures (FIPS 186-4) for IPsec traffic and peer authentication.
Asymmetric ECDSA Private Keys	Keys used for the calculation of ECDSA digital signatures (FIPS 186-4) for IPsec traffic and peer authentication.
Asymmetric RSA Public Keys	Keys used for the verification of RSA digital signatures (FIPS 186-4) for IPsec, TLS, Wi-Fi and signed product updates.
Asymmetric RSA Private Keys	Keys used for the calculation of RSA digital signatures (FIPS 186-4) for IPsec, TLS, and Wi-Fi as well as TPM-based health attestations. The key size can be 2048 or 3072 bits.
Asymmetric DSA Private Keys	Keys used for the calculation of DSA digital signatures (FIPS 186-4) for IPsec and TLS. The key size can be 2048 or 3072 bits.
Asymmetric DSA Public Keys	Keys used for the verification of DSA digital signatures (FIPS 186-4) for IPsec and TLS. The key size can be 2048 or 3072 bits.
DH Private and Public values	Private and public values used for Diffie-Hellman key establishment for TLS.
ECDH Private and Public values	Private and public values used for EC Diffie-Hellman key establishment for TLS.
DPAPI HMAC	160-bit MAC key used by DPAPI to generate the AES Master Key based on the user's password.
DPAPI master secret	512-bit random value used by DPAPI.
DPAPI master AES key	256-bit encryption key that protects the DPAPI master secret.
DPAPI AES key	256-bit encryption key used by DPAPI.
DRBG seed	512-bit seed for the main DRBG, zeroized during reseeding.

These keys are generated when they are needed using an approved RNG algorithm (according to *SP 800-90*) as part of the generation process.

In addition, neither the TSS nor the operational guidance identify any configuration or circumstances that do not strictly conform to the key destruction requirement.

8.1.2 Verdict

The evaluator considers that the TSS identifies for all volatile keys involved in the cryptographic processes under evaluation (*networking and DPAPI*), when are generated, the purpose of its usage and how are zeroized when they are no longer needed using the *RtlSecureZeroMemory* internal function. This information is in the TSS sections **6.2.1 Cryptographic Algorithms and Operations** and **6.2.4 Protecting Data with DPAPI**.

© 2019 Microsoft Corporation. All rights reserved.

Moreover, neither the TSS nor the ***Operational Guidance*** document, identify any configuration or circumstances that do not strictly conform to the key destruction requirement.

Due to this, the evaluator considers that the information provided in the TSS is enough and the requirements established in the assurance activity section are fulfilled. Hence, the **PASS** verdict is assigned to the documentation review activity.

8.2 Test Activity

8.2.1 Test 1

To do this test, the evaluator shall have access to the source code as well as access to the symbols to be able to debug the zeroization functions.

Due to this fact, this test shall be performed at Microsoft facilities.

8.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A Web Server supporting TLS1.2 connections is available (*IIS*).
- A Web Client is available on the TOE (e.g. *Internet Explorer*).
- A Kernel debugging environment is configured.

8.2.1.2 Procedure

8.2.1.2.1 Source code review

Review the source code regarding the zeroization functionality, checking that:

- For volatile memory, the destruction is executed by a single direct overwrite consisting of zeros.

For volatile memory keys:

- Perform a cryptographic operation in debug mode.
- Check how the key is generated.
- Check the memory value of the key variable before performing the cryptographic operation.
- Perform the cryptographic operation and the corresponding zeroization mechanism
- Check that the memory value of the key variable is overwritten with zeros
- Dump the TOE memory
- Search the memory value string (obtained on the second step) within the dumped file.
- Repeat the previous steps, using each hardware platform defined in section **Evaluated Platforms**.

© 2019 Microsoft Corporation. All rights reserved.

8.2.1.3 Results

8.2.1.3.1 Source code review (networking keys)

The evaluator has verified how the key zeroization functions are called in one way or another depending on the cipher suite used, as it can be shown in the following screenshots:

8.2.1.3.2 Zeroization of AES keys

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

8.2.1.3.3 Zeroization of HMAC keys

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.4 Zeroization of ECC keys

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.5 Zeroization of DH keys

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.6 Zeroization of DSA keys

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.7 Zeroization of RSA keys

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.8 Zeroization of DBRG seed

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.9 Zeroization functions

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

8.2.1.3.10 SYMCRYPT zeroization functions

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

© 2019 Microsoft Corporation. All rights reserved.

Source code:

This picture intentionally left blank

Function name: [**Text intentionally left blank**]

Source code:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Function name: **[Text intentionally left blank]**

Source code:

This picture intentionally left blank

Function name: **[Text intentionally left blank]**

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.11 Source code review (*"secure storage"* keys)

The following screenshots show functions through which some *"secure storage"* DPAPI keys are zeroized:

8.2.1.3.12 PBKDF2 HMAC key zeroization

The key used during the PBKDF2 HMAC process in the code is a buffer referred to as the **[Text intentionally left blank]** variable. This key is the length of an SHA1 hash (20 bytes). The following image is a screenshot of this key when is zeroized:

Function name: **[Text intentionally left blank]**

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.13 Zeroization of the symmetric key used to encrypt/decrypt the Master Key (DPAPI master AES key)

The symmetric key generated from the PBKDF2 process is a 256 bit AES key that is used to encrypt/decrypt the Master Key. The following image is the screenshot of the decryption of the master key along with the destruction of the 256 bit AES key that is derived from the PBKDF2 process.

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.14 Zeroization of the symmetric key derived from the Master Key (DPAPI AES key)

The DPAPI protected blob is protected with a 256 bit AES key derived from the Master Key. The screenshot below shows the decryption using this AES key and the subsequent zeroization:

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.15 Zeroization of the Master Key

Once the use of the Master Key is finished, then it is zeroized. The screenshot below shows the zeroization of the Master Key:

Function name: [**Text intentionally left blank**]

Source code:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.16 Explanation of BCryptDestroyKey and RtlSecureZeroMemory relationship

Function name: [**Text intentionally left blank**]

Call stack:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

8.2.1.3.17 Testing volatile memory keys (networking keys)

Scenario 1-1: Windows 10 October 2018 Update - HTTPS (TLS 1.2) communication with Symmetric cipher suite (AES)

Testing platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Test case:

The server has configured a web page only available for TLS connections.

The TOE has accessed to the web page of the server via web browser (e.g. _Internet Explorer*) forcing a TLS connection using a symmetric cipher suite (AES).

As part of the TLS protocols two symmetric keys are generated:

- Inbound key
- Outbound key

These keys are in both sides, client and server. The following images show the values of the keys structure (the key is in one contiguous memory location):

Key structure:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Once the keys are obtained, the client closes the connection, forcing the zeroization of the keys (given that the keys are only available in memory while the client session is alive).

The next image show how, when the session is closed, the [**Text intentionally left blank**] function overwrites with zeros the structure which contains the keys:

Zeroized key structure:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

In addition the following image shows the stack after performing the whole process:

Call stack:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Finally the memory is dumped into a *.dmp* file. This file is opened with a Hex editor (e.g. *HxD*), where the keys shown above have been searched without success.

Scenario 1-2: Windows 10 October 2018 Update - Pairwise consistency test with ECDH keys

Testing platforms:

- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

Test case:

Every time that the TOE is powered up, the TOE tests that the cryptographic algorithms still operate correctly. This is done by performing a pairwise consistency test with the cryptographic algorithm keys.

To perform this test, a breakpoint is set during the pairwise consistency check to verify the effective deletion of the used keys, the following keys are used:

- Private Key
- Public Key

The screenshot below is a dump of the private key structure:

Key structure:

This picture intentionally left blank

And the image below is a dump of the private key content ([**Text intentionally left blank**]):

Key content:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

In particular, the private key is the following:

Private key content:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Once the keys are obtained, a zeroization of the keys is forced. Given that the keys are only available in memory once the test is being performed.

The next image shows how when the test is completed, the [**Text intentionally left blank**] function overwrites with zeros the structure which contains the keys:

Zeroized key structure:

This picture intentionally left blank

In addition, the following image shows the stack after performing the whole process:

Call stack:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the keys shown above have been searched without success.

Scenario 1-3: Windows 10 October 2018 Update - Pairwise consistency test with RSA keys

Testing platforms:

- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Test case:

Every time that the TOE is powered up, the TOE tests that the cryptographic algorithms still operate correctly. This is done by performing a pairwise consistency test with the cryptographic algorithm keys.

To perform this test, a breakpoint is set during the pairwise consistency check to verify the effective deletion of the used keys, the following keys are used:

- Private Key
- Public Key

The next image shows the values of the keys structure:

Key structure:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

The following screenshot is a dump of the private key:

Private key:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Once the keys are obtained, a zeroization of the keys is forced. Given that the keys are only available in memory once the test is being performed

The next image shows how when the test is completed, the [**Text intentionally left blank**] function overwrites with zeros the structure which contains the keys:

Zeroized key structure:

This picture intentionally left blank

In addition, the following image show the stack after performing the whole process:

Call stack:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the keys shown above have been searched without success.

Scenario 1-4: Windows 10 October 2018 Update - HTTPS (TLS 1.2) Key Message Authentication Code (MAC)

Testing platforms:

- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

Test case:

The TLS record layer uses a keyed Message Authentication Code (MAC) to protect message integrity, which is based on a hash function [HMAC].

The server has configured a web page only available for TLS connections.

The TOE has accessed to the web page of the server via web browser (e.g. _Internet Explorer*) forcing a TLS connection using a keyed Message Authentication Code (MAC) to protect message integrity [HMAC].

The below image is a dump of the structure containing the HMAC key:

Key structure:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

The following screenshot is a dump of the HMAC key handler:

Key content:

This picture intentionally left blank

In particular, the HMAC key:

© 2019 Microsoft Corporation. All rights reserved.

HMAC key content:

This picture intentionally left blank

Once the HMAC is obtained, the client then closes the connection, forcing the zeroization of HMAC.

The next image shows how when the session is closed, the HMAC key handler is overwritten with zeros:

Zeroized key structure:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

In addition, the following image shows the stack after performing the whole process:

Call stack:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the key shown above has been searched without success.

Scenario 1-5: Windows 10 October 2018 Update - Pairwise consistency test with ECDSA keys

Testing platforms:

- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Test case:

Every time that the TOE is powered up, the TOE tests that the cryptographic algorithms still operate correctly. This is done by performing a pairwise consistency test with the cryptographic algorithm keys.

To perform this test, a breakpoint is set during the pairwise consistency check to verify the effective deletion of the used keys, the following keys are used:

- Private Key
- Public Key

The next image shows the values of the key structure:

Key structure:

This picture intentionally left blank

The following screenshot is a dump of the private key handler:

Private key content:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

An the dump of [**Text intentionally left blank**] field:

Key field:

This picture intentionally left blank

Finally, the dump of the private key:

© 2019 Microsoft Corporation. All rights reserved.

Private key content:

This picture intentionally left blank

Once the keys are obtained, a zeroization of the keys is forced. Given that the keys are only available in memory once the test is being performed

The next image shows how when the test is completed, the [**Text intentionally left blank**] function overwrites with zeros the structure which contains the keys:

Key name handler: [**Text intentionally left blank**]

Zeroized key:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Key field: [**Text intentionally left blank**]

Zeroized key field:

This picture intentionally left blank

And finally, the private key:

© 2019 Microsoft Corporation. All rights reserved.

Zeroized private key:

This picture intentionally left blank

In addition, the following image show the stack after performing the whole process:

Call stack:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the keys shown above have been searched without success.

8.2.1.3.18 Testing volatile memory keys (*secure storage* DPAPI keys)

Scenario 2: Encrypt and decrypt sensitive data using the [**Text intentionally left blank**] interfaces showing how the volatile keys are zeroized after using them.

DPAPI keys relationship:

This picture intentionally left blank

Testing platforms:

- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

Test case:

PBKDF2 HMAC key zeroization:

The screenshot of the debugging session below is for the zeroization of the code screenshot in the *PBKDF2 HMAC key zeroization* section of this document (buffer referred to as [**Text intentionally left blank**]):

Key content:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

Zeroized key:

This picture intentionally left blank

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the key shown above has been searched without success.

© 2019 Microsoft Corporation. All rights reserved.

AES keys zeroization when BCryptDestroyKey is called (DPAPI master AES key & DPAPI AES key):

The screenshots of the debugging session below is for the zeroization of an AES keys when [**Text intentionally left blank**] function is called. Note that the zeroization of AES keys actually occurs in the [**Text intentionally left blank**] function and that is the function that was being debugged in the screenshots below. These screenshots related to the code screenshot for *Zeroization of the symmetric key used to encrypt/decrypt the Master Key* and *Zeroization of the symmetric key derived from the Master Key* sections of this document. Since the two scenarios go through the same code path only one debugging screenshot is provided here.

First of all, the handler of the key:

Key structure:

This picture intentionally left blank

A dump of the key handler:

Key handler content:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

In particular, the key:

Key content:

This picture intentionally left blank

After that, the key handler is dumped after its zeroization:

© 2019 Microsoft Corporation. All rights reserved.

Zeroized key:

This picture intentionally left blank

Finally, the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the keys shown above have been searched without success.

DPAPI Master Key zeroization:

The screenshot of the debugging session below is for the zeroization of the Master Key and relates to the code screenshot in the *Zeroization of the Master Key* section of this document.

First, the content of the Master Key:

Master key content:

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

And then, the content of the memory after its zeroization:

Master key content zeroized:

This picture intentionally left blank

© 2019 Microsoft Corporation. All rights reserved.

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the key shown above has been searched without success.

8.2.1.3.19 Testing DRBG seed zeroization

Scenario 3: Check that the DRBG seed is 512 bits and it is zeroized after its use.

Testing platforms:

- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Test case:

DRBG seed zeroization:

As it can be observed, the following screenshots belong to a debugging session through the source code included in above section *Source code review*.

The memory which contains the seed has been dumped, and it can be observed that the length of the DRBG seed is 512 bits:

DBRG seed content:

This picture intentionally left blank

The same memory region has been dumped after its zeroization, checking that the DBRG seed has been zeroized:

© 2019 Microsoft Corporation. All rights reserved.

DBRG seed zeroized:

This picture intentionally left blank

Finally the memory is dumped. The generated file is opened with a Hex editor (e.g. *HxD*), where the key shown above has been searched without success.

8.2.1.4 Verdict

After performing several tests, to confirm that the volatile keys defined by the vendor in the table 16 of the ***Security Target*** document are zeroized, the evaluator considers that the behaviour described by the vendor matches with the one obtained as result of the testing. Therefore, the evaluator assumes that the vendor position is valid for all keys listed.

Due to this, the evaluator considers that the result obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

8.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_CKM.4.1 requirement.

© 2019 Microsoft Corporation. All rights reserved.

9 FCS_COP.1.1(1)

The assurance activity for the **FCS_COP.1.1(1)** requirement is stated as follows:

> The evaluator will verify that the AGD documents contains instructions required to configure the OS to use the required modes and key sizes. The evaluator will execute all instructions as specified to configure the OS to the appropriate state. The evaluator will perform all of the following tests for each algorithm implemented by the OS and used to satisfy the requirements of this PP:
>
> **AES-CBC Known Answer Tests**
>
> There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
>
> - KAT-1. To test the encrypt functionality of AES-CBC, the evaluator will supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all- zeros key. To test the decrypt functionality of AES-CBC, the evaluator will perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.
> - KAT-2. To test the encrypt functionality of AES-CBC, the evaluator will supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality of AES-CBC, the evaluator will perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.
> - KAT-3. To test the encrypt functionality of AES-CBC, the evaluator will supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. To test the decrypt functionality of AES-CBC, the evaluator will supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each

© 2019 Microsoft Corporation. All rights reserved.

pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.

- KAT-4. To test the encrypt functionality of AES-CBC, the evaluator will supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

To test the decrypt functionality of AES-CBC, the evaluator will perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

The evaluator will test the encrypt functionality by encrypting an i-block message where $1 < i \leq 10$. The evaluator will choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator will also test the decrypt functionality for each mode by decrypting an i-block message where $1 < i \leq 10$. The evaluator will choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator will test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key
  for i = 1 to 1000:
    if i == 1:
      CT[1] = AES-CBC-Encrypt(Key, IV, PT)
      PT = IV
    else:
      CT[i] = AES-CBC-Encrypt(Key, PT)
      PT = CT[i-1]
```

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator will test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Monte Carlo Tests

© 2019 Microsoft Corporation. All rights reserved.

The evaluator will test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

- 128 bit and 256 bit keys
- Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.
- Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.
- Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

The evaluator will test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator will test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-CCM Tests

The evaluator will test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

- 128 bit and 256 bit keys
- Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).
- Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 2 16 bytes, an associated data length of 216 bytes shall be tested.
- Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

© 2019 Microsoft Corporation. All rights reserved.

- Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

To test the generation-encryption functionality of AES-CCM, the evaluator will perform the following four tests:

- **Test 1:** For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- **Test 2:** For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- **Test 3:** For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator will supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.
- **Test 4:** For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator will compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator will supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

Additionally, the evaluator will use tests from the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi", dated September 10, 2002, Section 2.1 AESCCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.

AES-GCM Test

The evaluator will test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

- 128 bit and 256 bit keys
- Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.
- Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

© 2019 Microsoft Corporation. All rights reserved.

- Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

The evaluator will test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator will test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

XTS-AES Test

The evaluator will test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

- 256 bit (for AES-128) and 512 bit (for AES-256) keys
- Three data unit (i.e., plaintext) lengths. One of the data unit lengths shall be a nonzero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.

using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

The evaluator will test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTSAES decrypt.

AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test

The evaluator will test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

- 128 and 256 bit key encryption keys (KEKs)
- Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data

© 2019 Microsoft Corporation. All rights reserved.

unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator will use the AES-KW authenticated-encryption function of a known good implementation.

The evaluator will test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.

The evaluator will test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:

- One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).
- One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

The evaluator will test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.

9.1 Documentation Review activity

9.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[8]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

- *Certificates number #C 211[9] and #C 346[10], #C 347[11] and #C 350[12].*

[8] The test results are described in the evaluation and Assurance Activity Reports

© 2019 Microsoft Corporation. All rights reserved.

The vendor has specified the AES related NIST CAVP certificates:

AES-CBC
- Direction: Decrypt, Encrypt
- Key Length: 128, 192, 256

AES-CCM
- Key Length: 128, 192, 256
- Payload Length: 0-256
- IV Length: 56, 64, 72, 80, 88, 96, 104
- AAD Length: 0-524288
- Tag Length: 32, 48, 64, 80, 96, 112, 128
- Prerequisites:
  - C 211 (AES)

AES-CFB128
- Direction: Decrypt, Encrypt
- Key Length: 128, 192, 256

AES-CFB8
- Direction: Decrypt, Encrypt
- Key Length: 128, 192, 256

AES-CMAC
- Capabilities:
  - Direction: Generation
  - Key Length: 128
  - Message Length: 0, 8, 16, 128, 256, 524288
  - MAC: 128
- Capabilities:
  - Direction: Generation
  - Key Length: 192
  - Message Length: 0, 8, 16, 128, 256, 524288
  - MAC: 128
- Capabilities:
  - Direction: Generation
  - Key Length: 256
  - Message Length: 0, 8, 16, 128, 256, 524288
  - MAC: 128

[9] Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

[10] Validation for #C 346:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30742

[11] Validation for #C 347:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30743

[12] Validation for #C 350:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

AES-CCM

Key Length: 256

Payload Length: 0-256

IV Length: 96

AAD Length: 0-524288

Tag Length: 128

Prerequisites:

C 211 (AES)

AES-KW

Direction: Decrypt, Encrypt

Key Length: 128, 192, 256

Cipher: Cipher

Payload Length: 128, 192, 256, 320, 2048

Prerequisites:

C 211 (AES)

KDF

Capabilities:

KDF Mode: Counter

MAC Mode: CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512

Supported Lengths: 160, 168, 504, 512

Fixed Data Order: Before Fixed Data

Counter Length: 32

Prerequisites:

C 211 (AES)

C 211 (DRBG)

C 211 (HMAC)

C 211 (KAS)

© 2019 Microsoft Corporation. All rights reserved.

AES-CBC

Direction: Decrypt, Encrypt

Key Length: 128, 192, 256

AES-CFB128

Direction: Decrypt, Encrypt

Key Length: 128, 192, 256

AES-CTR

Direction: Encrypt

Key Length: 128, 192, 256

AES-OFB

Direction: Decrypt, Encrypt

Key Length: 128, 192, 256

The ***Operational Guidance*** document, states that for the evaluated version the following security policy needs to be applied (Section 5.2.5.2):

> - Local Policies\Security Options\System cryptography: Use FIPS 140 compliant cryptographic algorithms, including encryption, hashing and signing algorithm.

After applying this policy, only FIPS certified algorithms can be used, including the AES algorithms defined in the tables above.

The TOE satisfies the fulfillment of the cryptographic algorithms, *AES-CCMP*, *AES-CCMP-256* and *AES-GCMP-256* as specified in the ***Security Target*** document. A profound explanation is provided with more detail in the *FCS_CKM.1/WLAN* requirement.

9.1.2 Verdict

The evaluator considers that the AGD document identifies the required instructions to use only the Approved AES algorithms defined in the tables above.

Hence, the **PASS** verdict is assigned to the documentation review activity.

9.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the **PASS** verdict is assigned.

© 2019 Microsoft Corporation. All rights reserved.

9.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_COP.1.1(SYM) requirements.

© 2019 Microsoft Corporation. All rights reserved.

10 FCS_COP.1.1(2)

The assurance activity for the **FCS_COP.1.1(2)** requirement is stated as follows:

> The evaluator will check that the association of the hash function with other application cryptographic functions (for example, the digital signature verification function) is documented in the TSS.
>
> The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator will perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.
>
> The following tests require the developer to provide access to a test application that provides the evaluator with tools that are typically not found in the production application.
>
> - **Test 1:** Short Messages Test (Bit oriented Mode) - The evaluator will generate an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluator will compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
> - **Test 2:** Short Messages Test (Byte oriented Mode) The evaluator will generate an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluator will compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
> - **Test 3:** Selected Long Messages Test (Bit oriented Mode) - The evaluator will generate an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluator will compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
> - **Test 4:** Selected Long Messages Test (Byte oriented Mode) - The evaluator will generate an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 899i, where $1 \leq i \leq m/8$. The message text shall be pseudorandomly generated. The evaluator will compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

© 2019 Microsoft Corporation. All rights reserved.

- **Test 5:** Pseudorandomly Generated Messages Test This test is for byte-oriented implementations only. The evaluator will randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluator will then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluator will then ensure that the correct result is produced when the messages are provided to the TSF.

10.1 Documentation Review activity

10.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[13]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the related hashing algorithm NIST CAVP certificate:

- *Certificate number #C 211*[14].

[13] The test results are described in the evaluation and Assurance Activity Reports

[14] Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

© 2019 Microsoft Corporation. All rights reserved.

HMAC-SHA-1

Key Length: 80, 504, 512, 520, 8192

MAC: 160

Prerequisites:

C 211 (SHS)

HMAC-SHA2-256

Key Length: 128, 504, 512, 520, 8192

MAC: 256

Prerequisites:

C 211 (SHS)

HMAC-SHA2-384

Key Length: 192, 1016, 1024, 1032, 8192

MAC: 384

Prerequisites:

C 211 (SHS)

HMAC-SHA2-512

Key Length: 256, 1016, 1024, 1032, 8192

MAC: 512

Prerequisites:

C 211 (SHS)

In addition, in the same TSS section, the vendor has provided the following wordings regarding the usage of the hashing functions:

> An important feature of CNG is its native implementation of the Suite B algorithms, including algorithms for AES (128, 192, 256 key sizes), the SHA-1 and SHA-2 family (SHA-256, SHA-384 and SHA-512) of hashing algorithms, elliptic curve Diffie Hellman (ECDH), and elliptical curve DSA (ECDSA) over the NIST-standard prime curves P-256, P-384, and P-521.
>
> Protocols such as the Internet Key Exchange (IKE), and Transport Layer Security (TLS), make use of elliptic curve Diffie-Hellman (ECDH) included in Suite B as well as hashing functions.
>
> [...]
>
> Hashing is used by other FIPS Approved algorithms implemented in Windows (the hashed message authentication code, RSA, DSA, and EC DSA signature services, Diffie-Hellman and elliptic curve Diffie-Hellman key agreement, and random bit generation).

© 2019 Microsoft Corporation. All rights reserved.

10.1.2 Verdict

The evaluator considers that the TSS identifies the key hashing algorithms supported by each OS version as well as how these hashing algorithms are used with other cryptographic functions.

Hence, the **PASS** verdict is assigned to the documentation review activity.

10.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the **PASS** verdict is assigned.

10.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_COP.1.1(HASH) requirement.

© 2019 Microsoft Corporation. All rights reserved.

11 FCS_COP.1.1(3)

The assurance activity for the **FCS_COP.1.1(3)** requirement is stated as follows:

> The evaluator will perform the following activities based on the selections in the ST.
>
> The following tests require the developer to provide access to a test application that provides the evaluator with tools that are typically not found in the production application.
>
> **ECDSA Algorithm Tests**
>
> - **Test 1:** ECDSA FIPS 186-4 Signature Generation Test. For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator will generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator will use the signature verification function of a known good implementation.
> - **Test 2:** ECDSA FIPS 186-4 Signature Verification Test. For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator will generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator will verify that 5 responses indicate success and 5 responses indicate failure.
>
> **RSA Signature Algorithm Tests**
>
> - **Test 1:** Signature Generation Test. The evaluator will verify the implementation of RSA Signature Generation by the OS using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator will have the OS use its private key and modulus value to sign these messages. The evaluator will verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures.
> - **Test 2:** Signature Verification Test. The evaluator will perform the Signature Verification test to verify the ability of the OS to recognize another party's valid and invalid signatures. The evaluator will inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys, e, messages, IR format, and/or signatures. The evaluator will verify that the OS returns failure when validating each signature.

11.1 Documentation Review activity

11.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[15]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the related signature algorithm NIST CAVP certificates:

- *Certificates number #C 211[16], #C 348[17], #C 349[18] and #C 350[19] for RSA and,*
- *Certificate number #C 211 for DSA and,*
- *Certificates number #C 211, #C 348 and, #C 350 for ECDSA.*

11.1.1.1 RSA

[15] The test results are described in the evaluation and Assurance Activity Reports

[16] Validation for #C 211: https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

[17] Validation for #C 348: https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30744

[18] Validation for #C 349: https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30745

[19] Validation for #C 350: https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

RSA Decryption Primitive
- Modulus Length: 2048

RSA KeyGen
- Public Exponent Mode: Random
- Private Key Format: standard
- Capabilities:
  - Key Generation Mode: B.3.3
  - Properties:
    - Modulo: 2048
    - Primality Tests: C.2
  - Properties:
    - Modulo: 3072
    - Primality Tests: C.2
- Prerequisites:
  - C 211 (DRBG)
  - C 211 (SHS)

RSA SigGen
- Capabilities:
  - Signature Type: PKCS 1.5
  - Properties:
    - Modulo: 2048
    - Hash Pair:
      - Hash Algorithm: SHA-1
    - Hash Pair:
      - Hash Algorithm: SHA2-256
    - Hash Pair:
      - Hash Algorithm: SHA2-384
    - Hash Pair:
      - Hash Algorithm: SHA2-512
  - Properties:
    - Modulo: 3072
    - Hash Pair:
      - Hash Algorithm: SHA-1
    - Hash Pair:
      - Hash Algorithm: SHA2-256
    - Hash Pair:
      - Hash Algorithm: SHA2-384

© 2019 Microsoft Corporation. All rights reserved.

RSA Decryption Primitive

Modulus Length: 2048

RSA KeyGen

Public Exponent Mode: Fixed

Fixed Public Exponent: 10001

Private Key Format: standard

Capabilities:

Key Generation Mode: B.3.6

Properties:

Modulo: 2048

Primality Tests: C.3

Properties:

Modulo: 3072

Primality Tests: C.3

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

RSA SigGen

Capabilities:

Signature Type: PKCS 1.5

Properties:

Modulo: 2048

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

Hash Pair:

Hash Algorithm: SHA2-384

Hash Pair:

Hash Algorithm: SHA2-512

Properties:

Modulo: 3072

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

© 2019 Microsoft Corporation. All rights reserved.

RSA SigVer

Capabilities:

Signature Type: PKCS 1.5

Properties:

Modulo: 1024

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

Hash Pair:

Hash Algorithm: SHA2-384

Hash Pair:

Hash Algorithm: SHA2-512

Properties:

Modulo: 2048

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

Hash Pair:

Hash Algorithm: SHA2-384

Hash Pair:

Hash Algorithm: SHA2-512

Properties:

Modulo: 3072

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

Hash Pair:

Hash Algorithm: SHA2-384

Hash Pair:

Hash Algorithm: SHA2-512

Prerequisites:

C 211 (SHS)

© 2019 Microsoft Corporation. All rights reserved.

RSA SigGen

Capabilities:

Signature Type: PKCS 1.5

Properties:

Modulo: 2048

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

Hash Algorithm: SHA2-256

Hash Pair:

Hash Algorithm: SHA2-384

Capabilities:

Signature Type: PKCSPSS

Properties:

Modulo: 2048

Hash Pair:

Hash Algorithm: SHA-1

Salt Length: 160

Hash Pair:

Hash Algorithm: SHA2-256

Salt Length: 256

Hash Pair:

Hash Algorithm: SHA2-384

Salt Length: 384

Prerequisites:

C 211 (SHS)

RSA Signature Primitive

Private Key Format: standard

RSA SigVer

Capabilities:

Signature Type: PKCS 1.5

Properties:

Modulo: 1024

Hash Pair:

Hash Algorithm: SHA-1

Hash Pair:

© 2019 Microsoft Corporation. All rights reserved.

11.1.1.2 DSA

DSA KeyGen

Capabilities:

L: 2048

N: 256

Capabilities:

L: 3072

N: 256

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

DSA PQGGen

Capabilities:

L: 2048

N: 256

Hash Algorithm: SHA2-256

P/Q Generation Methods: probable

G Generation Methods: unverifiable

Capabilities:

L: 3072

N: 256

Hash Algorithm: SHA2-256

P/Q Generation Methods: probable

G Generation Methods: unverifiable

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

DSA PQGVer

Capabilities:

L: 2048

N: 256

Hash Algorithm: SHA2-256

P/Q Generation Methods: probable

G Generation Methods: unverifiable

Capabilities:

L: 3072

N: 256

11.1.1.3 ECDSA

© 2019 Microsoft Corporation. All rights reserved.

ECDSA KeyGen

Curve: P-256, P-384, P-521

Secret Generation Mode: Extra Bits

Prerequisites:

C 211 (DRBG)

ECDSA KeyVer

Curve: P-256, P-384, P-521

ECDSA SigGen

Capabilities:

Curve: P-256

Hash Algorithm: SHA2-256

Capabilities:

Curve: P-384

Hash Algorithm: SHA2-384

Capabilities:

Curve: P-521

Hash Algorithm: SHA2-512

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

ECDSA SigGen

Capabilities:

Curve: P-256

Hash Algorithm: SHA2-256

Capabilities:

Curve: P-384

Hash Algorithm: SHA2-384

Capabilities:

Curve: P-521

Hash Algorithm: SHA2-512

Component

Prerequisites:

C 211 (DRBG)

ECDSA SigVer

Capabilities:

Curve: P-256

© 2019 Microsoft Corporation. All rights reserved.

ECDSA KeyGen

Curve: P-256, P-384, P-521

Secret Generation Mode: Extra Bits

Prerequisites:

C 211 (DRBG)

ECDSA KeyVer

Curve: P-256, P-384, P-521

ECDSA SigGen

Capabilities:

Curve: P-256

Hash Algorithm: SHA2-256

Capabilities:

Curve: P-384

Hash Algorithm: SHA2-384

Capabilities:

Curve: P-521

Hash Algorithm: SHA2-512

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

ECDSA SigGen

Capabilities:

Curve: P-256

Hash Algorithm: SHA2-256

Capabilities:

Curve: P-384

Hash Algorithm: SHA2-384

Capabilities:

Curve: P-521

Hash Algorithm: SHA2-512

Component

Prerequisites:

C 211 (DRBG)

C 211 (SHS)

ECDSA KeyGen

Curve: P-256, P-384

Secret Generation Mode: Testing Candidates

Prerequisites:

C 350 (DRBG)

© 2019 Microsoft Corporation. All rights reserved.

11.1.2 Verdict

The assurance activity does not require any documentation information. Nevertheless, the evaluator has provided information about the Signature algorithms supported by the OS.

Hence, the **PASS** verdict is assigned to the documentation review activity.

11.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the **PASS** verdict is assigned.

11.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_COP.1.1(SIGN) requirement.

© 2019 Microsoft Corporation. All rights reserved.

12 FCS_COP.1.1(4)

The assurance activity for the **FCS_COP.1.1(4)** requirement is stated as follows:

> The evaluator will perform the following activities based on the selections in the ST.
>
> For each of the supported parameter sets, the evaluator will compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator will have the OS generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared against the result of generating HMAC tags with the same key and IV using a known-good implementation.

12.1 Documentation Review activity

12.1.1 Findings

The evaluator has reviewed the ***Security Target*** document and the information provided in the TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[20]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the related HMAC algorithm NIST CAVP certificate:

- *Certificates number #C 211[21] and, #C 350[22] for ECDSA.*

[20] The test results are described in the evaluation and Assurance Activity Reports

[21] Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

[22] Validation for #C 350:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

HMAC-SHA-1

Key Length: 80, 504, 512, 520, 8192

MAC: 160

Prerequisites:

C 211 (SHS)

HMAC-SHA2-256

Key Length: 128, 504, 512, 520, 8192

MAC: 256

Prerequisites:

C 211 (SHS)

HMAC-SHA2-384

Key Length: 192, 1016, 1024, 1032, 8192

MAC: 384

Prerequisites:

C 211 (SHS)

HMAC-SHA2-512

Key Length: 256, 1016, 1024, 1032, 8192

MAC: 512

Prerequisites:

C 211 (SHS)

© 2019 Microsoft Corporation. All rights reserved.

HMAC-SHA-1

Key Length: 80, 504, 512, 520, 8192

MAC: 160

Prerequisites:

C 211 (SHS)

HMAC-SHA2-256

Key Length: 128, 504, 512, 520, 8192

MAC: 256

Prerequisites:

C 211 (SHS)

HMAC-SHA2-384

Key Length: 192, 1016, 1024, 1032, 8192

MAC: 384

Prerequisites:

C 211 (SHS)

12.1.2 Verdict

The assurance activity does not require any documentation information. Nevertheless, the evaluator has provided information about the HMAC algorithms supported by the OS.

Hence, the **PASS** verdict is assigned to the documentation review activity.

12.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the **PASS** verdict is assigned.

12.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_COP.1.1(HMAC) requirement.

© 2019 Microsoft Corporation. All rights reserved.

13 FCS_DTLS_EXT.1.1

The assurance activity for the **FCS_DTLS_EXT.1.1** requirement is stated as follows:

> **Test 1:** The evaluator will attempt to establish a connection with a DTLS server, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as DTLS.
>
> Other tests are performed in conjunction with the Assurance Activity listed for FCS_TLSC_EXT.1.

13.1 Documentation Review activity

13.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

13.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

13.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a DTLS server:
   a. *W10DtlsServerAutomator.ps1*, a Powershell script which automatically configures the DTLS server prior each test case execution (choosing between DTLS 1.0 or DTLS 1.2). This script allows the evaluator execute test cases using the software *WebServer.exe*
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution (choosing between DTLS 1.0 or DTLS 1.2 and setting the IP address of the DTLS server). This script allows the evaluator execute test cases using the software *WebClient.exe*.

Both *WebClient.exe* and *WebServer.exe* are written in C++ and use the native APIs from Windows to perform DTLS connections. The provided source code was examined to ensure that this claim is true. It could then be verified that the client tool follows the guidelines exposed in the "Creating a Secure Connection Using Schannel" document from MSDN.

© 2019 Microsoft Corporation. All rights reserved.

Specifically, the key elements of a secure connection implementation using the system libraries were identified, and they are shown next. The fact that native functionality is being used is already hinted from the imported headers:

```
#include <tchar.h>
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <winsock.h>
#include <wincrypt.h>
#include <wintrust.h>
#include <schannel.h>

#define SECURITY_WIN32
#include <security.h>
#include <sspi.h>
```

Especially interesting is the presence of *schannel* and *sspi*. The creation of a Schannel credential structure is also important, since it defines the protocol that will be used to establish the secure connection.

```
//
// Build Schannel credential structure. Currently, this sample only
// specifies the protocol to be used (and optionally the certificate,
// of course). Real applications may wish to specify other parameters
// as well.
//

ZeroMemory(&SchannelCred, sizeof(SchannelCred));

SchannelCred.dwVersion  = SCHANNEL_CRED_VERSION;
if(pCertContext)
{
    SchannelCred.cCreds      = 1;
    SchannelCred.paCred      = &pCertContext;
}

SchannelCred.grbitEnabledProtocols = dwProtocol;
```

This is later used to create an SSPI credentials handle, which is in turn used to initialize the security context with the corresponding method from the API:

```
// Create an SSPI credential.
//

Status = g_pSSPI->AcquireCredentialsHandleA(
                    NULL,                   // Name of principal
                    UNISP_NAME_A,           // Name of package
                    SECPKG_CRED_OUTBOUND,   // Flags indicating use
                    NULL,                   // Pointer to logon ID
                    &SchannelCred,          // Package specific data
                    NULL,                   // Pointer to GetKey() func
                    NULL,                   // Value to pass to GetKey()
```

© 2019 Microsoft Corporation. All rights reserved.

```
                    phCreds,                // (out) Cred Handle
                    &tsExpiry);             // (out) Lifetime (optional)
if(Status != SEC_E_OK)
{
    printf("**** Error 0x%x returned by AcquireCredentialsHandle\n", Status);
    goto cleanup;
}
```

```
else
{
    scRet = g_pSSPI->InitializeSecurityContextA(
        phCreds,
        NULL,
        pszServerName,
        dwSSPIFlags,
        0,
        SECURITY_NATIVE_DREP,
        NULL,
        0,
        phContext,
        &OutBuffer,
        &dwSSPIOutFlags,
        &tsExpiry);
}

printf("SSPIFlags = %x\n", dwSSPIFlags);
printf("SSPIOutFlags = %x\n", dwSSPIOutFlags);

if(scRet != SEC_I_CONTINUE_NEEDED)
{
    printf("**** Error %d returned by InitializeSecurityContext (1)\n", scRet);
    return scRet;
}
```

The other critical part of the DTLS connection procedure is the certificate validation performed by the client which, as it could also be observed, uses the native functionality from *wincrypt* provided by the *CertVerifyCertificateChainPolicy* method.

```
//
// Validate certificate chain.
//

ZeroMemory(&polHttps, sizeof(HTTPSPolicyCallbackData));
polHttps.cbStruct            = sizeof(HTTPSPolicyCallbackData);
polHttps.dwAuthType          = AUTHTYPE_SERVER;
polHttps.fdwChecks           = dwCertFlags;
polHttps.pwszServerName      = pwszServerName;

memset(&PolicyPara, 0, sizeof(PolicyPara));
PolicyPara.cbSize            = sizeof(PolicyPara);
PolicyPara.pvExtraPolicyPara = &polHttps;
```

© 2019 Microsoft Corporation. All rights reserved.

```
memset(&PolicyStatus, 0, sizeof(PolicyStatus));
PolicyStatus.cbSize = sizeof(PolicyStatus);

if(!CertVerifyCertificateChainPolicy(
                        CERT_CHAIN_POLICY_SSL,
                        pChainContext,
                        &PolicyPara,
                        &PolicyStatus))
{
    Status = GetLastError();
    printf("Error 0x%x returned by CertVerifyCertificateChainPolicy!\n", Status);
    goto cleanup;
}
```

If an error is obtained in this step, the client aborts the connection (which in this case, since DTLS uses the UDP protocol, means that it just stops all communication with the server).

```
// Attempt to validate server certificate.
Status = VerifyServerCertificate(pRemoteCertContext,
                                 pszServerName,
                                 0);
if(Status)
{
    // The server certificate did not validate correctly. At this
    // point, we cannot tell if we are connecting to the correct
    // server, or if we are connecting to a "man in the middle"
    // attack server.

    // It is therefore best if we abort the connection.

    printf("**** Error authenticating HTTPS server credentials!\n", Status);
    printf("Terminating connection with server.\n");

    if (!fContinueAfterError)
    {
        goto cleanup;
    }
}
```

The Process Monitor tool was additionally used to check that the system libraries in charge of performing secure communications (e.g. *schannel.dll*) were loaded by *WebClient.exe* before performing network connections, effectively using the mechanism from the operating system.

© 2019 Microsoft Corporation. All rights reserved.

1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS	Image Base: 0x740...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\bcryptprimitives.dll	SUCCESS	Image Base: 0x741...
1:49:2...	WebClient.exe	172	Thread Create		SUCCESS	Thread ID: 1688
1:49:2...	WebClient.exe	172	Thread Create		SUCCESS	Thread ID: 2292
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\crypt32.dll	SUCCESS	Image Base: 0x744...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\ucrtbase.dll	SUCCESS	Image Base: 0x746...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\msasn1.dll	SUCCESS	Image Base: 0x757...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\secur32.dll	SUCCESS	Image Base: 0x6ffb...
1:49:2...	WebClient.exe	172	Thread Create		SUCCESS	Thread ID: 296
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\profapi.dll	SUCCESS	Image Base: 0x74e...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\schannel.dll	SUCCESS	Image Base: 0x6fc...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\mswsock.dll	SUCCESS	Image Base: 0x6f3...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\NapiNSP.dll	SUCCESS	Image Base: 0x6f3...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Image Base: 0x752...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\dnsapi.dll	SUCCESS	Image Base: 0x6fa...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\nsi.dll	SUCCESS	Image Base: 0x757...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\IPHLPAPI.DLL	SUCCESS	Image Base: 0x6fa...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\winmm.dll	SUCCESS	Image Base: 0x6f3...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\nlaapi.dll	SUCCESS	Image Base: 0x6f3...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\FWPUCLNT.DLL	SUCCESS	Image Base: 0x6f2f...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\bcrypt.dll	SUCCESS	Image Base: 0x6fe...
1:49:2...	WebClient.exe	172	Load Image	C:\Windows\SysWOW64\rasadhlp.dll	SUCCESS	Image Base: 0x6f2...
1:49:2...	WebClient.exe	172	UDP Send	WIN-9L39HP8B309:52582 -> WIN-9L39HP8B309:440	SUCCESS	Length: 160, seqn...
1:49:2...	WebClient.exe	172	UDP Receive	WIN-9L39HP8B309:52582 -> WIN-9L39HP8B309:440	SUCCESS	Length: 60, seqnu...
1:49:2...	WebClient.exe	172	UDP Send	WIN-9L39HP8B309:52582 -> WIN-9L39HP8B309:440	SUCCESS	Length: 192, seqn...

Regarding the specific tests performed, some of them require modification of outgoing server messages in the DTLS connection. Unless otherwise noted, the test cases included in *Web-Server.exe* were leveraged to perform those tests without the need of a MITM tool.

13.2.1 Test 1

In this test the evaluator will establish a connection to a DTLS Server and check that the connection succeeds and is identified as DTLS.

13.2.1.1 Setup

The following certificates shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es

The certificates listed form a valid certification path. The server certificate is available in pfx format and the root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Windows Server 2016 Standard)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.20.*

© 2019 Microsoft Corporation. All rights reserved.

Server machine shall contain the PowerShell script *W10DtlsServerAutomator.ps1* as well as the *WebServer.exe* tool also mentioned before.

Client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *Web-Client.exe* tool mentioned in the introduction to this section.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance*** document. The *Wireshark* network analyzer shall be installed on the client machine.

13.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run the Powershell script *W10DtlsServerAutomator.ps1* as follows:

```
.\W10DtlsServerAutomator.ps1 dtls1 #for DTLS10
.\W10DtlsServerAutomator.ps1 dtls12 #for DTLS12
```

- Select the option number 1 when the menu is prompt.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <dtls_version> (e.g. dlts1 for DTLS 1.0, dtls12 for DTLS 1.2 and <ip_server_address> and select the option number 10 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection from the client's output, server's output and the Wireshark capture.

13.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The evaluator has obtained the same results for all the tested platforms.

In the first part of Test 1, the evaluator has established a connection between the DTLS Server and the DTLS Client on the TOE, using the 1.0 version of the DTLS protocol. The connection is successful and makes use of the right protocol as it can be shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.007046	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009340	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.012047	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.013590	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.014875	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
7	0.031160	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.038141	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.041090	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.041655	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.061465	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.072978	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.302536	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.328469	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.332020	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.338677	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

© 2019 Microsoft Corporation. All rights reserved.

The client machine offers first the cipher suites available in the TOE, since the connection is made with the 1.0 version of the DTLS protocol; this version only supports four of the ten cipher suites selected according the ***Operational Guidance*** document.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.007046	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009340	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.012047	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.013590	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.014875	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
7	0.031160	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.038141	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.041090	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.041655	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.061465	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.072978	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.302536	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.328469	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.332020	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.338677	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
    Epoch: 0
    Sequence Number: 0
    Length: 94
  ▾ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 82
      Message Sequence: 0
      Fragment Offset: 0
      Fragment Length: 82
      Version: DTLS 1.0 (0xfeff)
    ▸ Random: 5bf53b8aed8588fa565a4b6f35f5b85f16818bd5b26bc915...
      Session ID Length: 0
      Cookie Length: 0
      Cipher Suites Length: 4
    ▾ Cipher Suites (2 suites)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Compression Methods Length: 1
    ▸ Compression Methods (1 method)
      Extensions Length: 36
    ▸ Extension: server_name (len=19)
    ▸ Extension: SessionTicket TLS (len=0)
```

Finally, in the Server Hello message, the DTLS Server use one of the cipher suites available in the client machine.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.007046	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009340	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.012047	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.013590	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.014875	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
7	0.031160	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.038141	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.041090	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.041655	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.061465	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.072978	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.302536	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.328469	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.332020	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.338677	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
▸ User Datagram Protocol, Src Port: 440, Dst Port: 64492
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.0 (0xfeff)
      ▸ Random: 5bf53815574eee676f1534f8e27c2e3506faa3ffcef3ebe5...
        Session ID Length: 32
        Session ID: 693800008ad08152bf7c9a61e5766f541abf68ca161e7211...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

In the second part of Test 1, a connection between the DTLS Server and the DTLS Client on the TOE, using the 1.2 version of the DTLS protocol is made. The connection is successful and makes use of the right protocol as it can be shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
5	17.614949	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
6	17.619312	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	17.623164	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
8	17.626658	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	17.630045	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	17.633618	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
11	17.677999	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	17.687049	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	17.689600	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	17.690048	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
15	17.712329	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
16	17.716802	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
25	29.642505	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
26	29.668007	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
27	29.674303	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
28	29.679667	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

The client machine offers first the cipher suites available in the TOE, this version supports all the cipher suites selected according to the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
5	17.614949	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
6	17.619312	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	17.623164	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
8	17.626658	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	17.630045	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	17.633618	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
11	17.677999	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	17.687049	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	17.689600	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	17.690048	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
15	17.712329	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
16	17.716802	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
25	29.642505	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
26	29.668007	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
27	29.674303	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
28	29.679667	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

```
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6ac15efbf5f1fdf9f2361db4c9553f0506294cb183914...
    Session ID Length: 0
    Cookie Length: 32
    Cookie: e99214ddb14732be6241469a8d0f23eef2f05586e7a2d02a...
    Cipher Suites Length: 26
  ▾ Cipher Suites (13 suites)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Methods Length: 1
  ▸ Compression Methods (1 method)
```

Finally, in the Server Hello message, the DTLS Server use one of the cipher suites available in the client machine.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
5	17.614949	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
6	17.619312	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	17.623164	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
8	17.626658	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	17.630045	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	17.633618	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
11	17.677999	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	17.687049	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	17.689600	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	17.690048	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
15	17.712329	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
16	17.716802	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
25	29.642505	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
26	29.668007	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
27	29.674303	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
28	29.679667	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

```
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 56723
▾ Datagram Transport Layer Security
  ▾ DTLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.2 (0xfefd)
      ▸ Random: 5bf6a8ef22631247a5b1a7e2f25a0ba1679a537bbe3d7eb4...
        Session ID Length: 32
        Session ID: 7d0b000092b017cff3cda5965165bddcef4f4d474de5e279...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

13.2.1.4 Verdict

From the results of the previous test, the evaluator considers that the results obtained from the test activity demonstrate the fulfillment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

13.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_DTLS_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

14 FCS_DTLS_EXT.1.2

The assurance activity for the **FCS_DTLS_EXT.1.2** requirement is stated as follows:

> The evaluator will perform the assurance activities listed for FCS_TLSC_EXT.1.

14.1 Documentation Review activity

14.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

14.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

14.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a DTLS server:
   a. *W10DtlsServerAutomator.ps1*, a Powershell script which automatically configures the DTLS server prior each test case execution (choosing between DTLS 1.0 or DTLS 1.2). This script allows the evaluator execute test cases using the software *WebServer.exe*
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution (choosing between DTLS 1.0 or DTLS 1.2 and setting the IP address of the DTLS server). This script allows the evaluator execute test cases using the software *WebClient.exe*.

The documentation for FCS_DTLS_EXT.1.1 includes information about their suitability for this evaluation.

14.2.1 Test 1 - FCS_TLSC_EXT.1.1 for DTLS

14.2.1.1 Setup

The following certificates shall be used to perform the assurance activities listed on the Protection Profile:

© 2019 Microsoft Corporation. All rights reserved.

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The server certificate is available in pfx format and the root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Windows Server 2016 Standard)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.20.*

Server machine shall contain the PowerShell script *W10DtlsServerAutomator.ps1* as well as the *WebServer.exe* tool also mentioned before.

Client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *WebClient.exe* tool mentioned in the introduction to this section.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance*** document. The *Wireshark* network analyzer shall be installed on the client machine.

14.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run the Powershell script *W10DtlsServerAutomator*.ps1 as follows:

```
.\W10DtlsServerAutomator.ps1 dtls1 #for DTLS 1.0
.\W10DtlsServerAutomator.ps1 dtls12 #for DTLS 1.2
```

- Select the option number 2 when the menu is prompt.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <dtls_version> (e.g. dlts1 for DTLS 1.0, dtls12 for DTLS 1.2 and <ip_server_address> and select the option number 11 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection from the client's output, server's output and the Wireshark capture.
- Repeat the steps above for both DTLS 1.0 and DTLS 1.2.

14.2.1.3 Results

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The cipher suites supported by the client are specified on the Client Hello, as shown in the following picture for DTLS 1.0:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.007046	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009340	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.012047	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.013590	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.014875	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
7	0.031160	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.038141	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.041090	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.041655	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.061465	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.072978	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.302536	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.328469	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.332020	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.338677	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
    Epoch: 0
    Sequence Number: 0
    Length: 94
  ▾ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 82
      Message Sequence: 0
      Fragment Offset: 0
      Fragment Length: 82
      Version: DTLS 1.0 (0xfeff)
    ▸ Random: 5bf53b8aed8588fa565a4b6f35f5b85f16818bd5b26bc915...
      Session ID Length: 0
      Cookie Length: 0
      Cipher Suites Length: 4
    ▾ Cipher Suites (2 suites)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Compression Methods Length: 1
    ▸ Compression Methods (1 method)
      Extensions Length: 36
    ▸ Extension: server_name (len=19)
    ▸ Extension: SessionTicket TLS (len=0)
```

Note that only four out of the ten cipher suites in the ***Operational Guidance*** are sent, due to DTLS 1.0 not supporting the newest cipher suites.

The server responds with the selected cipher suite on the Server Hello, as shown next for the first connection with DTLS 1.0:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
7	5.716146	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
8	5.720876	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
9	5.725868	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
10	5.734973	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
11	5.738423	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
12	5.741508	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
13	5.779948	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
14	5.785101	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
15	5.788181	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
16	5.788651	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
17	5.814297	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
18	5.818648	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
32	17.939142	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
33	17.955957	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
34	17.959057	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
35	17.961934	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 63892
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.0 (0xfeff)
      ▸ Random: 5bf539ff9edfa53a86b6388d4e084272385622965a0cda89...
        Session ID Length: 32
        Session ID: 6401000047ab0bfefcd4f5818fb997b5a1b2dd8961900abb...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

The fact that the connection is successful can be verified with the network capture, as seen the Application Data being transmitted.

For DTLS 1.2, the cipher suites are specified in the Client Hello in the same way:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
15	12.031878	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
16	12.035895	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
17	12.040262	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
18	12.050443	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
19	12.053054	10.10.20.40	10.10.20.52	DTLSv1.2	978	Certificate
20	12.064186	10.10.20.40	10.10.20.52	DTLSv1.2	212	Server Key Exchange
21	12.082965	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
22	12.090500	10.10.20.52	10.10.20.40	DTLSv1.2	133	Client Key Exchange
23	12.092467	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
24	12.093046	10.10.20.52	10.10.20.40	DTLSv1.2	135	Encrypted Handshake Message
25	12.112445	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
26	12.120626	10.10.20.40	10.10.20.52	DTLSv1.2	135	Encrypted Handshake Message
27	12.167792	10.10.20.52	10.10.20.40	DTLSv1.2	183	Application Data
28	12.174057	10.10.20.40	10.10.20.52	DTLSv1.2	151	Application Data
29	12.175885	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
30	12.177764	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

```
    Fragment Offset: 0
    Fragment Length: 140
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6bbede49cbf4a968566f3c8d70610aae3c6f5824c1851...
    Session ID Length: 0
    Cookie Length: 0
    Cipher Suites Length: 26
  ▾ Cipher Suites (13 suites)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Methods Length: 1
  ▸ Compression Methods (1 method)
    Extensions Length: 72
```

The server responds with the selected cipher suite on the Server Hello, as shown next for the first connection with DTLS 1.2:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
8	6.715409	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
9	6.719893	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
10	6.723954	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
11	6.726642	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
12	6.736649	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
13	6.739176	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
14	6.771926	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
15	6.777238	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
16	6.779965	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec

```
▸ Frame 11: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 49666
▾ Datagram Transport Layer Security
  ▾ DTLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.2 (0xfefd)
      ▸ Random: 5bf6bb39fc18cbdee417da22ca40b4b09a5280d85b81abab...
        Session ID Length: 32
        Session ID: f24b00000859951695d80a26dc3c10f31d54ec79b99ae2fb
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

The fact that the connection is successful can be verified with the network capture, as shown in the next picture for the first connection:

No.	Time	Source	Destination	Protocol	Length	Info
8	6.715409	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
9	6.719893	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
10	6.723954	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
11	6.726642	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
12	6.736649	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
13	6.739176	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
14	6.771926	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
15	6.777238	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
16	6.779965	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
17	6.780598	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
18	6.829607	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
19	6.840259	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
30	17.945392	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
31	17.959432	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
32	17.960902	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
33	17.962592	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

A total of 13 connections are performed in this test, with different cipher suites selected by the server for DTLS 1.0 and 1.2, so that the whole list of offered cipher suites is covered. The successful DTLS connection is checked for every case.

14.2.1.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

14.2.2 Test 2 - FCS_TLSC_EXT.1.1 for DTLS

14.2.2.1 Setup

The setup is identical to the one defined for Test 1.

14.2.2.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The first connection is performed with a server certificate that has the serverAuth purpose, as shown in the next picture:

```
▾ Handshake Protocol: Certificate (Reassembled)
    Handshake Type: Certificate (11)
    Length: 1108
    Message Sequence: 2
    Fragment Offset: 1071
    Fragment Length: 37
    Certificates Length: 1105
  ▾ Certificates (1105 bytes)
      Certificate Length: 1102
    ▾ Certificate: 3082044a30820232a003020102020101300d06092a864886... (id-at-commonName=test.epoche.es,id-at-organizational
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-extKeyUsage)
              Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
            ▾ KeyPurposeIDs: 1 item
                KeyPurposeId: 1.3.6.1.5.5.7.3.1 (id-kp-serverAuth)
```

The fact that the connection is established is checked on the network trace:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.003132	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.004788	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.010820	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.017208	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.019723	10.10.20.40	10.10.20.52	DTLSv1.0	104	Certificate (Reassembled)
7	0.051564	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.057179	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.058706	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.058707	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.088960	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.093446	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.692977	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.700404	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.702190	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.703879	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

It can also be observed on the client tool´s output, where the default web content was successfully transmitted and presented after decryption:

```
DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21          Hello, world!

14 bytes of (encrypted) application data received
0000  14 fe fd 00 00 00 00 00:00 00 06 00 01 01       ..............
```

For the next connection, a certificate without serverAuth purpose (codeSigning purpose is sent instead) is presented, as follows:

```
▾ Handshake Protocol: Certificate (Reassembled)
    Handshake Type: Certificate (11)
    Length: 1108
    Message Sequence: 2
    Fragment Offset: 1071
    Fragment Length: 37
    Certificates Length: 1105
  ▾ Certificates (1105 bytes)
      Certificate Length: 1102
    ▾ Certificate: 3082044a30820232a0030201020201300d06092a864886... (id-at-commonName=test.epoche.es,id-at-organizationalUni
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-extKeyUsage)
              Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
            ▾ KeyPurposeIDs: 1 item
                KeyPurposeId: 1.3.6.1.5.5.7.3.3 (id-kp-codeSigning)
```

The network capture reflects the connection failure:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
12	14.725452	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
13	14.729794	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
14	14.734120	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
15	14.737244	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
16	14.740594	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
17	14.750950	10.10.20.40	10.10.20.52	DTLSv1.0	104	Certificate (Reassembled)
18	14.778874	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
19	14.785791	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
20	14.789110	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
21	14.789682	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
22	14.824053	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
23	14.828819	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

As it can be observed, no application data is being sent, meaning that the handshake did not finish successfully. This is also reflected on the client´s output:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b0110 (CERT_E_WRONG_USAGE) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

Equivalent results were obtained when evaluating DTLS 1.2. When the server presents the first certificate, with *serverAuth* purpose, the connection is successful:

No.	Time	Source	Destination	Protocol	Length	Info
11	27.589959	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
12	27.591709	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
13	27.593053	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
14	27.593730	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
15	27.594493	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
16	27.595269	10.10.20.40	10.10.20.52	DTLSv1.2	104	Certificate (Reassembled)
17	27.602629	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
18	27.609068	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
19	27.616748	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
20	27.617270	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
21	27.640783	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
22	27.646739	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
34	39.702779	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
35	39.707040	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
36	39.708528	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
37	39.710123	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

The client tool also shows this fact:

```
DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21          Hello, world!

14 bytes of (encrypted) application data received
0000  14 fe fd 00 00 00 00 00:00 00 06 00 01 01       ..............
```

If the server presents the second certificate with *codeSigning* purpose, the connection does not finish successfully, just like with the DTLS 1.0 case:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004483	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.008221	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.011159	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.014138	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.017819	10.10.20.40	10.10.20.52	DTLSv1.2	104	Certificate (Reassembled)
7	0.063998	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.078024	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.096472	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.098068	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.109001	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.116352	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

The same validation error message is obtained on the client side.

14.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

14.2.3 Test 3 - FCS_TLSC_EXT.1.1 for DTLS

14.2.3.1 Setup

The setup is identical to the one defined for Test 1.

14.2.3.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the outgoing Server Hello and changes the selected cipher suite from TLS_RSA_WITH_AES_128_CBC_SHA to TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA, The corresponding code snippet where the modification is performed is shown in the next block:

```
if ((21 == iSFR && 3 == iTest && fInitContext) || (NULL != pszTest &&
    !strcmp(pszTest, "TLSC_EXT.1.1.3")))
        {
            printf("Checking for the right handshake message...\n");
            //The evaluator shall send a server certificate in the TLS
                connection that the does not match the server - selected
            // ciphersuite(for example, send a ECDSA certificate while using the
                TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or
            // send a RSA certificate while using one of the ECDSA ciphersuites.)
            // first protocol msg in DTLS is not the server hello
```

© 2019 Microsoft Corporation. All rights reserved.

```
            if ((IS_DTLS(dwProtocol) && handshakeLoopCounter == 1) ||
                (!IS_DTLS(dwProtocol) && fInitContext))
            {
                printf("Correct handshake message identified\n");
                // in DTLS the algorithm is at offset 0x5c. In non-DTLS, offset
                    is at 0x4c
                targetLocation = !IS_DTLS(dwProtocol) ? 0x4c : 0x5c;
                printf("FCS_TLSC_EXT.1.1 Test 3: Changing cipher suite to 0xc0,
                    0x09 (TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA)\n");
                printf("200 bytes of output before modification (cipher suite at
                    0x%x)\n", targetLocation);
                PrintHexDump(200, (PBYTE)OutBuffers[0].pvBuffer);
                pbTemp = (PBYTE)OutBuffers[0].pvBuffer;
                \*(pbTemp + targetLocation) = 0xc0; // (BYTE)iCipherSuite;
                \*(pbTemp + targetLocation + 1) = 0x09; //
                    TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
                printf("\n200 bytes of output after modification (cipher suite
                    at 0x%x)\n", targetLocation);
                PrintHexDump(200, (PBYTE)OutBuffers[0].pvBuffer);
            }
        }
```

The server´s output indicates the concrete modification that was performed:

```
FCS_TLSC_EXT.1.1 Test 3: Changing cipher suite to 0xc0, 0x09 (TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA)
200 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 52 8e 84  Q.......Q..[.R..
0020  23 2f 45 b3 de e9 e1 0d:0e 08 dd a0 43 92 07 4b  #/E.........C..K
0030  3e b8 05 e8 7c 37 f4 b1:50 4b 96 20 14 46 00 00  >...|7..PK. .F..
0040  bd e2 36 34 e0 31 6c 61:73 0c eb 06 d8 fc 45 fb  ..64.1las.....E.
0050  cf 18 8d 62 78 43 d3 2e:42 a9 a3 32 00 2f 00 00  ...bxC..B..2./..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:08 eb 4f 07 fb 00 00 00  LMEMp.....O.....
0080  00 00 00 00 00 00 00 00:3c 62 24 54 00 08 00 80  ........<b$T....
0090  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00a0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00b0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00c0  00 00 00 00 00 00 00 00:                         ........

200 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 52 8e 84  Q.......Q..[.R..
0020  23 2f 45 b3 de e9 e1 0d:0e 08 dd a0 43 92 07 4b  #/E.........C..K
0030  3e b8 05 e8 7c 37 f4 b1:50 4b 96 20 14 46 00 00  >...|7..PK. .F..
0040  bd e2 36 34 e0 31 6c 61:73 0c eb 06 d8 fc 45 fb  ..64.1las.....E.
0050  cf 18 8d 62 78 43 d3 2e:42 a9 a3 32 c0 09 00 00  ...bxC..B..2....
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:08 eb 4f 07 fb 00 00 00  LMEMp.....O.....
0080  00 00 00 00 00 00 00 00:3c 62 24 54 00 08 00 80  ........<b$T....
0090  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00a0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00b0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00c0  00 00 00 00 00 00 00 00:                         ........
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Server Hello can be appreciated in the next picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004588	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006118	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.058427	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.066413	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.077041	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.087871	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

```
▸ Frame 4: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 65262
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.0 (0xfeff)
      ▸ Random: 5bf5528e84232f45b3dee9e10d0e08dda04392074b3eb805...
        Session ID Length: 32
        Session ID: 14460000bde23634e0316c61730ceb06d8fc45fbcf188d62
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

The OS interrupts the handshake after receiving this certificate, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004588	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006118	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.058427	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.066413	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.077041	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.087871	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090331
(SEC_E_ALGORITHM_MISMATCH)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090331 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

© 2019 Microsoft Corporation. All rights reserved.

The same results are obtained for DTLS 1.2. The corresponding modified Server Hello is shown next, both as server's output and network capture:

```
FCS_TLSC_EXT.1.1 Test 3: Changing cipher suite to 0xc0, 0x09 (TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA)
200 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 bf ef d3  Q.......Q..[....
0020  b1 c6 ae 6a e2 11 67 32:ea 44 22 1a 52 b1 72 e0  ...j..g2.D".R.r.
0030  57 97 22 2c d9 cd ee 8a:23 60 1f 20 a2 2e 00 00  W.",....#` . ....
0040  b6 7f 45 eb 2e 1c 6f 31:93 43 6b 7f 85 f6 7d ff  ..E...o1.Ck...}.
0050  4c 7d d4 88 00 86 22 c6:17 37 b4 6d 00 2f 00 00  L}...."..7.m./..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:08 e9 d0 3f 01 00 00 00  LMEMp......?....
0080  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
0090  00 00 00 00 00 00 00 00:bd 22 94 c1 7a e2 00 0c  ........."..z...
00a0  1c 00 00 00 94 00 00 00:00 00 00 00 20 00 00 00  ............ ...
00b0  00 00 00 00 00 00 00 00:00 00 00 00 74 00 65 00  ............t.e.
00c0  2d 00 45 00 43 00 44 00:                          -.E.C.D.

200 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 bf ef d3  Q.......Q..[....
0020  b1 c6 ae 6a e2 11 67 32:ea 44 22 1a 52 b1 72 e0  ...j..g2.D".R.r.
0030  57 97 22 2c d9 cd ee 8a:23 60 1f 20 a2 2e 00 00  W.",....#` . ....
0040  b6 7f 45 eb 2e 1c 6f 31:93 43 6b 7f 85 f6 7d ff  ..E...o1.Ck...}.
0050  4c 7d d4 88 00 86 22 c6:17 37 b4 6d c0 09 00 00  L}...."..7.m....
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:08 e9 d0 3f 01 00 00 00  LMEMp......?....
0080  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
0090  00 00 00 00 00 00 00 00:bd 22 94 c1 7a e2 00 0c  ........."..z...
00a0  1c 00 00 00 94 00 00 00:00 00 00 00 20 00 00 00  ............ ...
00b0  00 00 00 00 00 00 00 00:00 00 00 00 74 00 65 00  ............t.e.
00c0  2d 00 45 00 43 00 44 00:                          -.E.C.D.
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6bfefd3b1c6ae6ae2116732ea44221a52b172e0579722...
    Session ID Length: 32
    Session ID: a22e0000b67f45eb2e1c6f3193436b7f85f67dff4c7dd488
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
16	18.066338	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
17	18.071296	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
18	18.075317	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
19	18.109671	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
20	18.116992	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
22	18.127577	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
23	18.134301	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

© 2019 Microsoft Corporation. All rights reserved.

14.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

14.2.4 Test 4 - FCS_TLSC_EXT.1.1 for DTLS

14.2.4.1 Setup

The setup is identical to the one defined for Test 1.

14.2.4.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the outgoing Server Hello and changes the selected cipher suite to TLS_NULL_WITH_NULL_NULL. The corresponding code snippet where the modification is performed is shown in the next block:

```
else if ( ((21 == iSFR && 4 == iTest && 'a' == cSubTest)
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.4"))) )
{

    // first protocol msg in DTLS is not the server hello
    if ((IS_DTLS(dwProtocol) && handshakeLoopCounter == 1) ||
        (!IS_DTLS(dwProtocol) && fInitContext))
    {
        targetLocation = !IS_DTLS(dwProtocol) ? 0x4c : 0x5c;
        printf("\nchanging cipher suite to TLS_NULL_WITH_NULL_NULL 0x00,0x00\n");
        printf("200 bytes of output before modification (cipher suite at
            0x%x)\n", targetLocation);
        PrintHexDump(200, (PBYTE)OutBuffers[0].pvBuffer);
        pbTemp = (PBYTE)OutBuffers[0].pvBuffer;
        \*(pbTemp + targetLocation) = 0x00; // (BYTE)iCipherSuite;
        \*(pbTemp + targetLocation + 1) = 0x00; // TLS_NULL_WITH_NULL_NULL
        printf("\n200 bytes of output after modification (cipher suite at
            0x%x)\n", targetLocation);
        PrintHexDump(200, (PBYTE)OutBuffers[0].pvBuffer);
    }
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
changing cipher suite to TLS_NULL_WITH_NULL_NULL 0x00,0x00
200 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 53 ff d4  Q.......Q..[.S..
0020  3f d6 cc ff 51 9c c9 28:85 53 48 42 73 28 7d aa  ?...Q..(.SHBs(}.
0030  17 90 a5 82 0a 90 b7 71:e5 27 93 20 63 3b 00 00  .......q.'. c;..
0040  8a 7b b9 2c 17 f6 dc a3:e7 05 40 76 a9 0d 1f fa  .{.,......@v....
0050  76 c0 01 94 42 0d ca 5a:1d 1c 29 9a 00 2f 00 00  v...B..Z..)../..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:b8 e8 cf 2a a0 00 00 00  LMEMp......*....
0080  00 00 00 00 00 00 00 00:c4 bc 2a ab 00 12 00 80  ..........*.....
0090  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00a0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00b0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00c0  00 00 00 00 00 00 00 00:                         ........

200 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 53 ff d4  Q.......Q..[.S..
0020  3f d6 cc ff 51 9c c9 28:85 53 48 42 73 28 7d aa  ?...Q..(.SHBs(}.
0030  17 90 a5 82 0a 90 b7 71:e5 27 93 20 63 3b 00 00  .......q.'. c;..
0040  8a 7b b9 2c 17 f6 dc a3:e7 05 40 76 a9 0d 1f fa  .{.,......@v....
0050  76 c0 01 94 42 0d ca 5a:1d 1c 29 9a 00 00 00 00  v...B..Z..).....
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:b8 e8 cf 2a a0 00 00 00  LMEMp......*....
0080  00 00 00 00 00 00 00 00:c4 bc 2a ab 00 12 00 80  ..........*.....
0090  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00a0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00b0  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
00c0  00 00 00 00 00 00 00 00:                         ........
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Server Hello can be appreciated in the next picture:

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.0 (0xfeff)
  ▸ Random: 5bf553ffd43fd6ccff519cc9288553484273287daa1790a5...
    Session ID Length: 32
    Session ID: 633b00008a7bb92c17f6dca3e7054076a90d1ffa76c00194...
    Cipher Suite: TLS_NULL_WITH_NULL_NULL (0x0000)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

The OS does not finish the handshake, as observed on the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004473	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009984	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.060417	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.067232	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.075479	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.081495	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090331
(SEC_E_ALGORITHM_MISMATCH)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090331 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified Server Hello is shown next, both as server's output and network capture:

```
changing cipher suite to TLS_NULL_WITH_NULL_NULL 0x00,0x00
200 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c0 6a 68  Q.......Q..[..jh
0020  d7 43 d0 b8 44 1b ff 29:41 f7 ab ca 84 cc 91 3f  .C..D..)A......?
0030  59 9a 36 a7 51 cc 5c b5:eb 29 8f 20 50 35 00 00  Y.6.Q.\..). P5..
0040  4e 45 39 6d 5a 61 6a 67:1f 16 3a 28 66 c0 f7 5d  NE9mZajg..:(f..]
0050  36 4c 5c 34 af c3 4b 9f:c1 46 ab 70 00 2f 00 00  6L\4..K..F.p./..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:48 e8 af dc 7f 00 00 00  LMEMp...H.......
0080  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
0090  00 00 00 00 00 00 00 00:2e 7c 1b 78 f6 56 00 0c  .........|.x.V..
00a0  1c 00 00 00 94 00 00 00:00 00 00 00 20 00 00 00  ............ ...
00b0  00 00 00 00 00 00 00 00:00 00 00 00 74 00 65 00  ............t.e.
00c0  2d 00 45 00 43 00 44 00:                          -.E.C.D.

200 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c0 6a 68  Q.......Q..[..jh
0020  d7 43 d0 b8 44 1b ff 29:41 f7 ab ca 84 cc 91 3f  .C..D..)A......?
0030  59 9a 36 a7 51 cc 5c b5:eb 29 8f 20 50 35 00 00  Y.6.Q.\..). P5..
0040  4e 45 39 6d 5a 61 6a 67:1f 16 3a 28 66 c0 f7 5d  NE9mZajg..:(f..]
0050  36 4c 5c 34 af c3 4b 9f:c1 46 ab 70 00 00 00 00  6L\4..K..F.p....
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:48 e8 af dc 7f 00 00 00  LMEMp...H.......
0080  00 00 00 00 00 00 00 00:00 00 00 00 00 00 00 00  ................
0090  00 00 00 00 00 00 00 00:2e 7c 1b 78 f6 56 00 0c  .........|.x.V..
00a0  1c 00 00 00 94 00 00 00:00 00 00 00 20 00 00 00  ............ ...
00b0  00 00 00 00 00 00 00 00:00 00 00 00 74 00 65 00  ............t.e.
00c0  2d 00 45 00 43 00 44 00:                          -.E.C.D.
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

© 2019 Microsoft Corporation. All rights reserved.

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6c06a68d743d0b8441bff2941f7abca84cc913f599a36...
    Session ID Length: 32
    Session ID: 503500004e45396d5a616a671f163a2866c0f75d364c5c34...
    Cipher Suite: TLS_NULL_WITH_NULL_NULL (0x0000)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.826424	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
3	0.830865	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
4	0.835711	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
5	0.900255	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
6	0.911075	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
8	0.921109	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
9	0.928376	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

14.2.4.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

14.2.5 Test 5.1 - FCS_TLSC_EXT.1.1 for DTLS

14.2.5.1 Setup

The setup is identical to the one defined for Test 1.

14.2.5.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.5.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *WebServer.exe* intercepts the outgoing Server Hello and changes the DTLS version to a non-supported one, setting the version bytes to 0xfe 0x06. The corresponding code snippet where the modification is performed is shown in the next block:

```
// Change TLS version number to bogus value
else if ((21 == iSFR && 5 == iTest && 'g' == cSubTest )
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.1")) )
{
    // first protocol msg in DTLS is not the server hello
    if ((IS_DTLS(dwProtocol) && handshakeLoopCounter == 1) ||
        (!IS_DTLS(dwProtocol) && fInitContext))
    {
        printf("\nFCS_TLSC_EXT.1.1 Test 5-1: Modifying the TLS version in the
            ServerHello\n");
        targetLocation = !IS_DTLS(dwProtocol) ? 0xA : 0x1A;
        printf("32 bytes of output before modification (TLS minor version starts
            at 0x%x)\n",
            targetLocation);
        PrintHexDump(32, (PBYTE)OutBuffers[0].pvBuffer);
        pbTemp = (PBYTE)OutBuffers[0].pvBuffer;
        \*(pbTemp + targetLocation) = 6; // change TLS minor version to bogus
            value of 6

        printf("\n32 bytes of output after modification\n\n");
        PrintHexDump(32, (PBYTE)OutBuffers[0].pvBuffer);
    }
}
```

The server´s output indicates the concrete modification that was performed:

```
FCS_TLSC_EXT.1.1 Test 5-1: Modifying the TLS version in the ServerHello
32 bytes of output before modification (TLS minor version starts at 0x1a)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 56 4d d5  Q.......Q..[.VM.

32 bytes of output after modification

0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe 06 5b f5 56 4d d5  Q.......Q..[.VM.
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Server Hello can be appreciated in the next picture:

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: Unknown (0xfe06)
  ▸ Random: 5bf5564dd5820ebb103e4070d6a103992b22c51dfa6f5520...
```

© 2019 Microsoft Corporation. All rights reserved.

The OS does not finish the handshake, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
3	4.438646	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
4	4.443008	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
5	4.449041	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
6	4.454697	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
7	4.460713	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
9	4.500241	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
10	4.530238	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090331
(SEC_E_ALGORITHM_MISMATCH)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090331 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified Server Hello is shown next, both as server's output and network capture:

```
FCS_TLSC_EXT.1.1 Test 5-1: Modifying the TLS version in the ServerHello
32 bytes of output before modification (TLS minor version starts at 0x1a)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c0 aa 3e  Q.......Q..[...>

32 bytes of output after modification

0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe 06 5b f6 c0 aa 3e  Q.......Q..[...>
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	207	Client Hello
2	0.004165	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009791	10.10.20.52	10.10.20.40	DTLSv1.0	239	Client Hello
4	0.023195	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.028556	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.042800	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.060432	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

```
▸ Frame 4: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 54041
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: Unknown (0xfe06)
      ▸ Random: 5bf6c0aa3e2c584c9540f4ad68fb5f0844e5f5b5f04e8a3a...
        Session ID Length: 32
        Session ID: 8d3c00000ebff3717e9e4d0b1e63fdeed238eec00b98b567...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	207	Client Hello
2	0.004165	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.009791	10.10.20.52	10.10.20.40	DTLSv1.0	239	Client Hello
4	0.023195	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.028556	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.042800	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.060432	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

It should be noted that Wireshark is not able to correctly detect the DTLS protocol version, wrongly defaulting to 1.0 even though 1.2 is being used.

14.2.5.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.1** activity.

14.2.6 Test 5.2 - FCS_TLSC_EXT.1.1 for DTLS

14.2.6.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

The setup is identical to the one defined for Test 1.

14.2.6.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.6.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the outgoing Server Hello and modifies the last byte in the nonce. The corresponding code snippet where the modification is performed is shown in the next block:

```
// Modify one byte in the server hello nonce/random
else if ((21 == iSFR && 5 == iTest && 'a' == cSubTest && fInitContext)
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.2")))
{
    if ((IS_DTLS(dwProtocol) && handshakeLoopCounter == 1) ||
        (!IS_DTLS(dwProtocol) && fInitContext))
    {
        printf("\nFCS_TLSC_EXT.1.1 Test 5-2: Modifying the Server random\n");
        targetLocation = !IS_DTLS(dwProtocol) ? 0xf : 0x20;
        printf("\n48 bytes of output before modification (Server random starts
            at 0x%x)\n",
            targetLocation);
        PrintHexDump(48, (PBYTE)OutBuffers[0].pvBuffer);

        pbTemp = (PBYTE)OutBuffers[0].pvBuffer;
        *(pbTemp + targetLocation) = ~*(pbTemp + targetLocation); // change a
            single byte

        printf("\n48 bytes of output after modification (changed byte is at
            0x%x)\n", targetLocation);
        PrintHexDump(48, (PBYTE)OutBuffers[0].pvBuffer);
    }
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
FCS_TLSC_EXT.1.1 Test 5-2: Modifying the Server random

48 bytes of output before modification (Server random starts at 0x20)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 57 36 5d  Q.......Q..[.W6]
0020  d8 4b 07 ef 6a a6 19 ac:03 49 20 ab 1b a5 d4 cf  .K..j....I .....

48 bytes of output after modification (changed byte is at 0x20)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 57 36 5d  Q.......Q..[.W6]
0020  27 4b 07 ef 6a a6 19 ac:03 49 20 ab 1b a5 d4 cf  'K..j....I .....
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Server Hello can be appreciated in the next picture:

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.0 (0xfeff)
  ▾ Random: 5bf557365d274b07ef6aa619ac034920ab1ba5d4cfaaf423...
      GMT Unix Time: Nov 21, 2018 14:01:42.000000000 CET
      Random Bytes: 5d274b07ef6aa619ac034920ab1ba5d4cfaaf42300a19453...
    Session ID Length: 32
    Session ID: 7307000015376b1dfd6dff8b0de3a14dba68a9b85a927caa...
```

The OS does not finish the handshake, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
4	20.061916	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
5	20.067137	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
6	20.069631	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
7	20.072258	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
8	20.083970	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
9	20.092201	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
10	20.105323	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090304
(SEC_E_INTERNAL_ERROR)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090304 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified Server Hello is shown next, both as server's output and network capture:

© 2019 Microsoft Corporation. All rights reserved.

```
FCS_TLSC_EXT.1.1 Test 5-2: Modifying the Server random

48 bytes of output before modification (Server random starts at 0x20)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c0 f2 99  Q.......Q..[....
0020  c0 cf e7 e7 9d 4a b5 ef:f7 a7 2d 02 55 b7 d1 7a  .....J....-.U..z

48 bytes of output after modification (changed byte is at 0x20)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c0 f2 99  Q.......Q..[....
0020  3f cf e7 e7 9d 4a b5 ef:f7 a7 2d 02 55 b7 d1 7a  ?....J....-.U..z
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

No.	Time	Source	Destination	Protocol	Length	Info
7	9.876506	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
8	9.881109	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
9	9.882687	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
10	9.894049	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
11	9.896597	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
12	9.901679	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
13	9.906191	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

```
▸ Frame 10: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 59615
▾ Datagram Transport Layer Security
  ▾ DTLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 0
      Sequence Number: 1
      Length: 93
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 81
        Message Sequence: 1
        Fragment Offset: 0
        Fragment Length: 81
        Version: DTLS 1.2 (0xfefd)
      ▾ Random: 5bf6c0f2993fcfe7e79d4ab5eff7a72d0255b7d17ad2a1b7...
          GMT Unix Time: Nov 22, 2018 15:45:06.000000000 CET
          Random Bytes: 993fcfe7e79d4ab5eff7a72d0255b7d17ad2a1b774c4654e...
        Session ID Length: 32
        Session ID: 3b030000c150dcda2376b469b37cdedafc2133d6ef6e92ba...
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
```

```
0020  14 34 01 b8 e8 df 00 72  3c f3 16 fe fd 00 00 00   ·4·····r <·......
0030  00 00 00 00 01 00 5d 02  00 00 51 00 01 00 00 00   ......]· ··Q·····
0040  00 00 51 fe fd 5b f6 c0  f2 99 3f cf e7 e7 9d 4a   ··Q··[·· ··?····J
0050  b5 ef f7 a7 2d 02 55 b7  d1 7a d2 a1 b7 74 c4 65   ·····-U· ·z···t·e
0060  4e 63 35 44 26 20 3b 03  00 00 c1 50 dc da 23 76   Nc5D& ;· ···P··#v
0070  b4 69 b3 7c de da fc 21  33 d6 ef 6e 92 ba 25 49   ·i·|···! 3··n··%I
0080  26 75 16 a5 d9 89 00 2f  00 00 09 00 17 00 00 ff   &u·····/ ........
0090  01 00 01 00                                        ····
```

The network traffic capture also shows no application data being transmitted in this case:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
7	9.876506	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
8	9.881109	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
9	9.882687	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
10	9.894049	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
11	9.896597	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
12	9.901679	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
13	9.906191	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

14.2.6.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.2** activity.

14.2.7 Test 5.3 - FCS_TLSC_EXT.1.1 for DTLS

14.2.7.1 Setup

The setup is identical to the one defined for Test 1.

14.2.7.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.7.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the outgoing Server Hello so that the server selects a cipher suite that was not presented by the client. The corresponding code snippet where the modification is performed is shown in the next block:

```
else if ((21 == iSFR && 5 == iTest && 'b' == cSubTest && fInitContext)
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.3")))
{
    if ((IS_DTLS(dwProtocol) && handshakeLoopCounter == 1) ||
        (!IS_DTLS(dwProtocol) && fInitContext))
    {
        printf("\nFCS_TLSC_EXT.1.1 Test 5-3: Change cipher suite to deprecated
            TLS_RSA_WITH_RC4_128_SHA {0x00, 0x05}\n");
        targetLocation = !IS_DTLS(dwProtocol) ? 0x4c : 0x5c;
        printf("\n128 bytes of output before modification (cipher suite at
            0x%x)\n", targetLocation);
        PrintHexDump(128, (PBYTE)OutBuffers[0].pvBuffer);
        pbTemp = (PBYTE)OutBuffers[0].pvBuffer;
        \*(pbTemp + targetLocation) = 0x00;
        \*(pbTemp + targetLocation + 1) = 0x05;
```

© 2019 Microsoft Corporation. All rights reserved.

```
        printf("\n128 bytes of output after modification (cipher suite at
            0x%x)\n", targetLocation);
        PrintHexDump(128, (PBYTE)OutBuffers[0].pvBuffer);
    }
}
```

The server´s output indicates the concrete modification that was performed for DTLS 1.0:

```
FCS_TLSC_EXT.1.1 Test 5-3: Change cipher suite to deprecated TLS_RSA_WITH_RC4_128_SHA {0x00, 0x05}

128 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 57 f1 1f  Q.......Q..[.W..
0020  ed d4 9f e4 03 9c 4a 18:cc bd 7c b0 1c 6a 4c c3  ......J...|..jL.
0030  22 00 23 62 49 57 96 8b:20 87 3e 20 43 47 00 00  ".#bIW.. .> CG..
0040  c4 d0 35 08 0f 10 84 65:57 c5 ae 68 ca 29 f8 08  ..5....eW..h.)..
0050  be 80 7f 37 e7 be fa ff:de af a2 43 00 2f 00 00  ...7.......C./..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:f8 ee f3 99 e4 00 00 00  LMEMp...........

128 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe ff 5b f5 57 f1 1f  Q.......Q..[.W..
0020  ed d4 9f e4 03 9c 4a 18:cc bd 7c b0 1c 6a 4c c3  ......J...|..jL.
0030  22 00 23 62 49 57 96 8b:20 87 3e 20 43 47 00 00  ".#bIW.. .> CG..
0040  c4 d0 35 08 0f 10 84 65:57 c5 ae 68 ca 29 f8 08  ..5....eW..h.)..
0050  be 80 7f 37 e7 be fa ff:de af a2 43 00 05 00 00  ...7.......C....
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:f8 ee f3 99 e4 00 00 00  LMEMp...........
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Server Hello can be appreciated in the next picture:

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.0 (0xfeff)
  ▸ Random: 5bf557f11fedd49fe4039c4a18ccbd7cb01c6a4cc3220023...
    Session ID Length: 32
    Session ID: 43470000c4d035080f10846557c5ae68ca29f808be807f37...
    Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

The selected cipher suite is not any of those sent in the Client Hello, shown next:

© 2019 Microsoft Corporation. All rights reserved.

```
▾ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 114
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 114
    Version: DTLS 1.0 (0xfeff)
  ▸ Random: 5bf5580249dd77a1b710db6e918e69a76f7051680e8dd409...
    Session ID Length: 0
    Cookie Length: 32
    Cookie: aa6900d2826593373bba712a71b65a30ce85e28bd8bdc2b1...
    Cipher Suites Length: 4
  ▾ Cipher Suites (2 suites)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Methods Length: 1
```

Note that only four out of the ten cipher suites in the ***Operational Guidance*** are sent, due to DTLS 1.0 not supporting the newest cipher suites.

The OS does not finish the handshake, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.003980	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.008728	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.043176	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.068579	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.092612	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.100316	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090331
(SEC_E_ALGORITHM_MISMATCH)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090331 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified Server Hello is shown next, both as server's output and network capture:

© 2019 Microsoft Corporation. All rights reserved.

```
FCS_TLSC_EXT.1.1 Test 5-3: Change cipher suite to deprecated TLS_RSA_WITH_RC4_128_SHA {0x00, 0x05}

128 bytes of output before modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c1 5d d0  Q.......Q..[..].
0020  75 ed e7 65 49 f3 04 a2:0a d2 30 52 6f bc ae 11  u..eI.....0Ro...
0030  a8 c0 e2 ae c3 a1 45 9e:26 0e 1e 20 15 38 00 00  ......E.&.. .8..
0040  e4 1f 4a 3f 75 88 0e d9:b4 09 b9 39 24 83 33 89  ..J?u......9$.3.
0050  52 d4 20 65 ce e5 a9 33:0c a1 80 e4 00 2f 00 00  R. e...3......./..
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:c8 e7 6f f9 35 00 00 00  LMEMp.....o.5...

128 bytes of output after modification (cipher suite at 0x5c)
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
0010  51 00 01 00 00 00 00 00:51 fe fd 5b f6 c1 5d d0  Q.......Q..[..].
0020  75 ed e7 65 49 f3 04 a2:0a d2 30 52 6f bc ae 11  u..eI.....0Ro...
0030  a8 c0 e2 ae c3 a1 45 9e:26 0e 1e 20 15 38 00 00  ......E.&.. .8..
0040  e4 1f 4a 3f 75 88 0e d9:b4 09 b9 39 24 83 33 89  ..J?u......9$.3.
0050  52 d4 20 65 ce e5 a9 33:0c a1 80 e4 00 05 00 00  R. e...3........
0060  09 00 17 00 00 ff 01 00:01 00 00 00 00 00 00 00  ................
0070  4c 4d 45 4d 70 00 00 00:c8 e7 6f f9 35 00 00 00  LMEMp.....o.5...
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 81
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6c15dd075ede76549f304a20ad230526fbcae11a8c0e2...
    Session ID Length: 32
    Session ID: 15380000e41f4a3f75880ed9b409b9392483338952d42065...
    Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
    Compression Method: null (0)
    Extensions Length: 9
```

The Client Hello presented in this case was:

© 2019 Microsoft Corporation. All rights reserved.

```
▾ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 172
    Message Sequence: 1
    Fragment Offset: 0
    Fragment Length: 172
    Version: DTLS 1.2 (0xfefd)
  ▸ Random: 5bf6c16ee4e9a351561bec5c3160a1f9887dcf5e77cc46b9...
    Session ID Length: 0
    Cookie Length: 32
    Cookie: edb3763d0c80f3571e2993fd2a5d8b3018a412d4bfc09587...
    Cipher Suites Length: 26
  ▾ Cipher Suites (13 suites)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Methods Length: 1
  ▸ Compression Methods (1 method)
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004631	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.010123	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.043828	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.077453	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
7	0.097678	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
8	0.103576	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

14.2.7.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.3** activity.

14.2.8 Test 5.4 - FCS_TLSC_EXT.1.1 for DTLS

14.2.8.1 Setup

The setup is identical to the one defined for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

14.2.8.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.8.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the outgoing Key Exchange handshake message and modifies its last byte, belonging to the signature block. The corresponding code snippet where the modification is performed is shown in the next block:

```
else if ((21 == iSFR && 5 == iTest && 'c' == cSubTest && fInitContext)
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.4")))
{
    if (IS_DTLS(dwProtocol) && handshakeLoopCounter == 1)
    {
        /** Modified by EE! \**/

        printf("\nFCS_TLSC_EXT.1.1 Test 5-4: Modify signature block in server
            key exchange. DTLS.\n");

        // Original implementation was considering the Server Key Exchange
            message as a separate
        // message from the server, not inside a Multiple Handshake Message
            (check Wireshark to
        // observe this)

        // We take the full length of the Multiple Handshake Message to modify
            the last byte
        // (corresponds to signature block in server key exchange).

        pwFieldLen = (PWORD)((PBYTE)OutBuffers[0].pvBuffer + 11);
        wFieldLen = *(pwFieldLen);
        wFieldLen = ntohs(wFieldLen);

        // 13 bytes for DTLS Multiple Handshake Messages wrapper
        pbTemp = (PBYTE)OutBuffers[0].pvBuffer + wFieldLen + 13;

        printf("\n0x20 bytes of output before modification\n");
        PrintHexDump(0x20, pbTemp - 0x20);

        *(pbTemp - 1) = ~*(pbTemp - 1); // change last byte

        printf("\n0x20 bytes of output after modification (changed byte at last
            position)\n");
        PrintHexDump(0x20, pbTemp - 0x20);
    }
```

The server's output indicates the concrete modification that was performed for DTLS 1.0:

© 2019 Microsoft Corporation. All rights reserved.

```
FCS_TLSC_EXT.1.1 Test 5-4: Modify signature block in server key exchange. DTLS.

0x20 bytes of output before modification
0000  4c b0 5e 4d f0 2f 5e e0:8e c8 36 69 12 24 8f e8  L.^M./^...6i.$..
0010  9e c4 00 2f 00 00 09 00:17 00 00 ff 01 00 01 00  .../............

0x20 bytes of output after modification (changed byte at last position)
0000  4c b0 5e 4d f0 2f 5e e0:8e c8 36 69 12 24 8f e8  L.^M./^...6i.$..
0010  9e c4 00 2f 00 00 09 00:17 00 00 ff 01 00 01 ff  .../............
Sent 106 handshake bytes to client
0000  16 fe ff 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The effect on the Renegotiation Info extension message can be appreciated in the next picture:

```
▸ Random: 5bf558ddeee7db2976c9d5cec6b2fb08dddc11e26ccf1473...
  Session ID Length: 32
  Session ID: 843b0000974d30807a804bfdb3084cb05e4df02f5ee08ec8...
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Compression Method: null (0)
  Extensions Length: 9
▸ Extension: extended_master_secret (len=0)
▾ Extension: renegotiation_info (len=1)
    Type: renegotiation_info (65281)
    Length: 1
  ▾ Renegotiation Info extension
    ▾ Renegotiation info extension length: 255
      ▾ [Expert Info (Error/Malformed): Vector length 255 is too large, truncating it to 0]
          [Vector length 255 is too large, truncating it to 0]
          [Severity level: Error]
          [Group: Malformed]

0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 86 1e b9 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ........ ·····(··
0020  14 34 01 b8 d3 ea 00 72  3c f3 16 fe ff 00 00 00   ·4·····r <·......
0030  00 00 00 00 01 00 5d 02  00 00 51 00 01 00 00 00   ......]· ··Q·····
0040  00 00 51 fe ff 5b f5 58  dd ee e7 db 29 76 c9 d5   ··Q··[·X ····)v··
0050  ce c6 b2 fb 08 dd dc 11  e2 6c cf 14 73 41 67 3e   ........ ·l··sAg>
0060  6f cb 63 73 4e 20 84 3b  00 00 97 4d 30 80 7a 80   o·csN ·; ···M0·z·
0070  4b fd b3 08 4c b0 5e 4d  f0 2f 5e e0 8e c8 36 69   K···L·^M ·/^···6i
0080  12 24 8f e8 9e c4 00 2f  00 00 09 00 17 00 00 ff   ·$·····/ ........
0090  01 00 01 ff                                        ····
```

The OS does not finish the handshake after the Server Hello message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.003795	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.005840	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.009544	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.012538	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
7	0.017030	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
8	0.025491	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
ISC returned: 0x80090326
(SEC_E_ILLEGAL_MESSAGE)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090326 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified packet is shown next, both as server's output and network capture:

```
FCS_TLSC_EXT.1.1 Test 5-4: Modify signature block in server key exchange. DTLS.

0x20 bytes of output before modification
0000  a7 70 64 bd 98 4d 08 3f:f8 f3 e2 82 bd 12 69 3e  .pd..M.?......i>
0010  48 73 00 2f 00 00 09 00:17 00 00 ff 01 00 01 00  Hs./............

0x20 bytes of output after modification (changed byte at last position)
0000  a7 70 64 bd 98 4d 08 3f:f8 f3 e2 82 bd 12 69 3e  .pd..M.?......i>
0010  48 73 00 2f 00 00 09 00:17 00 00 ff 01 00 01 ff  Hs./............
Sent 106 handshake bytes to client
0000  16 fe fd 00 00 00 00 00:00 00 01 00 5d 02 00 00  ............]...
Message fragment handling (continue)
```

The corresponding modified Server Hello is shown next:

```
▸ Random: 5bf6c1c150ebda4c51db1782a6febbec76e6442bfd4ddb8f...
  Session ID Length: 32
  Session ID: ef490000bb74e6cb82b354d16fb2a77064bd984d083ff8f3...
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Compression Method: null (0)
  Extensions Length: 9
▸ Extension: extended_master_secret (len=0)
▾ Extension: renegotiation_info (len=1)
    Type: renegotiation_info (65281)
    Length: 1
  ▾ Renegotiation Info extension
    ▾ Renegotiation info extension length: 255
      ▾ [Expert Info (Error/Malformed): Vector length 255 is too large, truncating it to 0]
          [Vector length 255 is too large, truncating it to 0]
          [Severity level: Error]
          [Group: Malformed]

0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 86 3d 1a 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ··=····· ·····(··
0020  14 34 01 b8 d5 dd 00 72  3c f3 16 fe fd 00 00 00   ·4·····r <·......
0030  00 00 00 00 01 00 5d 02  00 00 51 00 01 00 00 00   ......]· ··Q·····
0040  00 00 51 fe fd 5b f6 c1  c1 50 eb da 4c 51 db 17   ··Q··[·· ·P··LQ··
0050  82 a6 fe bb ec 76 e6 44  2b fd 4d db 8f 60 5f a8   ·····v·D +·M··`_·
0060  6c 08 f8 a9 88 20 ef 49  00 00 bb 74 e6 cb 82 b3   l···· ·I ···t····
0070  54 d1 6f b2 a7 70 64 bd  98 4d 08 3f f8 f3 e2 82   T·o··pd· ·M·?····
0080  bd 12 69 3e 48 73 00 2f  00 00 09 00 17 00 00 ff   ··i>Hs·/ ........
0090  01 00 01 ff                                        ····
```

The network traffic capture also shows no application data being transmitted in this case:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004500	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.006628	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.011521	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.017483	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
7	0.044144	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
8	0.054474	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done

14.2.8.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.4** activity.

14.2.9 Test 5.5 - FCS_TLSC_EXT.1.1 for DTLS

14.2.9.1 Setup

The setup is identical to the one defined for Test 1.

14.2.9.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.9.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the Server Finished handshake message and modifies one of its bytes. The corresponding code snippet where the modification is performed is shown in the next block:

```
else if ((21 == iSFR && 5 == iTest && 'e' == cSubTest && !fInitContext)
    || (NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.5") &&
        handshakeLoopCounter == 6))
{
    /** Modified by EE! **/

    // Just changed the handshakeLoopCounter check from 8 to 6 to account for
        the fact that Server
    // Hello, Certificate and Server Key Exchange are now part of the same
        Multiple Handshake Message

    // DTLS - 8th handshake message is the "Encrypted Handshake Message"
        (Wireshark) which is the
    // server finished message
    dwOutBufferSize = OutBuffers[0].cbBuffer;
    printf("\nModifying a byte in the Server Finished handshake message\n");
```

© 2019 Microsoft Corporation. All rights reserved.

```
    printf("\n0x%x bytes of output before modification\n", dwOutBufferSize);
    PrintHexDump(dwOutBufferSize, (PBYTE)OutBuffers[0].pvBuffer);
    printf("Modifying byte at offset 0x%x in the output buffer", dwOutBufferSize
        - 6);
    pbTemp = (PBYTE)OutBuffers[0].pvBuffer + dwOutBufferSize - 6;
    *pbTemp = ~*(pbTemp );
    printf("\n0x%x bytes of output after modification\n", dwOutBufferSize);
    PrintHexDump(dwOutBufferSize, (PBYTE)OutBuffers[0].pvBuffer);

}
```

The server´s output indicates the concrete modification that was performed:

```
Modifying a byte in the Server Finished handshake message

0x4d bytes of output before modification
0000  16 fe ff 00 01 00 00 00:00 00 00 00 40 f0 58 a4  ............@.X.
0010  82 cd 69 97 4d f6 73 52:e7 3f 98 4e 25 ab 9b 52  ..i.M.sR.?.N...R
0020  99 c2 63 f5 6b c4 35 17:b0 0e 62 72 5c 5d f1 85  ..c.k.5...br\]..
0030  e8 d5 c8 30 0d 61 54 f7:99 b0 f3 f9 8c fc 6c 05  ...0.aT.......l.
0040  10 07 f2 40 45 60 63 f7:16 95 b1 72 40           ...@E`c....r@
Modifying byte at offset 0x47 in the output buffer
0x4d bytes of output after modification
0000  16 fe ff 00 01 00 00 00:00 00 00 00 40 f0 58 a4  ............@.X.
0010  82 cd 69 97 4d f6 73 52:e7 3f 98 4e 25 ab 9b 52  ..i.M.sR.?.N...R
0020  99 c2 63 f5 6b c4 35 17:b0 0e 62 72 5c 5d f1 85  ..c.k.5...br\]..
0030  e8 d5 c8 30 0d 61 54 f7:99 b0 f3 f9 8c fc 6c 05  ...0.aT.......l.
0040  10 07 f2 40 45 60 63 08:16 95 b1 72 40           ...@E`c....r@
Sent 77 handshake bytes to client
0000  16 fe ff 00 01 00 00 00:00 00 00 00 40 f0 58 a4  ............@.X.
Finished SSPI Negotiation with client
calling decrypt
Incomplete message, getting more.
```

The effect on the Server Finished handshake message (which is encrypted) can be appreciated in the next picture:

```
└   18 11.799704   10.10.20.40   10.10.20.52   DTLSv1.0   119 Encrypted Handshake Message

▸ Frame 18: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 54956
▾ Datagram Transport Layer Security
  ▾ Record Layer
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 1
      Sequence Number: 0
      Length: 64
      Handshake Protocol

0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 69 27 88 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ·i'····· ·····(··
0020  14 34 01 b8 d6 ac 00 55  3c d6 16 fe ff 00 01 00   ·4·····U <·......
0030  00 00 00 00 00 00 40 f0  58 a4 82 cd 69 97 4d f6   ......@· X···i·M·
0040  73 52 e7 3f 98 4e 25 ab  9b 52 99 c2 63 f5 6b c4   sR·?·N%· ·R··c·k·
0050  35 17 b0 0e 62 72 5c 5d  f1 85 e8 d5 c8 30 0d 61   5···br\] ·····0·a
0060  54 f7 99 b0 f3 f9 8c fc  6c 05 10 07 f2 40 45 60   T····· l····@E`
0070  63 08 16 95 b1 72 40                               c·····r@
```

© 2019 Microsoft Corporation. All rights reserved.

The OS does not finish the handshake after the Server Finished message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
7	11.613885	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
8	11.621332	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
9	11.632871	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
10	11.637493	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
11	11.640917	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
12	11.644560	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
13	11.656024	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
14	11.747454	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
15	11.762530	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
16	11.771331	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
17	11.783672	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
18	11.799704	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be seen on the output of the tool used to perform the connection:

```
ISC returned: 0x80090326
(SEC_E_ILLEGAL_MESSAGE)
SSPIFlags = c11c
SSPIOutFlags = c11c
Nothing in OutBuffers to send
**** Error 0x80090326 returned by InitializeSecurityContext (2)
Deleting security context.
Error performing handshake
Done
```

The same results are obtained for DTLS 1.2. The corresponding modified Server Finished handshake message is shown next, both as server's output and network capture:

```
Modifying a byte in the Server Finished handshake message

0x4d bytes of output before modification
0000  16 fe fd 00 01 00 00 00:00 00 00 00 40 2f fd 4a  ............@/.J
0010  cf 97 9d f2 30 6f 04 65:72 05 3f 0e 14 96 5b 18  ....0o.er.?...[.
0020  db 32 0d af 2e 8c 27 31:c3 0b 44 ae e2 df 57 97  .2....'1..D...W.
0030  c5 90 9c 98 e6 4b 04 c9:3c ea 84 f2 c5 d2 fd b5  .....K..<.......
0040  0f 96 8e 57 76 61 c3 60:f1 54 9b 91 bb           ...Wva.`.T...
Modifying byte at offset 0x47 in the output buffer
0x4d bytes of output after modification
0000  16 fe fd 00 01 00 00 00:00 00 00 00 40 2f fd 4a  ............@/.J
0010  cf 97 9d f2 30 6f 04 65:72 05 3f 0e 14 96 5b 18  ....0o.er.?...[.
0020  db 32 0d af 2e 8c 27 31:c3 0b 44 ae e2 df 57 97  .2....'1..D...W.
0030  c5 90 9c 98 e6 4b 04 c9:3c ea 84 f2 c5 d2 fd b5  .....K..<.......
0040  0f 96 8e 57 76 61 c3 9f:f1 54 9b 91 bb           ...Wva...T...
Sent 77 handshake bytes to client
0000  16 fe fd 00 01 00 00 00:00 00 00 00 40 2f fd 4a  ............@/.J
Finished SSPI Negotiation with client
calling decrypt
Incomplete message, getting more.
```

© 2019 Microsoft Corporation. All rights reserved.

```
12 0.138571    10.10.20.40    10.10.20.52    DTLSv1.2    119 Encrypted Handshake Message

▸ Frame 12: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 54750
▾ Datagram Transport Layer Security
  ▾ Record Layer
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 1
      Sequence Number: 0
      Length: 64
      Handshake Protocol

0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 69 3d 24 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ·i=$···· ·····(··
0020  14 34 01 b8 d5 de 00 55  3c d6 16 fe fd 00 01 00   ·4·····U <·......
0030  00 00 00 00 00 00 40 2f  fd 4a cf 97 9d f2 30 6f   ......@/ ·J····0o
0040  04 65 72 05 3f 0e 14 96  5b 18 db 32 0d af 2e 8c   ·er·?··· [··2····
0050  27 31 c3 0b 44 ae e2 df  57 97 c5 90 9c 98 e6 4b   '1··D··· W......K
0060  04 c9 3c ea 84 f2 c5 d2  fd b5 0f 96 8e 57 76 61   ··<····· ·····Wva
0070  c3 9f f1 54 9b 91 bb                               ···T···
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.005056	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.007703	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.010175	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.013796	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.017531	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
7	0.060144	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.067192	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.069972	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.070420	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.093528	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.138571	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.9.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.5** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.5** activity.

14.2.10 Test 5.6 - FCS_TLSC_EXT.1.1 for DTLS

14.2.10.1 Setup

The setup is identical to the one defined for Test 1.

14.2.10.2 Procedure

The procedure is identical to the one defined for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

14.2.10.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *WebServer.exe* intercepts the server's message after Change Cipher Spec, changing the value of one if its bytes. The corresponding code snippet where the modification is performed is shown in the next screenshot:

```
// windows mobile 10 FCS_TLSC_EXT.1.1 <send garbled data after changechipherspec>
if ( (21 == iSFR && 5 == iTest && 'f' == cSubTest)
    || ( NULL != pszTest && !strcmp(pszTest, "TLSC_EXT.1.1.5.6")))
{
    printf("\nGarbling first message from server to client\n");
    printf("0x80 bytes of output before modification\n");
    PrintHexDump(0x80, (PBYTE)IoBuffer);
    printf("Modifying byte at offset 0x%x in the output buffer\n",
        Buffers[0].cbBuffer +
        Buffers[1].cbBuffer + Buffers[2].cbBuffer - 12);
    pbTemp = (PBYTE)IoBuffer + Buffers[0].cbBuffer + Buffers[1].cbBuffer +
        Buffers[2].cbBuffer - 12;
    \*pbTemp = ~\*pbTemp;
    printf("0x80 bytes of output after modification\n");
    PrintHexDump(0x80, (PBYTE)IoBuffer);
}
```

The server´s output indicates that the modification was performed:

```
Garbling first message from server to client
0x80 bytes of output before modification
0000  17 fe ff 00 01 00 00 00:00 00 01 00 50 a4 d4 e4  ............P...
0010  1a 2a fa 3b 7c 8d 64 db:95 e2 9e 16 3a 37 b5 2c  .*.;|.d.....:7.,
0020  f9 38 cc 42 e2 30 af be:87 d9 74 c2 64 68 a4 0f  .8.B.0....t.dh..
0030  a2 ae 19 83 ac 46 3a 3e:89 1b da 3d e1 20 0f a0  .....F:>...=. ..
0040  8b 14 9e 18 21 20 23 9c:9f e1 bc 39 4c 94 a0 d0  ....! #....9L...
0050  78 ee d7 5b f8 32 73 bf:c8 cf 11 dd c7 00 15 34  x..[.2s........4
0060  c8 56 e6 cf 6f 61 68 ba:fe 1b 05 eb 1c b6 a3 5f  .V..oah........_
0070  21 0b 0b 0b 0b 0b 0b 0b:0b 0b 0b 0b 0b a6 e0 6f  !...............o
Modifying byte at offset 0x51 in the output buffer
0x80 bytes of output after modification
0000  17 fe ff 00 01 00 00 00:00 00 01 00 50 a4 d4 e4  ............P...
0010  1a 2a fa 3b 7c 8d 64 db:95 e2 9e 16 3a 37 b5 2c  .*.;|.d.....:7.,
0020  f9 38 cc 42 e2 30 af be:87 d9 74 c2 64 68 a4 0f  .8.B.0....t.dh..
0030  a2 ae 19 83 ac 46 3a 3e:89 1b da 3d e1 20 0f a0  .....F:>...=. ..
0040  8b 14 9e 18 21 20 23 9c:9f e1 bc 39 4c 94 a0 d0  ....! #....9L...
0050  78 11 d7 5b f8 32 73 bf:c8 cf 11 dd c7 00 15 34  x..[.2s........4
0060  c8 56 e6 cf 6f 61 68 ba:fe 1b 05 eb 1c b6 a3 5f  .V..oah........_
0070  21 0b 0b 0b 0b 0b 0b 0b:0b 0b 0b 0b 0b a6 e0 6f  !...............o
```

The effect on this message can be appreciated in the next picture:

© 2019 Microsoft Corporation. All rights reserved.

24	11.284361	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.285895	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.287208	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
▸ Frame 24: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 54347
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 1
      Sequence Number: 1
      Length: 80
      Encrypted Application Data: a4d4e41a2afa3b7c8d64db95e29e163a37b52cf938cc42e2...
```

```
0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 79 1f 1c 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ·y····· ·····(··
0020  14 34 01 b8 d4 4b 00 65  3c e6 17 fe ff 00 01 00   ·4···K·e <·····
0030  00 00 00 00 01 00 50 a4  d4 e4 1a 2a fa 3b 7c 8d   ......P· ···*·;|·
0040  64 db 95 e2 9e 16 3a 37  b5 2c f9 38 cc 42 e2 30   d·····:7 ·,·8·B·0
0050  af be 87 d9 74 c2 64 68  a4 0f a2 ae 19 83 ac 46   ····t·dh ·····F
0060  3a 3e 89 1b da 3d e1 20  0f a0 8b 14 9e 18 21 20   :>···=·  ·····!
0070  23 9c 9f e1 bc 39 4c 94  a0 d0 78 11 d7 5b f8 32   #····9L· ··x··[·2
0080  73 bf c8 cf 11 dd c7                               s·····
```

The OS interrupts the handshake once it receives the modified message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004266	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.007532	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.010124	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.013115	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.021563	10.10.20.40	10.10.20.52	DTLSv1.0	110	Certificate (Reassembled)
7	0.027325	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.030281	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.032060	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.032486	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.051653	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.056122	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.247601	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.284361	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.285895	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.287208	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

This fact can also be seen on the output of the tool used to perform the connection:

```
93 bytes of (encrypted) application data received
0000  17 fe ff 00 01 00 00 00:00 00 01 00 50 a4 d4 e4  ............P...
0010  1a 2a fa 3b 7c 8d 64 db:95 e2 9e 16 3a 37 b5 2c  .*.;|.d.....:7.,
0020  f9 38 cc 42 e2 30 af be:87 d9 74 c2 64 68 a4 0f  .8.B.0....t.dh..
0030  a2 ae 19 83 ac 46 3a 3e:89 1b da 3d e1 20 0f a0  .....F:>...=. ..
0040  8b 14 9e 18 21 20 23 9c:9f e1 bc 39 4c 94 a0 d0  ....! #....9L...
0050  78 11 d7 5b f8 32 73 bf:c8 cf 11 dd c7           x..[.2s......

DecryptMessage returned: 0x80090330
**** Error 0x80090330 returned by DecryptMessage
Error fetching file from server
Done
```

© 2019 Microsoft Corporation. All rights reserved.

The same results are obtained for DTLS 1.2. The corresponding modified message is shown next, both as server's output and network capture:

```
Garbling first message from server to client
0x80 bytes of output before modification
0000  17 fe fd 00 01 00 00 00:00 00 01 00 50 1e 0e 8a  ............P...
0010  64 a0 44 73 e3 d1 aa 74:9d ba dc 05 ca 68 49 5f  d.Ds...t.....hI_
0020  bf 52 d6 fb 40 e6 7b 0c:51 dd c5 ac 14 d9 96 b6  .R..@.{.Q.......
0030  01 6f 04 6a 78 79 4e 6f:7b f6 5e 76 df 7a 71 5c  .o.jxyNo{.^v.zq\
0040  e8 c4 48 b5 4b 42 50 93:f5 59 6b 29 75 be b6 7a  ..H.KBP..Yk)u..z
0050  74 53 68 3a d2 0d d5 3b:66 d2 a6 21 ce 00 9c 48  tSh:...;f..!...H
0060  22 26 75 c7 12 f7 5d 87:27 0a 36 37 8d 45 45 47  "&u...].'.67.EEG
0070  8d 0b 0b 0b 0b 0b 0b 0b:0b 0b 0b 0b 0b b5 93 66  ...............f
Modifying byte at offset 0x51 in the output buffer
0x80 bytes of output after modification
0000  17 fe fd 00 01 00 00 00:00 00 01 00 50 1e 0e 8a  ............P...
0010  64 a0 44 73 e3 d1 aa 74:9d ba dc 05 ca 68 49 5f  d.Ds...t.....hI_
0020  bf 52 d6 fb 40 e6 7b 0c:51 dd c5 ac 14 d9 96 b6  .R..@.{.Q.......
0030  01 6f 04 6a 78 79 4e 6f:7b f6 5e 76 df 7a 71 5c  .o.jxyNo{.^v.zq\
0040  e8 c4 48 b5 4b 42 50 93:f5 59 6b 29 75 be b6 7a  ..H.KBP..Yk)u..z
0050  74 ac 68 3a d2 0d d5 3b:66 d2 a6 21 ce 00 9c 48  t.h:...;f..!...H
0060  22 26 75 c7 12 f7 5d 87:27 0a 36 37 8d 45 45 47  "&u...].'.67.EEG
0070  8d 0b 0b 0b 0b 0b 0b 0b:0b 0b 0b 0b 0b b5 93 66  ...............f
```

No.	Time	Source	Destination	Protocol	Length	Info
26	31.568436	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
27	31.573975	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
36	43.229053	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
37	43.273504	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
38	43.275660	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
39	43.278218	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

```
▸ Frame 37: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 54752
▾ Datagram Transport Layer Security
  ▾ DTLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 1
      Sequence Number: 1
      Length: 80
      Encrypted Application Data: 1e0e8a64a04473e3d1aa749dbadc05ca68495fbf52d6fb40...
```

```
0000  d4 6e 0e 0d fd 56 08 00  27 85 46 ae 08 00 45 00   ·n···V·· '·F···E·
0010  00 79 3d 43 00 00 80 11  00 00 0a 0a 14 28 0a 0a   ·y=C···· ·····(··
0020  14 34 01 b8 d5 e0 00 65  3c e6 17 fe fd 00 01 00   ·4·····e <·......
0030  00 00 00 00 01 00 50 1e  0e 8a 64 a0 44 73 e3 d1   ......P· ··d·Ds··
0040  aa 74 9d ba dc 05 ca 68  49 5f bf 52 d6 fb 40 e6   ·t·····h I_·R··@·
0050  7b 0c 51 dd c5 ac 14 d9  96 b6 01 6f 04 6a 78 79   {·Q····· ···o·jxy
0060  4e 6f 7b f6 5e 76 df 7a  71 5c e8 c4 48 b5 4b 42   No{·^v·z q\··H·KB
0070  50 93 f5 59 6b 29 75 be  b6 7a 74 ac 68 3a d2 0d   P··Yk)u· ·zt·h:··
0080  d5 3b 66 d2 a6 21 ce                               ·;f··!·
```

The network traffic capture also shows no application data being transmitted in this case:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
16	31.468347	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
17	31.472861	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
18	31.475349	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
19	31.478351	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
20	31.487548	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
21	31.490000	10.10.20.40	10.10.20.52	DTLSv1.2	110	Certificate (Reassembled)
22	31.518594	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
23	31.522276	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
24	31.525481	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
25	31.526006	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
26	31.568436	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
27	31.573975	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
36	43.229053	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
37	43.273504	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
38	43.275660	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
39	43.278218	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

14.2.10.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.6** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.6** activity.

14.2.11 Test 1 - FCS_TLSC_EXT.1.2 for DTLS

14.2.11.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = invalid.epoche.es

The listed certificates form a valid certification path. The server certificate is available in pfx format and the root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Windows Server 2016 Standard)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.20.*

Server machine shall contain the PowerShell script *W10DtlsServerAutomator.ps1* as well as the *WebServer.exe* tool also mentioned before.

Client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *Web-Client.exe* tool mentioned in the introduction to this section.

© 2019 Microsoft Corporation. All rights reserved.

Moreover, the client machine shall have the secure configuration enabled, according to the section *1.2 Configuration* of the ***Operational Guidance***. The *Wireshark* network analyzer shall be installed on the client machine.

14.2.11.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run the Powershell script *W10DtlsServerAutomator*.ps1 as follows:

```
.\W10DtlsServerAutomator.ps1 dtls1 #for DTLS 1.0
.\W10DtlsServerAutomator.ps1 dtls12 #for DTLS 1.2
```

- Select the option number 3 when the menu is prompt.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <dtls_version> (e.g. dlts1 for DTLS 1.0, dtls12 for DTLS 1.2 and <ip_server_address> and select the option number 12 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection from the client's output, server's output and the Wireshark capture.
- Repeat the steps above for both DTLS 1.0 and DTLS 1.2.

14.2.11.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name and SAN identifier (DNS Name) not matching the reference identifier test.epoche.es, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.009803	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.012925	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.016250	10.10.20.40	10.10.20.52	DTLSv1.0	116	Certificate (Reassembled)
7	0.057175	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.062501	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.065527	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.066064	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message

```
    Fragment Offset: 1071
    Fragment Length: 49
    Certificates Length: 1117
  ▾ Certificates (1117 bytes)
      Certificate Length: 1114
    ▾ Certificate: 308204563082023ea003020102020101300d06092a864886... (id-at-commonName=invalid.epoche.es,id-
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,id-at-org
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-subjectAltName)
              Extension Id: 2.5.29.17 (id-ce-subjectAltName)
            ▾ GeneralNames: 1 item
              ▾ GeneralName: dNSName (2)
                  dNSName: invalid.epoche.es
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
        encrypted: 8272e1187527691bbb7d8f7133d3c937e20a6d4547629f23...
```

The client interrupts the handshake, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004401	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006992	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.009803	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.012925	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.016250	10.10.20.40	10.10.20.52	DTLSv1.0	116	Certificate (Reassembled)
7	0.057175	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.062501	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.065527	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.066064	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.106036	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.111161	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=invalid.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

The same results are obtained for DTLS 1.2 when presenting the same certificate:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
11	25.233404	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
12	25.237795	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
13	25.240254	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
14	25.243262	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
15	25.246837	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
16	25.250180	10.10.20.40	10.10.20.52	DTLSv1.2	116	Certificate (Reassembled)
17	25.287235	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
18	25.292685	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
19	25.295925	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
20	25.296409	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
21	25.328911	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

```
    version: v3 (2)
    serialNumber: 1
  ▸ signature (sha256WithRSAEncryption)
  ▸ issuer: rdnSequence (0)
  ▸ validity
  ▾ subject: rdnSequence (0)
    ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,id-
      ▸ RDNSequence item: 1 item (id-at-countryName=ee)
      ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
      ▸ RDNSequence item: 1 item (id-at-localityName=ee)
      ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
      ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
      ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
  ▾ subjectPublicKeyInfo
    ▸ algorithm (rsaEncryption)
    ▸ subjectPublicKey: 3082010a0282010100cc6a2c34ef86f4a44ead7e58d70ad0...
  ▾ extensions: 1 item
    ▾ Extension (id-ce-subjectAltName)
        Extension Id: 2.5.29.17 (id-ce-subjectAltName)
      ▾ GeneralNames: 1 item
        ▾ GeneralName: dNSName (2)
            dNSName: invalid.epoche.es
```

The network traffic capture also shows no application data being transmitted in this case:

No.	Time	Source	Destination	Protocol	Length	Info
11	25.233404	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
12	25.237795	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
13	25.240254	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
14	25.243262	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
15	25.246837	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
16	25.250180	10.10.20.40	10.10.20.52	DTLSv1.2	116	Certificate (Reassembled)
17	25.287235	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
18	25.292685	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
19	25.295925	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
20	25.296409	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
21	25.328911	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
22	25.334517	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.11.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

14.2.12 Test 2 - FCS_TLSC_EXT.1.2 for DTLS

14.2.12.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

© 2019 Microsoft Corporation. All rights reserved.

- CN = EE Test Root CA
- CN = test.epoche.es

14.2.12.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.12.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents server certificates with a Common Name that matches the reference identifier test.epoche.es and a SAN identifier (DNS Name) that does not match, as shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
3	0.172234	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
4	0.177585	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
5	0.180006	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
6	0.182692	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
7	0.185549	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
8	0.188724	10.10.20.40	10.10.20.52	DTLSv1.0	113	Certificate (Reassembled)
9	0.214341	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
10	0.218020	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
11	0.234202	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
12	0.234205	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
13	0.246728	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
14	0.251859	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

```
Certificates Length: 1114
Certificates (1114 bytes)
  Certificate Length: 1111
  Certificate: 308204533082023ba003020102020101300d06092a864886... (id-at-commonName=test.epoche.es,id-at
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=test.epoche.es,id-at-organizationalUnitName=ee,id-at-orgar
          RDNSequence item: 1 item (id-at-countryName=ee)
          RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          RDNSequence item: 1 item (id-at-localityName=ee)
          RDNSequence item: 1 item (id-at-organizationName=ee)
          RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          RDNSequence item: 1 item (id-at-commonName=test.epoche.es)
      subjectPublicKeyInfo
      extensions: 1 item
        Extension (id-ce-subjectAltName)
          Extension Id: 2.5.29.17 (id-ce-subjectAltName)
        GeneralNames: 1 item
          GeneralName: dNSName (2)
            dNSName: invalid.epoche.es
```

The client interrupts the handshake, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.172234	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
4	0.177585	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
5	0.180006	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
6	0.182692	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
7	0.185549	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
8	0.188724	10.10.20.40	10.10.20.52	DTLSv1.0	113	Certificate (Reassembled)
9	0.214341	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
10	0.218020	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
11	0.234202	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
12	0.234205	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
13	0.246728	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
14	0.251859	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

The same results are obtained for DTLS 1.2 when presenting the same certificate:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004404	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.006871	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.009998	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.013429	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.025158	10.10.20.40	10.10.20.52	DTLSv1.2	113	Certificate (Reassembled)
7	0.053517	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.057805	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.059896	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.060377	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.127665	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.134008	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

```
▾ Certificates (1114 bytes)
    Certificate Length: 1111
  ▾ Certificate: 308204533082023ba00302010202010130 0d06092a864886... (id-at-commonName=test.epoche.es,id-at-orga
    ▾ signedCertificate
        version: v3 (2)
        serialNumber: 1
      ▸ signature (sha256WithRSAEncryption)
      ▸ issuer: rdnSequence (0)
      ▸ validity
      ▾ subject: rdnSequence (0)
        ▾ rdnSequence: 6 items (id-at-commonName=test.epoche.es,id-at-organizationalUnitName=ee,id-at-organizat
          ▸ RDNSequence item: 1 item (id-at-countryName=ee)
          ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          ▸ RDNSequence item: 1 item (id-at-localityName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          ▸ RDNSequence item: 1 item (id-at-commonName=test.epoche.es)
      ▸ subjectPublicKeyInfo
      ▾ extensions: 1 item
        ▾ Extension (id-ce-subjectAltName)
            Extension Id: 2.5.29.17 (id-ce-subjectAltName)
          ▾ GeneralNames: 1 item
            ▾ GeneralName: dNSName (2)
                dNSName: invalid.epoche.es
```

The network traffic capture also shows no application data being transmitted in this case:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004404	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.006871	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.009998	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.013429	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.025158	10.10.20.40	10.10.20.52	DTLSv1.2	113	Certificate (Reassembled)
7	0.053517	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.057805	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.059896	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.060377	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.127665	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.134008	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.12.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

14.2.13 Test 3 - FCS_TLSC_EXT.1.2 for DTLS

14.2.13.1 Setup

The setup is identical to the one defined for Test 2.

14.2.13.2 Procedure

The procedure is identical to the one defined for Test 2.

14.2.13.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name that matches the reference identifier test.epoche.es and no SAN extension, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
7	5.973234	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
8	5.977434	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
9	5.982913	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
10	5.990320	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
11	5.993410	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
12	5.996586	10.10.20.40	10.10.20.52	DTLSv1.0	204	Certificate (Reassembled)
13	6.019135	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
14	6.022535	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
15	6.025209	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
16	6.030251	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
17	6.060166	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
18	6.068359	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
34	17.190973	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
36	17.210824	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
37	17.216744	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
38	17.231376	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
Certificates Length: 1205
Certificates (1205 bytes)
  Certificate Length: 1202
  Certificate: 308204ae30820296a003020102020101300d06092a864886... (id-at-commonName=test.epoche.es,id-at-organizatio
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=test.epoche.es,id-at-organizationalUnitName=ee,id-at-organizationName=
          RDNSequence item: 1 item (id-at-countryName=ee)
          RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          RDNSequence item: 1 item (id-at-localityName=ee)
          RDNSequence item: 1 item (id-at-organizationName=ee)
          RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          RDNSequence item: 1 item (id-at-commonName=test.epoche.es)
      subjectPublicKeyInfo
      extensions: 4 items
        Extension (id-ce-basicConstraints)
          Extension Id: 2.5.29.19 (id-ce-basicConstraints)
          BasicConstraintsSyntax [0 length]
```

The connection is successfully performed, as shown in the network capture, where application data is being transmitted after the handshake:

No.	Time	Source	Destination	Protocol	Length	Info
7	5.973234	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
8	5.977434	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
9	5.982913	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
10	5.990320	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
11	5.993410	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
12	5.996586	10.10.20.40	10.10.20.52	DTLSv1.0	204	Certificate (Reassembled)
13	6.019135	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
14	6.022535	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
15	6.025209	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
16	6.030251	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
17	6.060166	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
18	6.068359	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
34	17.190973	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
36	17.210824	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
37	17.216744	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
38	17.231376	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

This fact can also be verified from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
77 bytes of (encrypted) application data received
0000  17 fe ff 00 01 00 00 00:00 00 02 00 40 3b 9a 06  ............@;..
0010  8b ad 10 af fd 5b a8 ab:f4 93 e6 45 fd 0b 72 85  .....[.....E..r.
0020  cb e9 56 56 ff 92 a0 c6:01 c2 d7 03 20 52 e5 41  ..VV........ R.A
0030  06 85 d8 85 6c 12 85 2c:cd 6f 32 16 35 03 42 06  ....l..,.o2.5.B.
0040  11 07 d7 bc ea 2d 16 2a:98 51 ed 1d 2a           .....-.*.Q..*

DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21           Hello, world!
```

The same results are obtained for DTLS 1.2 when presenting the same certificate:

No.	Time	Source	Destination	Protocol	Length	Info
5	6.230781	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
6	6.234809	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	6.236994	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
8	6.240392	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	6.243666	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	6.251428	10.10.20.40	10.10.20.52	DTLSv1.2	204	Certificate (Reassembled)
11	6.263175	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	6.272534	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	6.275341	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec

```
    Fragment Length: 137
    Certificates Length: 1205
  ▾ Certificates (1205 bytes)
      Certificate Length: 1202
    ▾ Certificate: 308204ae30820296a003020102020101300d06092a864886... (id-at-commonName=test.epoche.es,id-a
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=test.epoche.es,id-at-organizationalUnitName=ee,id-at-org
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=test.epoche.es)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 4 items
          ▾ Extension (id-ce-basicConstraints)
              Extension Id: 2.5.29.19 (id-ce-basicConstraints)
              BasicConstraintsSyntax [0 length]
```

The network traffic capture also shows application data being successfully transmitted in these cases:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
5	6.230781	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
6	6.234809	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	6.236994	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
8	6.240392	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	6.243666	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	6.251428	10.10.20.40	10.10.20.52	DTLSv1.2	204	Certificate (Reassembled)
11	6.263175	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	6.272534	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	6.275341	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	6.275724	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
15	6.301141	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
16	6.306597	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
29	17.900572	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
30	17.916840	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
31	17.922892	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
32	17.926130	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

14.2.13.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

14.2.14 Test 4 - FCS_TLSC_EXT.1.2 for DTLS

14.2.14.1 Setup

The setup is identical to the one defined for Test 1.

14.2.14.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.14.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name that does not match the reference identifier test.epoche.es and a SAN identifier (DNS Name) that matches, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.002862	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.005253	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.006777	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.007781	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.008718	10.10.20.40	10.10.20.52	DTLSv1.0	113	Certificate (Reassembled)
7	0.022914	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.030810	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.034178	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.034617	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.062983	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
    Certificates Length: 1114
  ▾ Certificates (1114 bytes)
      Certificate Length: 1111
    ▾ Certificate: 308204533082023ba003020102020101300d06092a864886... (id-at-commonName=invalid.epoche.es,id-at-organiz
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,id-at-organizationN
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-subjectAltName)
              Extension Id: 2.5.29.17 (id-ce-subjectAltName)
            ▾ GeneralNames: 1 item
              ▾ GeneralName: dNSName (2)
                    dNSName: test.epoche.es
        ▸ algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted: 9bc997359e3c3cf1f945ced7f741f844a03073762f03bdcc...
```

The connection is successfully performed, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.002862	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.005253	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.006777	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.007781	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.008718	10.10.20.40	10.10.20.52	DTLSv1.0	113	Certificate (Reassembled)
7	0.022914	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.030810	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.034178	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.034617	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.062983	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.069575	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.196252	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.219514	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.222982	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.226513	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

This fact can also be verified from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
77 bytes of (encrypted) application data received
0000  17 fe ff 00 01 00 00 00:00 00 02 00 40 ae 4d 3d  ............@.M=
0010  05 1d ec 0d 8b 74 cc dc:b5 d6 a4 57 cd 32 17 ad  .....t.....W.2..
0020  e3 52 f2 91 6e e3 1c ca:82 db 12 b3 ce 69 e1 57  .R..n........i.W
0030  1a 1e 76 2d e8 6e fb b9:92 db 01 69 2a f0 3a 8a  ..v-.n.....i*.:.
0040  0f 88 40 ec ed a3 62 50:2f 49 5a 27 22           ..@...bP/IZ'"

DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21          Hello, world!
```

The same results are obtained for DTLS 1.2 when presenting the same certificate:

```
▾ Certificates (1114 bytes)
    Certificate Length: 1111
  ▾ Certificate: 308204533082023ba003020102020101300d06092a864886... (id-at-commonName=invalid.epoche.es,id-at-organizatio
    ▾ signedCertificate
        version: v3 (2)
        serialNumber: 1
      ▸ signature (sha256WithRSAEncryption)
      ▸ issuer: rdnSequence (0)
      ▸ validity
      ▾ subject: rdnSequence (0)
        ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,id-at-organizationName
          ▸ RDNSequence item: 1 item (id-at-countryName=ee)
          ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          ▸ RDNSequence item: 1 item (id-at-localityName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
      ▸ subjectPublicKeyInfo
      ▾ extensions: 1 item
        ▾ Extension (id-ce-subjectAltName)
            Extension Id: 2.5.29.17 (id-ce-subjectAltName)
          ▾ GeneralNames: 1 item
            ▾ GeneralName: dNSName (2)
                dNSName: test.epoche.es
    ▸ algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: 9bc997359e3c3cf1f945ced7f741f844a03073762f03bdcc...
```

The network traffic capture also shows application data being successfully transmitted in this case.

No.	Time	Source	Destination	Protocol	Length	Info
3	4.727138	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
4	4.730852	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
5	4.736044	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
6	4.738017	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
7	4.740195	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
8	4.742381	10.10.20.40	10.10.20.52	DTLSv1.2	113	Certificate (Reassembled)
9	4.759534	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
10	4.766631	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
11	4.769798	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
12	4.770367	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
13	4.827548	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
14	4.834128	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
24	16.761209	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
25	16.784296	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
26	16.789796	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
27	16.794466	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

14.2.14.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

14.2.15 Test 5.1 - FCS_TLSC_EXT.1.2 for DTLS

14.2.15.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = foo.*.test.epoche.es

14.2.15.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.15.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name with value foo.*.test.epoche.es, as shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
6	31.761232	10.10.20.52	10.10.20.40	DTLSv1.0	157	Client Hello
7	31.765190	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
8	31.768051	10.10.20.52	10.10.20.40	DTLSv1.0	189	Client Hello
9	31.778053	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
10	31.780357	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
11	31.782952	10.10.20.40	10.10.20.52	DTLSv1.0	210	Certificate (Reassembled)
12	31.814778	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
13	31.819741	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
14	31.823035	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
15	31.823506	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message

```
    Length: 1214
    Message Sequence: 2
    Fragment Offset: 1071
    Fragment Length: 143
    Certificates Length: 1211
  ▾ Certificates (1211 bytes)
      Certificate Length: 1208
    ▾ Certificate: 308204b43082029ca0030201020201300d06092a864886... (id-at-commonName=foo.*.test.epoche.es,id-at
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=foo.*.test.epoche.es,id-at-organizationalUnitName=ee,id-at-orga
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=foo.*.test.epoche.es)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 4 items
          ▾ Extension (id-ce-basicConstraints)
```

© 2019 Microsoft Corporation. All rights reserved.

The client interrupts the handshake, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
6	31.761232	10.10.20.52	10.10.20.40	DTLSv1.0	157	Client Hello
7	31.765190	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
8	31.768051	10.10.20.52	10.10.20.40	DTLSv1.0	189	Client Hello
9	31.778053	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
10	31.780357	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
11	31.782952	10.10.20.40	10.10.20.52	DTLSv1.0	210	Certificate (Reassembled)
12	31.814778	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
13	31.819741	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
14	31.823035	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
15	31.823506	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
16	31.853115	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
17	31.857952	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=foo.*.test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

The same results are obtained for DTLS 1.2 when presenting the same certificate:

No.	Time	Source	Destination	Protocol	Length	Info
5	16.885124	10.10.20.52	10.10.20.40	DTLSv1.2	215	Client Hello
6	16.889478	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	16.891512	10.10.20.52	10.10.20.40	DTLSv1.2	247	Client Hello
8	16.894421	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	16.897808	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	16.906507	10.10.20.40	10.10.20.52	DTLSv1.2	210	Certificate (Reassembled)
11	16.944278	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	16.950667	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	16.953981	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	16.954508	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message

```
    Fragment Offset: 1071
    Fragment Length: 143
    Certificates Length: 1211
  ▾ Certificates (1211 bytes)
      Certificate Length: 1208
    ▾ Certificate: 308204b43082029ca0030201020201300d06092a864886... (id-at-commonName=foo.*.test.epoche.es,id-a
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=foo.*.test.epoche.es,id-at-organizationalUnitName=ee,id-at-org
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=foo.*.test.epoche.es)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 4 items
          ▾ Extension (id-ce-basicConstraints)
              Extension Id: 2.5.29.19 (id-ce-basicConstraints)
              BasicConstraintsSyntax [0 length]
          ▸ Extension (ns_cert_exts.comment)
```

The network traffic capture also shows no application data being transmitted in this case.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
5	16.885124	10.10.20.52	10.10.20.40	DTLSv1.2	215	Client Hello
6	16.889478	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
7	16.891512	10.10.20.52	10.10.20.40	DTLSv1.2	247	Client Hello
8	16.894421	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
9	16.897808	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
10	16.906507	10.10.20.40	10.10.20.52	DTLSv1.2	210	Certificate (Reassembled)
11	16.944278	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
12	16.950667	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
13	16.953981	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
14	16.954508	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
15	16.986901	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
16	17.000962	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.15.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.1** activity.

14.2.16 Test 5.2 - FCS_TLSC_EXT.1.2 for DTLS

14.2.16.1 Setup

The setup is identical to the one defined for Test 2.

14.2.16.2 Procedure

The procedure is identical to the one defined for Test 2.

14.2.16.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name with value *.test.epoche.es, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	153	Client Hello
2	0.004272	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006593	10.10.20.52	10.10.20.40	DTLSv1.0	185	Client Hello
4	0.009212	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.012668	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.016341	10.10.20.40	10.10.20.52	DTLSv1.0	206	Certificate (Reassembled)
7	0.037368	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.041536	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.044321	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.044837	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.059219	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.063184	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.261648	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.287070	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.297897	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data

```
Certificates Length: 1207
Certificates (1207 bytes)
  Certificate Length: 1204
  Certificate: 308204b030820298a0030201020201300d06092a864886... (id-at-commonName=*.test.epoche.es,id
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=*.test.epoche.es,id-at-organizationalUnitName=ee,id-at-o
          RDNSequence item: 1 item (id-at-countryName=ee)
          RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          RDNSequence item: 1 item (id-at-localityName=ee)
          RDNSequence item: 1 item (id-at-organizationName=ee)
          RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          RDNSequence item: 1 item (id-at-commonName=*.test.epoche.es)
      subjectPublicKeyInfo
      extensions: 4 items
        Extension (id-ce-basicConstraints)
          Extension Id: 2.5.29.19 (id-ce-basicConstraints)
          BasicConstraintsSyntax [0 length]
```

When the client connects with a reference identifier with value *foo.test.epoche.es*, the connection is successful, as shown in the network capture for DTLS 1.0:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	153	Client Hello
2	0.004272	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006593	10.10.20.52	10.10.20.40	DTLSv1.0	185	Client Hello
4	0.009212	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.012668	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.016341	10.10.20.40	10.10.20.52	DTLSv1.0	206	Certificate (Reassembled)
7	0.037368	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.041536	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.044321	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.044837	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.059219	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.063184	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
23	11.261648	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
24	11.287070	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
25	11.297897	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
26	11.303497	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

This fact can also be verified from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
77 bytes of (encrypted) application data received
0000  17 fe ff 00 01 00 00 00:00 00 02 00 40 b4 8a 23  ............@..#
0010  97 41 56 6b 2c 92 97 af:c1 24 5d b3 7b 7b 5f f5  .AVk,....$].{{_.
0020  e7 14 fd ae a5 e5 9f be:3c 3b d2 a0 5c 9d ef d6  ........<;..\...
0030  96 de b6 21 0f ae c5 3a:1a 8e bc dc 79 b8 20 85  ...!...:....y. .
0040  92 c5 39 57 4c 93 18 d3:80 a4 df 25 91           ..9WL........

DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21          Hello, world!

14 bytes of (encrypted) application data received
0000  14 fe ff 00 00 00 00 00:00 00 06 00 01 01       ..............
```

When the client connects with a reference identifier with value *test.epoche.es*, the client interrupts the handshake, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
6	11.018096	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
7	11.022474	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
8	11.024460	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
9	11.027543	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
10	11.030485	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
11	11.035982	10.10.20.40	10.10.20.52	DTLSv1.0	206	Certificate (Reassembled)
12	11.064485	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
13	11.069362	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
14	11.072342	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
15	11.072830	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
16	11.086573	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
17	11.091854	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=*.test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

When the client connects with a reference identifier with value *foo.bar.test.epoche.es*, the client also stops the connection, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
12	13.662198	10.10.20.52	10.10.20.40	DTLSv1.0	157	Client Hello
13	13.663539	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
14	13.665216	10.10.20.52	10.10.20.40	DTLSv1.0	189	Client Hello
15	13.665638	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
16	13.666114	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
17	13.666637	10.10.20.40	10.10.20.52	DTLSv1.0	206	Certificate (Reassembled)
18	13.671730	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
19	13.679067	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
20	13.680859	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
21	13.686619	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
22	13.696633	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
23	13.701608	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

© 2019 Microsoft Corporation. All rights reserved.

This fact can also be verified from the output of the tool used to perform the connection, where the same error is displayed.

The same results are obtained for DTLS 1.2 when presenting the same certificate:

No.	Time	Source	Destination	Protocol	Length	Info
3	2.346370	10.10.20.52	10.10.20.40	DTLSv1.2	211	Client Hello
4	2.350854	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
5	2.355609	10.10.20.52	10.10.20.40	DTLSv1.2	243	Client Hello
6	2.358425	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
7	2.362227	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
8	2.375322	10.10.20.40	10.10.20.52	DTLSv1.2	206	Certificate (Reassembled)
9	2.393495	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
10	2.396045	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
11	2.397491	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec

```
Fragment Offset: 1071
Fragment Length: 139
Certificates Length: 1207
Certificates (1207 bytes)
  Certificate Length: 1204
  Certificate: 308204b030820298a003020102020101300d06092a864886... (id-at-commonName=*.test.epoche.es,id-at
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=*.test.epoche.es,id-at-organizationalUnitName=ee,id-at-orga
          RDNSequence item: 1 item (id-at-countryName=ee)
          RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          RDNSequence item: 1 item (id-at-localityName=ee)
          RDNSequence item: 1 item (id-at-organizationName=ee)
          RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          RDNSequence item: 1 item (id-at-commonName=*.test.epoche.es)
      subjectPublicKeyInfo
      extensions: 4 items
        Extension (id-ce-basicConstraints)
          Extension Id: 2.5.29.19 (id-ce-basicConstraints)
```

The network traffic capture also shows application data being successfully transmitted for the first case (reference identifier set to *foo.test.epoche.es)*:

No.	Time	Source	Destination	Protocol	Length	Info
3	2.346370	10.10.20.52	10.10.20.40	DTLSv1.2	211	Client Hello
4	2.350854	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
5	2.355609	10.10.20.52	10.10.20.40	DTLSv1.2	243	Client Hello
6	2.358425	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
7	2.362227	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
8	2.375322	10.10.20.40	10.10.20.52	DTLSv1.2	206	Certificate (Reassembled)
9	2.393495	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
10	2.396045	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
11	2.397491	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
12	2.397945	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
13	2.424040	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
14	2.432048	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
37	13.490456	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
38	13.518238	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
39	13.527102	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
40	13.534317	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

For the other two reference identifiers (*test.epoche.es* and *foo.bar.test.epoche.es*) the handshake is interrupted by the client, as shown in the following network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
4	23.743750	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
5	23.748138	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
6	23.750438	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
7	23.753498	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
8	23.756865	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
9	23.760031	10.10.20.40	10.10.20.52	DTLSv1.2	206	Certificate (Reassembled)
10	23.774682	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
11	23.776912	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
12	23.780250	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
13	23.780678	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
14	23.801266	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
15	23.805861	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.16.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.2** activity.

14.2.17 Test 5.3 - FCS_TLSC_EXT.1.2 for DTLS

14.2.17.1 Setup

The setup is identical to the one defined for Test 2.

14.2.17.2 Procedure

The procedure is identical to the one defined for Test 2.

14.2.17.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The server presents a certificate with a Common Name with value *.es, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	144	Client Hello
2	0.002025	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.004164	10.10.20.52	10.10.20.40	DTLSv1.0	176	Client Hello
4	0.005001	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.005651	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.006136	10.10.20.40	10.10.20.52	DTLSv1.0	194	Certificate (Reassembled)
7	0.010832	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.026273	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.029092	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.029628	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.100656	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.107510	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

```
    Length: 1198
    Message Sequence: 2
    Fragment Offset: 1071
    Fragment Length: 127
    Certificates Length: 1195
  ▾ Certificates (1195 bytes)
      Certificate Length: 1192
    ▾ Certificate: 308204a43082028ca0030201020201300d06092a864886... (id-at-commonName=*.es,id-at-organizat
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=*.es,id-at-organizationalUnitName=ee,id-at-organizationNa
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=*.es)
        ▸ subjectPublicKeyInfo
```

When the client connects with a reference identifier with value *foo.epoche.es*, the client stops the handshake, as shown in the network capture for DTLS 1.0:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	144	Client Hello
2	0.002025	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.004164	10.10.20.52	10.10.20.40	DTLSv1.0	176	Client Hello
4	0.005001	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.005651	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.006136	10.10.20.40	10.10.20.52	DTLSv1.0	194	Certificate (Reassembled)
7	0.010832	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.026273	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.029092	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.029628	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.100656	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.107510	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=*.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

When the client connects with a reference identifier with value *epoche.es*, the client also interrupts the handshake, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.003611	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.006254	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.008962	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.010657	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.023814	10.10.20.40	10.10.20.52	DTLSv1.0	194	Certificate (Reassembled)
7	0.043249	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.046990	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.070111	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.070113	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.098347	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.102951	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection, obtaining the same output as in the previous case.

The same results are obtained for DTLS 1.2 when presenting the same certificate:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	202	Client Hello
2	0.002049	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.005492	10.10.20.52	10.10.20.40	DTLSv1.2	234	Client Hello
4	0.006747	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.007951	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.009512	10.10.20.40	10.10.20.52	DTLSv1.2	194	Certificate (Reassembled)
7	0.020784	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.024317	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.026324	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.026782	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.045201	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.053385	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

```
    Length: 1198
    Message Sequence: 2
    Fragment Offset: 1071
    Fragment Length: 127
    Certificates Length: 1195
  ▾ Certificates (1195 bytes)
      Certificate Length: 1192
    ▾ Certificate: 308204a43082028ca0030201020201300d06092a864886... (id-at-commonName=*.es,id-at-organizationalUnitN
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 6 items (id-at-commonName=*.es,id-at-organizationalUnitName=ee,id-at-organizationName=ee,id-a
            ▸ RDNSequence item: 1 item (id-at-countryName=ee)
            ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
            ▸ RDNSequence item: 1 item (id-at-localityName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
            ▸ RDNSequence item: 1 item (id-at-commonName=*.es)
        ▸ subjectPublicKeyInfo
```

The network traffic capture also shows no application data being transmitted for both cases:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	202	Client Hello
2	0.002049	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.005492	10.10.20.52	10.10.20.40	DTLSv1.2	234	Client Hello
4	0.006747	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.007951	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.009512	10.10.20.40	10.10.20.52	DTLSv1.2	194	Certificate (Reassembled)
7	0.020784	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.024317	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.026324	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.026782	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.045201	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.053385	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.17.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.3** activity.

14.2.18 Test 1 - FCS_TLSC_EXT.1.3 for DTLS

14.2.18.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es

The listed certificates form a valid certification path. All certificates are available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Windows Server 2016 Standard)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.20.*

Server machine shall contain the PowerShell script *W10DtlsServerAutomator.ps1* as well as the *WebServer.exe* tool also mentioned before.

Client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *Web-Client.exe* tool mentioned in the introduction to this section.

Moreover, the client machine shall have the secure configuration enabled, according to the section *1.2 Configuration* of the ***Operational Guidance***. The *Wireshark* network analyzer shall be installed on the client machine.

14.2.18.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run the Powershell script *W10DtlsServerAutomator*.ps1 as follows:

```
.\W10DtlsServerAutomator.ps1 dtls1 #for DTLS 1.0
.\W10DtlsServerAutomator.ps1 dtls12 #for DTLS 1.2
```

- Select the option number 4 when the menu is prompt.

© 2019 Microsoft Corporation. All rights reserved.

- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <dtls_version> (e.g. dlts1 for DTLS 1.0, dtls12 for DTLS 1.2 and <ip_server_address> and select the option number 13 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection from the client's output, server's output and the Wireshark capture.
- Repeat the steps above for both DTLS 1.0 and DTLS 1.2.

14.2.18.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

During this test, the server only sends the server certificate (CN = test.epoche.es) along with the handshake, as it can be appreciated in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
11	16.415248	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
12	16.418953	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
13	16.419405	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
14	16.424825	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
15	16.431892	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
16	16.434626	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
17	16.464498	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
18	16.469128	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
19	16.472160	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
20	16.472550	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
21	16.513485	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
22	16.518703	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

```
▸ Frame 16: 561 bytes on wire (4488 bits), 561 bytes captured (4488 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 51196
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Certificate (Reassembled)
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 0
      Sequence Number: 3
      Length: 506
    ▾ Handshake Protocol: Certificate (Reassembled)
        Handshake Type: Certificate (11)
        Length: 1565
        Message Sequence: 2
        Fragment Offset: 1071
        Fragment Length: 494
        Certificates Length: 1562
      ▸ Certificates (1562 bytes)
    ▸ [2 Message fragments (1565 bytes): #15(1071), #16(494)]
```

For the first part of this test, the connection failure when the full certificate chain is missing has been verified from the tool's output, which prints details about the error found in the connection process, as shown in the following screenshot:

© 2019 Microsoft Corporation. All rights reserved.

```
Server subject: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Intermediate CA 2

Called CertVerifyCertificateChainPolicy
Error 0x800b0109 (CERT_E_UNTRUSTEDROOT) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

This can also be seen in the following screenshot from the *Wireshark* capture for DTLS 1.0:

No.	Time	Source	Destination	Protocol	Length	Info
11	16.415248	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
12	16.418953	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
13	16.419405	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
14	16.424825	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
15	16.431892	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
16	16.434626	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
17	16.464498	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
18	16.469128	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
19	16.472160	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
20	16.472550	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
21	16.513485	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
22	16.518703	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

For the second part of this test, the connection is successful, as it can be seen in the following screenshot from the tool's output:

```
77 bytes of (encrypted) application data received
0000  17 fe ff 00 01 00 00 00:00 00 02 00 40 cd cb e3  ............@...
0010  e6 1f d5 52 d4 3f 9d 5d:93 c3 dc 14 5c fc 76 f1  ...R.?.]....\.v.
0020  ef c1 11 6b 82 97 aa 38:6f 2c dd 64 28 77 65 47  ...k...8o,.d(weG
0030  47 c6 e0 0f 70 7c 54 34:11 89 6d 4e af 33 5a 0e  G...p|T4..mN.3Z.
0040  19 74 6c ab 41 76 43 25:96 3f f7 17 a0           .tl.AvC..?...

DecryptMessage returned: 0x00000000
Buffers[1].BufferType = SECBUFFER_DATA
Decrypted data: 13 bytes
0000  48 65 6c 6c 6f 2c 20 77:6f 72 6c 64 21           Hello, world!

14 bytes of (encrypted) application data received
0000  14 fe ff 00 00 00 00 00:00 00 06 00 01 01        ..............
```

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
15	18.911174	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
16	18.915681	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
17	18.920948	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
18	18.924279	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
19	18.927745	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
20	18.931151	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
21	18.963283	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
22	18.968919	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
23	18.972199	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
24	18.972607	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
25	19.000187	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
26	19.005967	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message
41	23.392101	10.10.20.52	10.10.20.40	DTLSv1.0	167	Application Data
42	23.399601	10.10.20.40	10.10.20.52	DTLSv1.0	135	Application Data
43	23.402407	10.10.20.40	10.10.20.52	DTLSv1.0	119	Application Data
44	23.405405	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec

```
▸ Frame 20: 561 bytes on wire (4488 bits), 561 bytes captured (4488 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 60939
▾ Datagram Transport Layer Security
  ▾ DTLSv1.0 Record Layer: Handshake Protocol: Certificate (Reassembled)
      Content Type: Handshake (22)
      Version: DTLS 1.0 (0xfeff)
      Epoch: 0
      Sequence Number: 3
      Length: 506
    ▾ Handshake Protocol: Certificate (Reassembled)
        Handshake Type: Certificate (11)
        Length: 1565
        Message Sequence: 2
        Fragment Offset: 1071
        Fragment Length: 494
        Certificates Length: 1562
      ▸ Certificates (1562 bytes)
    ▸ [2 Message fragments (1565 bytes): #19(1071), #20(494)]
```

This time, as it can be appreciated, the connection is successful and both client and server send application data to each other.

For the third part of this test, the Intermediate CA 1 certificate is removed from the chain, and the certificate connection failure is verified in the same way. The evidence is identical to the one presented for the first part.

The same results are obtained for DTLS 1.2. The same certificate is presented:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004534	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.009999	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.011601	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.012131	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.012624	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
7	0.017570	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.030296	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.033195	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.033656	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.092873	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.098665	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

```
▸ Frame 6: 561 bytes on wire (4488 bits), 561 bytes captured (4488 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ User Datagram Protocol, Src Port: 440, Dst Port: 58957
▾ Datagram Transport Layer Security
  ▾ DTLSv1.2 Record Layer: Handshake Protocol: Certificate (Reassembled)
      Content Type: Handshake (22)
      Version: DTLS 1.2 (0xfefd)
      Epoch: 0
      Sequence Number: 3
      Length: 506
    ▾ Handshake Protocol: Certificate (Reassembled)
        Handshake Type: Certificate (11)
        Length: 1565
        Message Sequence: 2
        Fragment Offset: 1071
        Fragment Length: 494
        Certificates Length: 1562
      ▸ Certificates (1562 bytes)
    ▸ [2 Message fragments (1565 bytes): #5(1071), #6(494)]
```

The network traffic capture also shows no application data being transmitted for the first and third test cases:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004534	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.009999	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.011601	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.012131	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.012624	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
7	0.017570	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.030296	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.033195	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.033656	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.092873	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.098665	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

Accordingly, for the second test case the connection is successful and the application data transmission can be observed:

No.	Time	Source	Destination	Protocol	Length	Info
3	10.929725	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
4	10.934233	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
5	10.936664	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
6	10.940014	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
7	10.943485	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
8	10.947139	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
9	10.982729	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
10	10.986465	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
11	10.989163	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
12	10.989655	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
13	11.005337	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
14	11.010122	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message
28	22.083029	10.10.20.52	10.10.20.40	DTLSv1.2	167	Application Data
29	22.089886	10.10.20.40	10.10.20.52	DTLSv1.2	135	Application Data
30	22.091414	10.10.20.40	10.10.20.52	DTLSv1.2	119	Application Data
31	22.093186	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec

© 2019 Microsoft Corporation. All rights reserved.

14.2.18.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

14.2.19 Test 2 - FCS_TLSC_EXT.1.3 for DTLS

14.2.19.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (revoked)

The listed certificates form a valid certification path. All certificates are available in pem format as well as pfx.

14.2.19.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.19.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

A scenario with a revoked server certificate is tested using a CRL imported in the system's certificate store. As it can be appreciated in the following network capture for DTLS 1.0, the client interrupts the handshake when it receives the revoked certificate.

No.	Time	Source	Destination	Protocol	Length	Info
19	18.977376	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
20	18.980520	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
21	18.983117	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
22	18.985602	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
23	18.988652	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
24	19.001548	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
25	19.020670	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
26	19.027743	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
27	19.030792	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
28	19.031282	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
29	19.049262	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
30	19.057773	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

The connection failure due to certificate revocation is verified from the output from client's output as well.

© 2019 Microsoft Corporation. All rights reserved.

```
Server subject: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Intermediate CA 2

Called CertVerifyCertificateChainPolicy
Error 0x80092010 (CRYPT_E_REVOKED) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

The same result is obtained for DTLS 1.2. The network traffic capture also shows no application data being transmitted:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.001182	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.002380	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.003398	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.004334	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.005133	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
7	0.014078	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.021569	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.025146	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.025150	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.066004	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.072970	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

Identical client output is obtained:

```
Server subject: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Intermediate CA 2

Called CertVerifyCertificateChainPolicy
Error 0x80092010 (CRYPT_E_REVOKED) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

14.2.19.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

14.2.20 Test 3 - FCS_TLSC_EXT.1.3 for DTLS

14.2.20.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (expired certificate)

© 2019 Microsoft Corporation. All rights reserved.

The listed certificates form a valid certification path. All certificates are available in pem format as well as pfx.

14.2.20.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.20.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

During the course of this test, the server only sends the expired server certificate (CN = test.epoche.es) along with the handshake. The fact that it is expired can be checked in the following picture for DTLS 1.0:

No.	Time	Source	Destination	Protocol	Length	Info
10	11.812023	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
11	11.816333	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
12	11.818231	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
13	11.829995	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
14	11.833136	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
15	11.836033	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
16	11.862064	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
17	11.868180	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
18	11.871218	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
19	11.871743	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
20	11.890640	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
21	11.895383	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

```
Fragment Length: 494
Certificates Length: 1562
Certificates (1562 bytes)
  Certificate Length: 1559
  Certificate: 30820613308203fba003020102020210023 00d06092a8648... (id-at-commonName=test.epoche.es,id-
    signedCertificate
      version: v3 (2)
      serialNumber: 4098
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
        notBefore: utcTime (0)
          utcTime: 12-01-01 01:01:01 (UTC)
        notAfter: utcTime (0)
          utcTime: 15-01-01 01:01:01 (UTC)
      subject: rdnSequence (0)
```

The connection failure due to the lack of a valid certification path has been verified from the tool's output, which prints out details about the error found in the connection process, as shown in the following screenshot:

```
Server subject: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=test.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Intermediate CA 2

Called CertVerifyCertificateChainPolicy
Error 0x800b0101 (CERT_E_EXPIRED) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

© 2019 Microsoft Corporation. All rights reserved.

The connection failure is verified from the output from the *Wireshark* capture as well, where it can be appreciated that there is no application data being transmitted.

No.	Time	Source	Destination	Protocol	Length	Info
10	11.812023	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
11	11.816333	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
12	11.818231	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
13	11.829995	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
14	11.833136	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
15	11.836033	10.10.20.40	10.10.20.52	DTLSv1.0	561	Certificate (Reassembled)
16	11.862064	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
17	11.868180	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
18	11.871218	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
19	11.871743	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
20	11.890640	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
21	11.895383	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

The same result is obtained for DTLS 1.2 when the same expired certificate is presented:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004354	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.009957	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.017254	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.017864	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.018398	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
7	0.023769	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.044591	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.047612	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.048157	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.111186	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.116763	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

```
    Fragment Length: 494
    Certificates Length: 1562
  ▾ Certificates (1562 bytes)
      Certificate Length: 1559
    ▾ Certificate: 30820613308203fba0030201020202100230 0d06092a8648... (id-at-commonName=test.epoche.es,id-
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 4098
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▾ validity
          ▾ notBefore: utcTime (0)
              utcTime: 12-01-01 01:01:01 (UTC)
          ▾ notAfter: utcTime (0)
              utcTime: 15-01-01 01:01:01 (UTC)
```

The network traffic capture also shows no application data being transmitted:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.004354	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.009957	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.017254	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.017864	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.018398	10.10.20.40	10.10.20.52	DTLSv1.2	561	Certificate (Reassembled)
7	0.023769	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.044591	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.047612	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.048157	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.111186	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.116763	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.20.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

14.2.21 Test 4 - FCS_TLSC_EXT.1.3 for DTLS

14.2.21.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

14.2.21.2 Procedure

The procedure is identical to the one defined for Test 1.

14.2.21.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The *WebServer.exe* tool presents a server certificate with a Common Name not matching the reference identifier test.epoche.es (and no SAN identifier), as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004060	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.008916	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.011520	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.014873	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.018234	10.10.20.40	10.10.20.52	DTLSv1.0	116	Certificate (Reassembled)
7	0.039649	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.043607	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.046417	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.046857	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.059870	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.064365	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

```
serialNumber: 1
▸ signature (sha256WithRSAEncryption)
▸ issuer: rdnSequence (0)
▸ validity
▾ subject: rdnSequence (0)
  ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,i
    ▸ RDNSequence item: 1 item (id-at-countryName=ee)
    ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
    ▸ RDNSequence item: 1 item (id-at-localityName=ee)
    ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
    ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
    ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
▸ subjectPublicKeyInfo
▾ extensions: 1 item
  ▾ Extension (id-ce-subjectAltName)
      Extension Id: 2.5.29.17 (id-ce-subjectAltName)
    ▾ GeneralNames: 1 item
      ▾ GeneralName: dNSName (2)
          dNSName: invalid.epoche.es
```

The client interrupts the handshake, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.0	149	Client Hello
2	0.004060	10.10.20.40	10.10.20.52	DTLSv1.0	102	Hello Verify Request
3	0.008916	10.10.20.52	10.10.20.40	DTLSv1.0	181	Client Hello
4	0.011520	10.10.20.40	10.10.20.52	DTLSv1.0	148	Server Hello
5	0.014873	10.10.20.40	10.10.20.52	DTLSv1.0	1138	Certificate (Fragment)
6	0.018234	10.10.20.40	10.10.20.52	DTLSv1.0	116	Certificate (Reassembled)
7	0.039649	10.10.20.40	10.10.20.52	DTLSv1.0	67	Server Hello Done
8	0.043607	10.10.20.52	10.10.20.40	DTLSv1.0	325	Client Key Exchange
9	0.046417	10.10.20.52	10.10.20.40	DTLSv1.0	60	Change Cipher Spec
10	0.046857	10.10.20.52	10.10.20.40	DTLSv1.0	119	Encrypted Handshake Message
11	0.059870	10.10.20.40	10.10.20.52	DTLSv1.0	56	Change Cipher Spec
12	0.064365	10.10.20.40	10.10.20.52	DTLSv1.0	119	Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection, where an error informing about the CN mismatch is obtained:

```
Server subject: C=ee, S=ee, L=ee, O=ee, OU=ee, CN=invalid.epoche.es
Server issuer: C=ES, S=Spain, L=ee, O=ee, OU=ee, CN=EE Test Root CA

Called CertVerifyCertificateChainPolicy
Error 0x800b010f (CERT_E_CN_NO_MATCH) returned by CertVerifyCertificateChainPolicy or CertGetCertificateChain!
**** Error authenticating HTTPS server credentials!
Terminating connection with server.
Done
```

The same result is obtained for DTLS 1.2 when the same certificate is presented:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.003766	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.005704	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.007822	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.010331	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.012737	10.10.20.40	10.10.20.52	DTLSv1.2	116	Certificate (Reassembled)
7	0.055646	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.060660	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.063991	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec

```
▾ Certificates (1117 bytes)
    Certificate Length: 1114
  ▾ Certificate: 308204563082023ea003020102020101300d06092a864886... (id-at-commonName=invalid.epoche.es,
    ▾ signedCertificate
        version: v3 (2)
        serialNumber: 1
      ▸ signature (sha256WithRSAEncryption)
      ▸ issuer: rdnSequence (0)
      ▸ validity
      ▾ subject: rdnSequence (0)
        ▾ rdnSequence: 6 items (id-at-commonName=invalid.epoche.es,id-at-organizationalUnitName=ee,id-at
          ▸ RDNSequence item: 1 item (id-at-countryName=ee)
          ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=ee)
          ▸ RDNSequence item: 1 item (id-at-localityName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationName=ee)
          ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=ee)
          ▸ RDNSequence item: 1 item (id-at-commonName=invalid.epoche.es)
      ▸ subjectPublicKeyInfo
      ▾ extensions: 1 item
        ▾ Extension (id-ce-subjectAltName)
            Extension Id: 2.5.29.17 (id-ce-subjectAltName)
          ▾ GeneralNames: 1 item
            ▾ GeneralName: dNSName (2)
                dNSName: invalid.epoche.es
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
```

The network traffic capture also shows no application data being transmitted:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.40	DTLSv1.2	207	Client Hello
2	0.003766	10.10.20.40	10.10.20.52	DTLSv1.2	102	Hello Verify Request
3	0.005704	10.10.20.52	10.10.20.40	DTLSv1.2	239	Client Hello
4	0.007822	10.10.20.40	10.10.20.52	DTLSv1.2	148	Server Hello
5	0.010331	10.10.20.40	10.10.20.52	DTLSv1.2	1138	Certificate (Fragment)
6	0.012737	10.10.20.40	10.10.20.52	DTLSv1.2	116	Certificate (Reassembled)
7	0.055646	10.10.20.40	10.10.20.52	DTLSv1.2	67	Server Hello Done
8	0.060660	10.10.20.52	10.10.20.40	DTLSv1.2	325	Client Key Exchange
9	0.063991	10.10.20.52	10.10.20.40	DTLSv1.2	60	Change Cipher Spec
10	0.064375	10.10.20.52	10.10.20.40	DTLSv1.2	119	Encrypted Handshake Message
11	0.097343	10.10.20.40	10.10.20.52	DTLSv1.2	56	Change Cipher Spec
12	0.102681	10.10.20.40	10.10.20.52	DTLSv1.2	119	Encrypted Handshake Message

14.2.21.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

14.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_DTLS_EXT.1.2.

© 2019 Microsoft Corporation. All rights reserved.

15 FCS_RBG.1.1

The assurance activity for the **FCS_RBG.1.1** requirement is stated as follows:

> The evaluator will perform the following tests:
>
> The evaluator will perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator will perform 15 trials for each configuration. The evaluator will also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.
>
> If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator will generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).
>
> If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator will generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.
>
> The following values should be set/generated as described:
>
> - **Entropy input:** The length of the entropy input value must equal the seed length.
> - **Nonce:** If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.
> - **Personalization string:** The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator will use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.
> - **Additional input:** The additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

© 2019 Microsoft Corporation. All rights reserved.

15.1 Documentation Review activity

15.1.1 Findings

The evaluator has reviewed the information provided in TSS, section **6.2.1 Cryptographic Algorithms and Operations**. This section includes a list of the cryptographic algorithms supported by Windows 10 October 2018 Update:

Cryptographic Operation	Standard	Requirement	Evaluation Method
Encryption/Decryption	FIPS 197 AES	FCS_COP.1(SYM)	NIST CAVP #C 211, #C 346, #C 347, #C 350
	NIST SP 800-38A CBC mode		#C 211, #C 350
	NIST SP 800-38C CCM mode		#C 211, #C 346
	NIST SP 800-38E XTS mode		#C 211
	NIST SP 800-38F KW mode		#C 347
	NIST SP 800-38D GCM mode		#C 211
Digital signature (key generation)	FIPS 186-4 RSA	FCS_CKM.1	NIST CAVP #C 211, #C 348
Digital signature (generation)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (verification)	FIPS 186-4 RSA	FCS_COP.1(SIGN)	NIST CAVP #C 211, #C 348, #C 349 #C 350
Digital signature (generation and verification)	FIPS 186-4 DSA	Added as a prerequisite of NIST CAVP KAS #C 211	NIST CAVP #C 211
Digital signature (key generation)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1(WLAN)	NIST CAVP #C 211, #C 348, #C 350
Digital signature (key generation, signature generation and verification)	FIPS 186-4 ECDSA	FCS_CKM.1 FCS_CKM.1 (WLAN)	NIST CAVP #C 211, #C 348
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	FCS_COP.1(HASH)	NIST CAVP #C 211
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	FCS_COP.1(HMAC)	NIST CAVP #C 211, #C 350
Random number generation	NIST SP 800-90 CRT_DRBG	FCS_RBG_EXT.1	NIST CAVP #C 211, #C 350
Key agreement	NIST SP 800-56A ECDH	FCS_CKM.2	NIST CAVP #C 211, #C 350

© 2019 Microsoft Corporation. All rights reserved.

Cryptographic Operation	Standard	Requirement	Evaluation Method
Key establishment	NIST SP 800-56B RSA	FCS_CKM.2 FCS_CKM.2 (WLAN)	NIST CVL #C 211, #C 348 Tested by the CC evaluation lab[23]
Key-Based key derivation	SP800-108		NIST CAVP #C 347, #C 350
IKEv1	SP800-135		NIST CVL # C 211
IKEv2	SP800-135		NIST CVL # C 211
TLS	SP800-135	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.2 (WLAN), FCS_TLSC_EXT.3, FCS_TLSC_EXT.4, FCS_DTLS_EXT.1	NIST CVL # C 211

The vendor has specified the related random number generation algorithm (CRT_DRGB) NIST CAVP certificate:

- *Certificates number #C 211[24]and #C 350[25].

Counter DRBG

Supports Reseed

Prediction Resistance: No

Capabilities:

Mode: AES-256

Entropy Input: 256

Nonce: 128

Personalization String Length: 0

Additional Input: 0

Returned Bits: 512

Derivation Function Enabled: Yes

Prerequisites:

C 211 (AES)

[23]The test results are described in the evaluation and Assurance Activity Reports

[24]Validation for #C 211:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30607

[25]Validation for #C 350:https://csrc.nist.gov/projects/cryptographic-/algorithm-/validation-/program/details?validation=30746

© 2019 Microsoft Corporation. All rights reserved.

Counter DRBG

Supports Reseed

Prediction Resistance: No

Capabilities:

Mode: AES-256

Entropy Input: 256

Nonce: 128

Personalization String Length: 0

Additional Input: 0

Returned Bits: 512

Derivation Function Enabled: No

Prerequisites:

C 350 (AES)

The ***Operational Guidance*** document, states that for the evaluated version the following security policy needs to be applied (*section 1.2.1.3.1*):

> - Local Policies\Security Options\System cryptography: Use FIPS 140 compliant cryptographic algorithms, including encryption, hashing and signing algorithm

After applying this policy, only FIPS certified algorithm can be used, including random number generation algorithms defined in the above table. The vendor has also included the following wording:

> Windows 10 automatically generate random bits according to SP-800-90A, no configuration is necessary.

15.1.2 Verdict

The evaluator considers that the ***Operational Guidance*** document, defines how the FIPS security policy should be applied. Once this policy is enabled, only the approved random number generation algorithm described above can be used. No further configurations are needed to generate random bit according to NIST Special Publication 800-90A.

In addition, the evaluator has provided information about the Random number generation algorithm (*CTR_DRBG*) supported by the OS. The RNG implementation is the same (*CTR_DRBG*) which is conforming to NIST Special Publication *800-90A*.

Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

15.2 Test Activity

The evaluator has reviewed the NIST certificates and considers that this test activity is covered by FIPS certification. Therefore the PASS verdict is assigned.

15.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_RBG_EXT.1.1 requirement.

© 2019 Microsoft Corporation. All rights reserved.

16 FCS_RBG.1.2

The assurance activity for the **FCS_RBG.1.2** requirement is stated as follows:

> Documentation shall be produced - and the evaluator will perform the activities - in accordance with Appendix E - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex.
>
> In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.

16.1 Documentation Review activity

16.1.1 Findings

The vendor has provided the following documentation regarding the entropy:

- Windows 10 version 1809 RNG Design (November 6, 2018).
- Entropy Validation on Windows 10 for GP OS Evaluation.
- ***Security Target***.

The information provided in these documents describes with the expected level of detail how the entropy source is generated based on different entropy pools such as Interrupt timings or TPM, the available API functions that can process this entropy for a random number generation (*CTR_DRBG*), etc. In short, the life-cycle of the entropy is specified.

In addition, the ***Security Target*** document, includes more information about how the entropy source is health-tested before use, as shown in the following statement provided in the section **6.2.1 Cryptographic Algorithms and Operations**:

> *Windows has different entropy sources (deterministic and nondeterministic) which produce entropy data that is used for random numbers generation. In particular, this entropy data together with other data (such as the nonce) seed the DRBG algorithm. The entropy pool is populated using the following values*:
>
> - *An initial entropy value from a seed file provided to the Windows OS Loader at boot time (512 bits of entropy).*
> - *A calculated value based on the high-resolution CPU cycle counter which fires after every 1024 interrupts (a continuous source providing 16384 bits of entropy).*
> - *Random values gathered periodically from the Trusted Platform Module (TPM), (320 bits of entropy on boot, text intentionally left blank).*
> - *Random values gathered periodically by calling the RDRAND CPU instruction, (256 bits of entropy text intentionally left blank).*

© 2019 Microsoft Corporation. All rights reserved.

> *The entropy data is obtained from the entropy sources in a raw format and is health-tested before using it as input for the DRBG. text intentionally left blank.*

To complete this assurance activity and satisfy the activities described in Appendix E, the evaluator shall have access to internal APIs to perform the entropy tests.

Due to this fact, this activities shall be performed at Microsoft facilities.

Moreover, the vendor has also provided the following document: *Entropy Validation on Windows 10 for GP OS Evaluation*. This document contains information about the statistical tests performed on the entropy sources taking into account all platforms under evaluation, the statistical tests results are shown in the following tables:

Table 1 - Entropy Source Data (number of bits) by Evaluated Platform

This picture intentionally left blank

Table 2 - Entropy Validation Results

© 2019 Microsoft Corporation. All rights reserved.

This picture intentionally left blank

16.1.2 Verdict

The evaluator considers that the *Windows 10 version 1809 RNG Design (November 6, 2018)* document describes in detail how the life-cycle of the entropy used by the OS works. In addition, the information provided in the ***Security Target*** document, at TSS section *6.2.1 Cryptographic Algorithms and Operations*, about how the entropy source is health-tested before using it comparing the generated entropy blocks between them, is clear. The evaluator has also checked the statistical tests results provided in the *entropy validation* document, verifying that at least all the platforms, the operating systems and architectures under evaluation have been considered.

Due to this, the evaluator considers that the information provided in all these documents is enough for the fulfillment of the content requirement established in the *Annex E.Entropy Documentation and Assessment* of the Protection Profile.

Hence, the **PASS** verdict is assigned to the documentation review activity.

16.2 Test Activity

The assurance activity does not require any testing activities. Therefore, the **PASS** verdict is assigned.

© 2019 Microsoft Corporation. All rights reserved.

16.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_RBG_EXT.1.2 requirement.

© 2019 Microsoft Corporation. All rights reserved.

17 FCS_STO_EXT.1.1

The assurance activity for the **FCS_STO_EXT.1.1** requirement is stated as follows:

> The evaluator will check the TSS to ensure that it lists all persistent sensitive data for which the OS provides a storage capability. For each of these items, the evaluator will confirm that the TSS lists for what purpose it can be used, and how it is stored. The evaluator will confirm that cryptographic operations used to protect the data occur as specified in FCS_COP.1(1).
>
> The evaluator will also consult the developer documentation to verify that an interface exists for applications to securely store credentials.

17.1 Documentation Review activity

17.1.1 Findings

The evaluator has reviewed the section **6.2.4. Protecting Data with DPAPI** of the***Security Target*** document. This section states that Windows provides the *Data Protection API*, CNG DPAPI, which can be used to protect any persistent data which the developer deems to be sensitive.

DPAPI uses the *AES-CBC* algorithm to encrypt and decrypt the sensitive information. Once the sensitive information has been encrypted, it is stored in a directory which is part of the user's profile

DPAPI provides two functions to encrypt (*CryptProtectData* function) and decrypt (*CryptUnprotectData* function) data. This section of the TSS also includes a link to the Microsoft Developer Network (*MSDN*) where detailed information about the usage of these API functions is provided. The following images, which have been obtained from the *MSDN* website, describe the syntax of these API functions:

CryptProtectData:

© 2019 Microsoft Corporation. All rights reserved.

CryptProtectData function

12/05/2018 • 3 minutes to read

The **CryptProtectData** function performs encryption on the data in a DATA_BLOB structure. Typically, only a user with the same logon credential as the user who encrypted the data can decrypt the data. In addition, the encryption and decryption usually must be done on the same computer. For information about exceptions, see Remarks.

Syntax

C++ Copy

```
DPAPI_IMP BOOL CryptProtectData(
  DATA_BLOB                 *pDataIn,
  LPCWSTR                   szDataDescr,
  DATA_BLOB                 *pOptionalEntropy,
  PVOID                     pvReserved,
  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  DWORD                     dwFlags,
  DATA_BLOB                 *pDataOut
);
```

Parameters

`pDataIn`

A pointer to a DATA_BLOB structure that contains the plaintext to be encrypted.

`szDataDescr`

A string with a readable description of the data to be encrypted. This description string is included with the encrypted data. This parameter is optional and can be set to **NULL**.

`pOptionalEntropy`

A pointer to a DATA_BLOB structure that contains a password or other additional entropy used to encrypt the data. The **DATA_BLOB** structure used in the encryption phase must also be used in the decryption phase. This parameter can be set to **NULL** for no additional entropy. For information about protecting passwords, see Handling Passwords.

CryptUnprotectData:

© 2019 Microsoft Corporation. All rights reserved.

CryptUnprotectData function

12/05/2018 • 3 minutes to read

The **CryptUnprotectData** function decrypts and does an integrity check of the data in a DATA_BLOB structure. Usually, the only user who can decrypt the data is a user with the same logon credentials as the user who encrypted the data. In addition, the encryption and decryption must be done on the same computer. For information about exceptions, see the Remarks section of CryptProtectData.

Syntax

C++ Copy

```
DPAPI_IMP BOOL CryptUnprotectData(
  DATA_BLOB                 *pDataIn,
  LPWSTR                    *ppszDataDescr,
  DATA_BLOB                 *pOptionalEntropy,
  PVOID                     pvReserved,
  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  DWORD                     dwFlags,
  DATA_BLOB                 *pDataOut
);
```

Parameters

`pDataIn`

A pointer to a DATA_BLOB structure that holds the encrypted data. The **DATA_BLOB** structure's **cbData** member holds the length of the **pbData** member's byte string that contains the text to be encrypted.

`ppszDataDescr`

A pointer to a string-readable description of the encrypted data included with the encrypted data. This parameter can be set to **NULL**. When you have finished using *ppszDataDescr*, free it by calling the LocalFree function.

`pOptionalEntropy`

17.1.2 Verdict

The evaluator considers that the TSS provides enough information related to the method used for sensitive data protection. The vendor states that any persistent data which the developer deems to be sensitive can be protected using DPAPI. After protecting the data, the information is stored in a directory inside the user's profile.

The vendor has also identified the available interfaces for data protection (*CryptProtectData* and *CryptUnprotectData*), as well as the cryptographic algorithm used, AES256-CBC, which is one of the selected in the FCS_COP.1(SYM) requirement.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

17.2 Test Activity

The assurance activity does not require any testing activities. However the evaluator has carried out a test to verify the API functionality.

17.2.1 Test 1

17.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A C compiler or an IDE with a C compiler (e.g. *Microsoft Visual Studio*)

17.2.1.2 Procedure

The evaluator has compiled and executed the following C code, which uses the API functions described in the Document Review Activity section, to encrypt and decrypt some data.

```
#pragma comment(lib, "crypt32.lib")

#include <stdio.h>
#include <windows.h>
#include <Wincrypt.h>
#define MY_ENCODING_TYPE  (PKCS_7_ASN_ENCODING | X509_ASN_ENCODING)
void MyHandleError(char *s);

void main()
{

// Copyright (C) Microsoft.  All rights reserved.
// Encrypt data from DATA_BLOB DataIn to DATA_BLOB DataOut.
// Then decrypt to DATA_BLOB DataVerify.

//-------------------------------------------------------------------
// Declare and initialize variables.

DATA_BLOB DataIn;
DATA_BLOB DataOut;
DATA_BLOB DataVerify;
BYTE *pbDataInput =(BYTE *)"Hello world of data protection.";
DWORD cbDataInput = strlen((char *)pbDataInput)+1;
DataIn.pbData = pbDataInput;
DataIn.cbData = cbDataInput;
CRYPTPROTECT_PROMPTSTRUCT PromptStruct;
LPWSTR pDescrOut = NULL;

//-------------------------------------------------------------------
//  Begin processing.

printf("The data to be encrypted is: %s\n",pbDataInput);
```

© 2019 Microsoft Corporation. All rights reserved.

```
//--------------------------------------------------------------------
//  Initialize PromptStruct.

ZeroMemory(&PromptStruct, sizeof(PromptStruct));
PromptStruct.cbSize = sizeof(PromptStruct);
PromptStruct.dwPromptFlags = CRYPTPROTECT_PROMPT_ON_PROTECT;
PromptStruct.szPrompt = L"This is a user prompt.";

//--------------------------------------------------------------------
//  Begin protect phase.

if(CryptProtectData(
     &DataIn,
     L"This is the description string.", // A description string.
     NULL,                               // Optional entropy
                                         // not used.
     NULL,                               // Reserved.
     &PromptStruct,                      // Pass a PromptStruct.
     0,
     &DataOut))
{
     printf("The encryption phase worked. \n");
}
else
{
    MyHandleError("Encryption error!");
}
//--------------------------------------------------------------------
//   Begin unprotect phase.

if (CryptUnprotectData(
        &DataOut,
        &pDescrOut,
        NULL,                  // Optional entropy
        NULL,                  // Reserved
        &PromptStruct,         // Optional PromptStruct
        0,
        &DataVerify))
{
     printf("The decrypted data is: %s\n", DataVerify.pbData);
     printf("The description of the data was: %S\n",pDescrOut);
}
else
{
    MyHandleError("Decryption error!");
}
//--------------------------------------------------------------------
// At this point, memcmp could be used to compare DataIn.pbData and
// DataVerify.pbDate for equality. If the two functions worked
// correctly, the two byte strings are identical.

//--------------------------------------------------------------------
```

© 2019 Microsoft Corporation. All rights reserved.

```
//  Clean up.

LocalFree(pDescrOut);
LocalFree(DataOut.pbData);
LocalFree(DataVerify.pbData);

printf("Press enter to exit program...\n");
getchar();
} // End of main

//-----------------------------------------------------------------------
//  This example uses the function MyHandleError, a simple error
//  handling function, to print an error message to the
//  standard error (stderr) file and exit the program.
//  For most applications, replace this function with one
//  that does more extensive error reporting.

void MyHandleError(char *s)
{
    fprintf(stderr,"An error occurred in running the program. \n");
    fprintf(stderr,"%s\n",s);
    fprintf(stderr, "Error number %x.\n", GetLastError());
    fprintf(stderr, "Program terminating. \n");
    exit(1);
} // End of MyHandleError
```

17.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

© 2019 Microsoft Corporation. All rights reserved.

The code shown above, has been compiled and executed. First, the program encrypts some provided data using the function *CryptProtectData*. After that, the encrypted data is decrypted using the function *CryptUnprotectData*, showing in the console the unprotected value.

The following pictures show the output generated during the program execution:

© 2019 Microsoft Corporation. All rights reserved.

```
The data to be encrypted is: Hello world of data protection.
The encryption phase worked.
The decrypted data is: Hello world of data protection.
The description of the data was: This is the description string.
Press enter to exit program...
```

17.2.1.4 Verdict

As it can be appreciated in the obtained results, the evaluator has been able to encrypt and decrypt data using the functions provided by DPAPI.

The evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

17.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_STO_EXT.1.

© 2019 Microsoft Corporation. All rights reserved.

18 FCS_TLSC_EXT.1.1

The assurance activity for the **FCS_TLSC_EXT.1.1** requirement is stated as follows:

> The evaluator will check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator will check the TSS to ensure that the cipher suites specified include those listed for this component. The evaluator will also check the operational guidance to ensure that it contains instructions on configuring the OS so that TLS conforms to the description in the TSS. The evaluator will also perform the following tests:
>
> **Test 1**: The evaluator will establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
>
> **Test 2**: The evaluator will attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.
>
> **Test 3**: The evaluator will send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator will verify that the OS disconnects after receiving the server's Certificate handshake message.
>
> **Test 4**: The evaluator will configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
>
> **Test 5**: The evaluator will perform the following modifications to the traffic:
>
> - **Test 5.1**: Change the TLS version selected by the server in the Server Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.
> - **Test 5.2**: Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE cipher suite) or that the server denies the client's Finished handshake message.
> - **Test 5.3**: Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator will verify that the client rejects the connection after receiving the Server Hello.

- **Test 5.4**: [conditional]: If an ECDHE or DHE ciphersuite is selected, modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.
- **Test 5.5**: Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.
- **Test 5.6**: Send a garbled message from the Server after the Server has issued the Change Cipher Spec message and verify that the client denies the connection.

18.1 Documentation Review activity

18.1.1 Findings

Section **6.2.3 Networking (TLS)** of the TSS, in the ***Security Target*** document, provides a website where the different cipher suites supported by the *Schannel* library are described.

The TSS identifies the following cipher suites used during the evaluated configuration:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

These cipher suites match with the selected ones in the requirement definition. The cipher suites listed above are a subset of the ones implemented in the *Schannel* library, as it can be checked in the previous URL.

Moreover, section **6.3 Managing Transport Layer Security (TLS)** of the ***Operational Guidance*** document describes how to configure these TLS cipher suites. The information can be found in the following MSDN links:

- MSDN Link 1
- MSDN Link 2

Selection of TLS cipher suites in the TLS handshake process is performed according to the order defined which can be configured as described in the links listed above.

© 2019 Microsoft Corporation. All rights reserved.

18.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

18.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
   a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
      i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
      ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Using *TlsClientTest* tool, the evaluator is able to identify the main cause when the SSL connection fails. Also, testing with web browser, while providing less failure output, allow the evaluator to validate the obtained findings in scenarios closer to the real ones a TOE user may encounter.

TlsCientTest tool was developed in C# using the System.Net.Security classes from .NET Framework to perform SSL connections, closely based on an example from Microsoft's MSDN. It effectively uses the authentication mechanism from the OS, which is the target of the evaluation.

Special care was taken to verify this claim, so several checks were performed for confirmation after the tool was developed. First, some code snippets are shown for reference.

Importing the security libraries:

```
using System;
using System.Collections;
using System.Net;
using System.Net.Security;
using System.Net.Sockets;
using System.Security.Authentication;
using System.Text;
using System.Security.Cryptography.X509Certificates;
using System.IO;
```

© 2019 Microsoft Corporation. All rights reserved.

Establishing the TLS 1.2 connection, importing client certificates in the process (if there are any):

```
SslStream sslStream = new SslStream(
    client.GetStream(),
   false,
   new RemoteCertificateValidationCallback(ValidateServerCertificate),
   null
);

// Load available certificates in store, including client certificates
X509CertificateCollection certCollection = new X509CertificateCollection();
X509Store store = new X509Store(StoreLocation.CurrentUser);
store.Open(OpenFlags.ReadOnly);
foreach (X509Certificate cert in store.Certificates)
{
    certCollection.Add(cert);
}
store.Close();
    // The server name must match the name on the server certificate.
try
{
    sslStream.AuthenticateAsClient(serverName, certCollection,
        SslProtocols.Tls12,
        checkCertificateRevocation);
}
catch (Exception e)
{
    Console.WriteLine("Exception: {0}\r\n", e.Message);
    if (e.InnerException != null)
    {
        Console.WriteLine("Inner exception: {0}", e.InnerException.Message);
    }
    Console.WriteLine("Authentication failed — closing the connection.\r\n");
    client.Close();
    return;
}
```

Rejecting the connection if certificate validation fails, printing failure output:

```
// The following method is invoked by the RemoteCertificateValidationDelegate.
public static bool ValidateServerCertificate(
    object sender,
    X509Certificate certificate,
    X509Chain chain,
    SslPolicyErrors sslPolicyErrors)
{
    if (sslPolicyErrors == SslPolicyErrors.None)
        return true;

    Console.WriteLine("Certificate error: {0} \n", sslPolicyErrors);
    foreach (X509ChainStatus status in chain.ChainStatus)
    {
        Console.WriteLine("Chain status: {0}", status.StatusInformation);
    }
```

© 2019 Microsoft Corporation. All rights reserved.

```
    // Do not allow this client to communicate with unauthenticated servers.
    return false;
}
```

Process Monitor from Windows Sysinternals is used to verify that the Operating System shared libraries related to TLS and cryptography are being used when performing a connection with this tool.

First, the tool's launch is observed, where the different DLLs needed are loaded. Among them we can find libraries such as *cryptbase.dll* and *bcryptprimitives.dll*.

Time of Day	Process Name	PID	Operation	Path
2:48:17.44996...	TlsClientTest.exe	6028	Process Start	
2:48:17.45000...	TlsClientTest.exe	6028	Thread Create	
2:48:17.45348...	TlsClientTest.exe	6028	Load Image	C:\Users\evaluador\Desktop\TlsClientTest\TlsClientTest.exe
2:48:17.45357...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\ntdll.dll
2:48:17.45369...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\ntdll.dll
2:48:17.47942...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\wow64.dll
2:48:17.47960...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\wow64win.dll
2:48:17.48040...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\kernel32.dll
2:48:17.48061...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\kernel32.dll
2:48:17.48075...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\kernel32.dll
2:48:17.48088...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\user32.dll
2:48:17.48160...	TlsClientTest.exe	6028	Load Image	C:\Windows\System32\wow64cpu.dll
2:48:17.48361...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\mscoree.dll
2:48:17.48384...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\kernel32.dll
2:48:17.48412...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\KernelBase.dll
2:48:17.48648...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\advapi32.dll
2:48:17.48674...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\msvcrt.dll
2:48:17.48689...	TlsClientTest.exe	6028	Thread Create	
2:48:17.48703...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\sechost.dll
2:48:17.48723...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\rpcrt4.dll
2:48:17.48751...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\sspicli.dll
2:48:17.48770...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\cryptbase.dll
2:48:17.48792...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\bcryptprimitives.dll
2:48:17.48805...	TlsClientTest.exe	6028	Thread Create	
2:48:17.48821...	TlsClientTest.exe	6028	Thread Create	
2:48:17.86031...	TlsClientTest.exe	6028	Load Image	C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscoreei.dll
2:48:17.86421...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\shlwapi.dll

During the TCP connection it can be observed how other key DLLs are leveraged. Some of them are *crypt32.dll*, *secur32.dll*, *schannel.dll* and *bcrypt.dll*.

© 2019 Microsoft Corporation. All rights reserved.

Time of Day	Process Name	PID	Operation	Path
2:48:17.95677...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\ws2_32.dll
2:48:17.96618...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\mswsock.dll
2:48:17.97250...	TlsClientTest.exe	6028	TCP Connect	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:17.97415...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\crypt32.dll
2:48:17.97448...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\msasn1.dll
2:48:17.97593...	TlsClientTest.exe	6028	Thread Create	
2:48:17.97956...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\profapi.dll
2:48:18.00388...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\secur32.dll
2:48:18.01237...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\psapi.dll
2:48:18.01426...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\schannel.dll
2:48:18.01703...	TlsClientTest.exe	6028	TCP Send	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.01758...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.01761...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.01764...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.01776...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.02349...	TlsClientTest.exe	6028	TCP Send	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.02357...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.02360...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.02562...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\mskeyprotect.dll
2:48:18.02716...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\bcrypt.dll
2:48:18.02861...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\ncrypt.dll
2:48:18.03037...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\ntasn1.dll
2:48:18.03331...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\ncryptsslp.dll
2:48:18.04410...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\shell32.dll
2:48:18.04465...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\cfgmgr32.dll
2:48:18.04585...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\windows.storage.dll
2:48:18.04621...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\powrprof.dll
2:48:18.04654...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\SHCore.dll
2:48:18.09449...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\cryptsp.dll
2:48:18.09626...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\rsaenh.dll
2:48:18.14650...	TlsClientTest.exe	6028	Load Image	C:\Windows\SysWOW64\gpapi.dll
2:48:18.14755...	TlsClientTest.exe	6028	Thread Create	
2:48:18.17827...	TlsClientTest.exe	6028	TCP Send	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.17835...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.17875...	TlsClientTest.exe	6028	TCP Receive	DESKTOP-AVNLF70:49911 -> test.com:https
2:48:18.18095...	TlsClientTest.exe	6028	TCP Disconnect	DESKTOP-AVNLF70:49911 -> test.com:https

Moreover, by enabling *Schannel event logging* as indicated by the vendor, the evaluator can observe in the *Windows Event Viewer* that the connections are performed via *Schannel*, as shown in the next picture:

© 2019 Microsoft Corporation. All rights reserved.

W10ClientAutomator.ps1 has been developed to automatically configure the TOE (i.e. certificates installation, validation or revocation) for each test case. After executing the script, a menu is prompted in order to select which requirement is going to be tested, as it can observed in the following code snippet:

```
function ShowMenu {
    Write-Host "======================"
    Write-Host "W10 CC Client Automator"
    Write-Host "======================"
    Write-Host "Please select group to test:"
    Write-Host "1. FCS_TLSC_EXT.1.1"
    Write-Host "2. FCS_TLSC_EXT.1.2"
    Write-Host "3. FCS_TLSC_EXT.1.3"
    Write-Host "4. FCS_TLSC_EXT.2.1"
    Write-Host "5. FCS_TLSC_EXT.3.1"
    Write-Host "6. FCS_TLSC_EXT.4.1"
    Write-Host "7. FIA_X509_EXT.1.1"
    Write-Host "8. FIA_X509_EXT.1.2"
    Write-Host "9. FIA_X509_EXT.2.1"
    Write-Host "10. FCS_DTLS_EXT.1.1"
    Write-Host "11. FCS_DTLS_EXT.1.2 - FCS_TLSC_EXT.1.1"
    Write-Host "12. FCS_DTLS_EXT.1.2 - FCS_TLSC_EXT.1.2"
    Write-Host "13. FCS_DTLS_EXT.1.2 - FCS_TLSC_EXT.1.3"
}
```

After choosing the requirement, the script automatically set the configuration on TOE using auxiliary functions to install, remove or revoke the certificates required to perform each test cases. The following screenshots show a couple of these auxiliary functions:

```
function AddRootCA {
    param(
        [string]$cert
```

© 2019 Microsoft Corporation. All rights reserved.

```
    )
    process {
        Import-Certificate -CertStoreLocation Cert:\CurrentUser\Root -FilePath
            $cert
    }
}

function AddClientCert {
    param(
        [string]$cert
    )
    process {
        $securestring = ConvertTo-SecureString -String "eeee" -AsPlainText -Force
        Import-PfxCertificate -FilePath $cert Cert:\CurrentUser\My -Password
            $securestring
    }
}

function ClearChain {
    # Delete all certs for first test
    Get-ChildItem -Path Cert:\CurrentUser\Root | Where-Object {$_.Subject -match
        "EE Test Root CA"}
        | Remove-Item
    Get-ChildItem -Path Cert:\CurrentUser\CA
        | Where-Object {$_.Subject -match "EE Test Intermediate CA 1"} |
            Remove-Item
    Get-ChildItem -Path Cert:\CurrentUser\CA
        | Where-Object {$_.Subject -match "EE Test Intermediate CA 2"} |
            Remove-Item
    # Delete CRL, could be there from other test
    certutil.exe -delstore CA "EE Test Intermediate CA 2"
}
```

Finally, each requirement has a function in which all the test defined in the Assurance Activity are executed:

```
function Fcs-Tlsc-Ext-1-1 {
    $title = "FCS_TLSC_EXT.1.1"
    $tests = @("1","2 (a)","2 (b)","3","4","5_1","5_2","5_3","5_4","5_5","5_6")
    $n_cipher_suites = 12
    AddRootCA ..\root_ca_cert.pem
    AddToHostsFile $ip_address $def_hostname
    foreach ($test in $tests) {
        if ($test -eq "1") {
            for ($i = 0; $i -lt $n_cipher_suites; $i++) {
                PerformTest "$($title) Test$($test)
                    ($($i+1)/$($n_cipher_suites))" $def_hostname
        }
        } else {
            PerformTest "$($title) Test$($test)" $def_hostname
        }
    }
}
```

© 2019 Microsoft Corporation. All rights reserved.

18.2.1 Test 1

18.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

18.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 1 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection from the client's output and the *Wireshark* capture.

© 2019 Microsoft Corporation. All rights reserved.

18.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* creates a TLS server that selects one of the supported cipher suites (the ones defined on the TSS) subsequently on every connection. The cipher suites supported by the client are specified on the Client Hello, as shown in the following picture:

The server responds with the selected cipher suite on the Server Hello, as shown next for the first connection:

A total of 10 connections are performed in this test, with different cipher suites selected by the server, so that the whole list of offered cipher suites is covered. The successful TLS connection is checked for every case.

The fact that the connection is successful can be verified from the network capture, as shown in the next picture for the first connection:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
99	0.000191	10.10.20.52	10.10.20.30	TCP	66	49673 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
100	0.000195	10.10.20.30	10.10.20.52	TCP	66	443 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
101	0.000197	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
102	0.000199	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
103	0.000202	10.10.20.30	10.10.20.52	TCP	60	443 → 49673 [ACK] Seq=1 Ack=149 Win=30336 Len=0
104	0.000204	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
105	0.000231	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0
106	0.000234	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
107	0.000237	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
108	0.000240	10.10.20.52	10.10.20.30	TLSv1.2	123	Application Data
109	0.000242	10.10.20.30	10.10.20.52	TLSv1.2	315	Application Data
110	0.000244	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
111	0.000246	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=560 Ack=3028 Win=262400 Len=0
112	0.000248	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [FIN, ACK] Seq=560 Ack=3028 Win=262400 Len=0
113	0.000249	10.10.20.30	10.10.20.52	TCP	60	443 → 49673 [ACK] Seq=3028 Ack=561 Win=31360 Len=0

It can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.1 Test1 (1/12)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

18.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

18.2.2 Test 2

18.2.2.1 Setup

The setup is identical to the one defined for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

18.2.2.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

In this case, web browsers are used to perform the connection, since the serverAuth purpose in the extendedKeyUsage field of certificates needs to be tested, and it refers to "TLS WWW server authentication" according to RFC 5280.

The first connection is performed with a server certificate that has the serverAuth purpose, as shown in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000015	10.10.20.52	10.10.20.30	TCP	66	49685 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
5	0.000018	10.10.20.30	10.10.20.52	TCP	66	443 → 49685 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
6	0.000020	10.10.20.52	10.10.20.30	TCP	60	49685 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
7	0.000021	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
8	0.000024	10.10.20.30	10.10.20.52	TCP	60	443 → 49685 [ACK] Seq=1 Ack=149 Win=30336 Len=0
9	0.000025	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
10	0.000031	10.10.20.52	10.10.20.30	TCP	60	49685 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
11	0.000033	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.000035	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
13	0.000075	10.10.20.52	10.10.20.30	TLSv1.2	123	Application Data
14	0.000078	10.10.20.30	10.10.20.52	TLSv1.2	315	Application Data
15	0.000080	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
16	0.000081	10.10.20.52	10.10.20.30	TCP	60	49685 → 443 [ACK] Seq=560 Ack=3028 Win=2101760 Len=0
17	0.000082	10.10.20.52	10.10.20.30	TCP	60	49685 → 443 [FIN, ACK] Seq=560 Ack=3028 Win=2101760 Len=0

```
    Certificate Length: 1106
  ▾ Certificate: 3082044e30820236a0030201020204248b9eac300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.e
    ▾ signedCertificate
        version: v3 (2)
        serialNumber: 613129900
      ▸ signature (sha256WithRSAEncryption)
      ▸ issuer: rdnSequence (0)
      ▸ validity
      ▸ subject: rdnSequence (0)
      ▸ subjectPublicKeyInfo
      ▾ extensions: 2 items
        ▸ Extension (id-ce-subjectAltName)
        ▾ Extension (id-ce-extKeyUsage)
            Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
          ▾ KeyPurposeIDs: 1 item
              KeyPurposeId: 1.3.6.1.5.5.7.3.1 (id-kp-serverAuth)
    ▸ algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
```

The fact that the connection is established is checked on the Application Data shown above.

It can also be observed on the browser window and on the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test2 (a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

© 2019 Microsoft Corporation. All rights reserved.

For the next connection, a certificate without serverAuth purpose (codeSigning purpose is sent instead) is presented, as follows:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49686 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49686 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49686 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000027	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000028	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000031	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [FIN, ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000034	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert

```
▾ Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 2561
    Certificates Length: 2558
  ▾ Certificates (2558 bytes)
      Certificate Length: 1106
    ▾ Certificate: 3082044e30820236a0030201020204013922623 00d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.e
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 20521570
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 2 items
          ▸ Extension (id-ce-subjectAltName)
          ▾ Extension (id-ce-extKeyUsage)
              Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
            ▾ KeyPurposeIDs: 1 item
                KeyPurposeId: 1.3.6.1.5.5.7.3.3 (id-kp-codeSigning)
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
```

The network capture reflects the connection failure:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49686 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49686 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49686 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000027	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000028	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000031	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [FIN, ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000034	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000074	10.10.20.30	10.10.20.52	TCP	60	443 → 49686 [FIN, ACK] Seq=2766 Ack=492 Win=31360 Len=0
13	0.000076	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [RST, ACK] Seq=492 Ack=2766 Win=0 Len=0
14	0.000077	10.10.20.52	10.10.20.30	TCP	60	49686 → 443 [RST] Seq=492 Win=0 Len=0

As it can be observed, the client stops the TLS handshake process right after it receives the server certificate. The user is informed on the several browser screens and the *TlsClientTool* as follows:

```
[+] Performing FCS_TLSC_EXT.1.1 Test2 (b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is not valid for the requested usage.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

© 2019 Microsoft Corporation. All rights reserved.

18.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

18.2.3 Test 3

18.2.3.1 Setup

The setup is identical to the one defined for Test 1.

18.2.3.2 Procedure

© 2019 Microsoft Corporation. All rights reserved.

The procedure is identical to the one defined for Test 1.

18.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the outgoing Server Hello and changes the selected cipher suite from TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 to TLS_RSA_WITH_AES_128_CBC_SHA, while keeping the ECDSA server certificate that is sent to the client. The corresponding code snippet where the modification is performed is shown next:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_1*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_SERVER_HELLO) //server
            hello
        {
            cipher_suite = ssl_get_cipher_suites(handshake);
            write_out(PRINT_INFO, "Changing cipher suite from 0x%04x to 0x%04x.",
                ntohs(*cipher_suite), hstate->new_cipher);
            \*cipher_suite = htons(hstate->new_cipher);
            hstate->success = 1;
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

```
 > Executing FCS_TLSC_EXT.1.1 Test 3
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49682
   I> Starting TLS handshake...
      I> Changing cipher suite from 0xc023 to 0x002f.
      E> Socket recv error: Connection reset by peer
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.1.1 Test 3
```

© 2019 Microsoft Corporation. All rights reserved.

The effect on the Server Hello can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60530 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000023	10.10.20.30	10.10.20.25	TCP	66	443 → 60530 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000025	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000027	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000030	10.10.20.30	10.10.20.25	TCP	60	443 → 60530 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000032	10.10.20.30	10.10.20.25	TLSv1.2	2648	Server Hello, Certificate, Server Key Exchange[Malformed Packet]
7	0.000039	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=147 Ack=2595 Win=262656 Len=0
8	0.000041	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [RST, ACK] Seq=147 Ack=2595 Win=0 Len=0

```
▸ Frame 6: 2648 bytes on wire (21184 bits), 2648 bytes captured (21184 bits)
▸ Ethernet II, Src: PcsCompu_f0:2a:11 (08:00:27:f0:2a:11), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.25
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60530, Seq: 1, Ack: 147, Len: 2594
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 61
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 57
        Version: TLS 1.2 (0x0303)
      ▸ Random: 6b918360afe5a1349942750f5ce44926e47434423874b249...
        Session ID Length: 0
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 17
      ▸ Extension: renegotiation_info (len=1)
      ▸ Extension: ec_point_formats (len=4)
      ▸ Extension: extended_master_secret (len=0)
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
```

As the following screenshot shows, the certificate sent by the server is still ECDSA:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60530 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000023	10.10.20.30	10.10.20.25	TCP	66	443 → 60530 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000025	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000027	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000030	10.10.20.30	10.10.20.25	TCP	60	443 → 60530 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000032	10.10.20.30	10.10.20.25	TLSv1.2	2648	Server Hello, Certificate, Server Key Exchange[Malformed Packet]
7	0.000039	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=147 Ack=2595 Win=262656 Len=0
8	0.000041	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [RST, ACK] Seq=147 Ack=2595 Win=0 Len=0

```
    Version: TLS 1.2 (0x0303)
    Length: 2362
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2358
      Certificates Length: 2355
    ▾ Certificates (2355 bytes)
        Certificate Length: 903
      ▾ Certificate: 30820383308201 6ba003020102020452 7003c6300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.
        ▾ signedCertificate
            version: v3 (2)
            serialNumber: 1383072710
          ▸ signature (sha256WithRSAEncryption)
          ▸ issuer: rdnSequence (0)
          ▸ validity
          ▸ subject: rdnSequence (0)
          ▾ subjectPublicKeyInfo
            ▾ algorithm (id-ecPublicKey)
                Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
              ▾ ECParameters: namedCurve (0)
                  namedCurve: 1.2.840.10045.3.1.7 (secp256r1)
              Padding: 0
              subjectPublicKey: 046584aca5bb5549a4e5ab2f80009dad4f731e91e6bf58e1...
          ▸ extensions: 2 items
        ▸ algorithmIdentifier (sha256WithRSAEncryption)
```

The OS disconnects after receiving this certificate, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60530 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000023	10.10.20.30	10.10.20.25	TCP	66	443 → 60530 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000025	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000027	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000030	10.10.20.30	10.10.20.25	TCP	60	443 → 60530 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000032	10.10.20.30	10.10.20.25	TLSv1.2	2648	Server Hello, Certificate, Server Key Exchange[Malformed Packet]
7	0.000039	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [ACK] Seq=147 Ack=2595 Win=262656 Len=0
8	0.000041	10.10.20.25	10.10.20.30	TCP	60	60530 → 443 [RST, ACK] Seq=147 Ack=2595 Win=0 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test3

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The message received was unexpected or badly formatted
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements es-

© 2019 Microsoft Corporation. All rights reserved.

tablished in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

18.2.4 Test 4

18.2.4.1 Setup

The setup is identical to the one defined for Test 1.

18.2.4.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the outgoing Server Hello and changes the selected cipher suite to TLS_NULL_WITH_NULL_NULL. The corresponding code snippet where the modification is performed is shown next:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_1*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_SERVER_HELLO) //server
            hello
        {
            cipher_suite = ssl_get_cipher_suites(handshake);
            write_out(PRINT_INFO, "Changing cipher suite from 0x%04x to 0x%04x.",
                ntohs(*cipher_suite), hstate->new_cipher);
            \*cipher_suite = htons(hstate->new_cipher);
            hstate->success = 1;
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 4
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49688
   I> Starting TLS handshake...
      I> Changing cipher suite from 0x002f to 0x0000.
      E> Socket recv error: Connection reset by peer
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.1.1 Test 4
```

The effect on the Server Hello can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49688 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000013	10.10.20.30	10.10.20.52	TCP	66	443 → 49688 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000015	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000017	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000020	10.10.20.30	10.10.20.52	TCP	60	443 → 49688 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000022	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000030	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0

```
▸ Frame 6: 2691 bytes on wire (21528 bits), 2691 bytes captured (21528 bits)
▸ Ethernet II, Src: PcsCompu_bb:22:97 (08:00:27:bb:22:97), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.52
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49688, Seq: 1, Ack: 149, Len: 2637
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 53
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 49
        Version: TLS 1.2 (0x0303)
      ▸ Random: aebebf4259d66105bc088bf4e1c7d48ea7a7aea96c0aef2f...
        Session ID Length: 0
        Cipher Suite: TLS_NULL_WITH_NULL_NULL (0x0000)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: renegotiation_info (len=1)
      ▸ Extension: extended_master_secret (len=0)
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

The OS denies the connection, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49688 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000013	10.10.20.30	10.10.20.52	TCP	66	443 → 49688 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000015	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000017	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000020	10.10.20.30	10.10.20.52	TCP	60	443 → 49688 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000022	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000030	10.10.20.52	10.10.20.30	TCP	60	49688 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test4

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The client and server cannot communicate, because they do not possess a common algorithm
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.4.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

© 2019 Microsoft Corporation. All rights reserved.

18.2.5 Test 5.1

18.2.5.1 Setup

The setup is identical to the one defined for Test 1.

18.2.5.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.5.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the outgoing Server Hello and changes the TLS version to a non-supported one like 1.3 (bytes 03 04). The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_1*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_SERVER_HELLO) //server
            hello
       {
            write_out(PRINT_INFO, "Changing Server Hello version from from
                0x%04x to 0x%04x.",
                ntohs(handshake->version), FAKE_SSL_VERSION);
            handshake->version = htons(FAKE_SSL_VERSION);
            hstate->success = 1;
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 5.1
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49689
   I> Starting TLS handshake...
      I> Changing Server Hello version from from 0x0303 to 0x0304.
      E> Socket recv error: Connection reset by peer
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.1.1 Test 5.1
```

The effect on the Server Hello can be appreciated in the next picture:

```
4 0.000015   10.10.20.52   10.10.20.30   TLSv1.2   202 Client Hello
5 0.000019   10.10.20.30   10.10.20.52   TCP        60 443 → 49689 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6 0.000020   10.10.20.30   10.10.20.52   TLSv1.2  2691 Server Hello, Certificate, Server Hello Done
7 0.000027   10.10.20.52   10.10.20.30   TCP        60 49689 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0
8 0.000028   10.10.20.52   10.10.20.30   TCP        60 49689 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0

> Frame 6: 2691 bytes on wire (21528 bits), 2691 bytes captured (21528 bits)
> Ethernet II, Src: PcsCompu_bb:22:97 (08:00:27:bb:22:97), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
> Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.52
> Transmission Control Protocol, Src Port: 443 (443), Dst Port: 49689 (49689), Seq: 1, Ack: 149, Len: 2637
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 53
    v Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 49
        Version: Unknown (0x0304)
      > Random
        Session ID Length: 0
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Compression Method: null (0)
        Extensions Length: 9
```

The OS denies the connection, as observed on the network capture:

```
3 0.000014   10.10.20.52   10.10.20.30   TCP        60 49689 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4 0.000015   10.10.20.52   10.10.20.30   TLSv1.2   202 Client Hello
5 0.000019   10.10.20.30   10.10.20.52   TCP        60 443 → 49689 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6 0.000020   10.10.20.30   10.10.20.52   TLSv1.2  2691 Server Hello, Certificate, Server Hello Done
7 0.000027   10.10.20.52   10.10.20.30   TCP        60 49689 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0
8 0.000028   10.10.20.52   10.10.20.30   TCP        60 49689 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0
```

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The client and server cannot communicate, because they do not possess a common algorithm
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.5.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.1** activity.

© 2019 Microsoft Corporation. All rights reserved.

18.2.6 Test 5.2

18.2.6.1 Setup

The setup is identical to the one defined for Test 1.

18.2.6.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.6.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document. The test case in *ee-tls-tool* intercepts the outgoing Server Hello and modifies the last byte in the nonce. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_1*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_SERVER_HELLO) //server
            hello
        {
            write_out(PRINT_INFO, "Changing last byte in Server Hello nonce from
                from
                0x%02x to 0x%02x.", (unsigned int)handshake->random[27],
                (unsigned int)(handshake->random[27] + 1));
            ++handshake->random[27];
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 5.2
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.55 on port 49673
   I> Starting TLS handshake...
      I> Changing last byte in Server Hello nonce from from 0xc1 to 0xc2.
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.1.1 Test 5.2
```

The effect on the Server Hello can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000012	10.10.20.55	10.10.20.30	TCP	66	49673 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	0.000016	10.10.20.30	10.10.20.55	TCP	66	443 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	0.000018	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
5	0.000020	10.10.20.55	10.10.20.30	TLSv1.2	202	Client Hello
6	0.000022	10.10.20.30	10.10.20.55	TCP	60	443 → 49673 [ACK] Seq=1 Ack=149 Win=30336 Len=0
7	0.000023	10.10.20.30	10.10.20.55	TLSv1.2	2650	Server Hello, Certificate, Server Key Exchange, Server Hello Done
8	0.000030	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=149 Ack=2597 Win=2102272 Len=0
9	0.000031	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [FIN, ACK] Seq=149 Ack=2597 Win=2102272 Len=0
10	0.000032	10.10.20.30	10.10.20.55	TCP	60	443 → 49673 [FIN, ACK] Seq=2597 Ack=150 Win=30336 Len=0
11	0.000033	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=150 Ack=2598 Win=2102272 Len=0

```
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49673, Seq: 1, Ack: 149, Len: 2596
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 61
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 57
        Version: TLS 1.2 (0x0303)
      ▾ Random: fb3a8e631684a331a67cbb4a66c1fae1f49cd8809f58ee5f...
          GMT Unix Time: Jul 26, 2103 19:45:55.000000000 CET
          Random Bytes: 1684a331a67cbb4a66c1fae1f49cd8809f58ee5f2e5565e9...
        Session ID Length: 0
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
        Compression Method: null (0)
        Extensions Length: 17
      ▸ Extension: renegotiation_info (len=1)
      ▸ Extension: ec_point_formats (len=4)
      ▸ Extension: extended_master_secret (len=0)
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done

0040  03 fb 3a 8e 63 16 84 a3  31 a6 7c bb 4a 66 c1 fa   ··:·c··· 1·|·Jf··
0050  e1 f4 9c d8 80 9f 58 ee  5f 2e 55 65 e9 db 54 aa   ######X· _.Ue··T·
0060  c2 00 c0 23 00 00 11 ff  01 00 01 00 00 0b 00 04   ###····· ########
0070  03 00 01 02 00 17 00 00  16 03 03 09 3a 0b 00 09   ######## ····:···
0080  36 00 09 33 00 03 87 30  82 03 83 30 82 01 6b a0   6··3···0 ···0··k·
0090  03 02 01 02 02 04 73 40  c0 22 30 0d 06 09 2a 86   ######s@ ·"0···*·
```

The OS denies the connection after the Server Key Exchange handshake message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000012	10.10.20.55	10.10.20.30	TCP	66	49673 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	0.000016	10.10.20.30	10.10.20.55	TCP	66	443 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	0.000018	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
5	0.000020	10.10.20.55	10.10.20.30	TLSv1.2	202	Client Hello
6	0.000022	10.10.20.30	10.10.20.55	TCP	60	443 → 49673 [ACK] Seq=1 Ack=149 Win=30336 Len=0
7	0.000023	10.10.20.30	10.10.20.55	TLSv1.2	2650	Server Hello, Certificate, Server Key Exchange, Server Hello Done
8	0.000030	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=149 Ack=2597 Win=2102272 Len=0
9	0.000031	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [FIN, ACK] Seq=149 Ack=2597 Win=2102272 Len=0
10	0.000032	10.10.20.30	10.10.20.55	TCP	60	443 → 49673 [FIN, ACK] Seq=2597 Ack=150 Win=30336 Len=0
11	0.000033	10.10.20.55	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=150 Ack=2598 Win=2102272 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The Local Security Authority cannot be contacted
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.6.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.2** activity.

18.2.7 Test 5.3

18.2.7.1 Setup

The setup is identical to the one defined for Test 1.

18.2.7.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.7.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the incoming Client Hello so that the server selects a cipher suite that was not presented by the client. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        hstate->success = 0;
       handshake = (TLS_handshake_1*)(record + 1);
       if (handshake->handshake_type == TLS_HANDSHAKE_CLIENT_HELLO) //client
           hello
       {
            cipher_suites = ssl_get_cipher_suites(handshake);
            cipher_suite_count = ntohs(*cipher_suites) / 2;
                //this is the size, and each entry is 2 bytes long
           ++cipher_suites;

           cipher =
               htons(CIPHER_SUITES_PARAMS.at("TLS_DHE_RSA_WITH_AES_256_CCM").id);
           for (i = 0; i < cipher_suite_count; ++i)
            cipher_suites[i] = cipher;
           hstate->success = 1;
        }
    }
    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
```

© 2019 Microsoft Corporation. All rights reserved.

```
}
```

The effect on the Server Hello can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49691 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000011	10.10.20.30	10.10.20.52	TCP	66	443 → 49691 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49691 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	3479	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=149 Ack=3426 Win=2102272 Len=0
8	0.000069	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [RST, ACK] Seq=149 Ack=3426 Win=0 Len=0

```
▸ Frame 6: 3479 bytes on wire (27832 bits), 3479 bytes captured (27832 bits)
▸ Ethernet II, Src: PcsCompu_bb:22:97 (08:00:27:bb:22:97), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.52
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49691, Seq: 1, Ack: 149, Len: 3425
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 53
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 49
        Version: TLS 1.2 (0x0303)
      ▸ Random: c6f821759be77939f4afe524818fcf6f93e85b8e56c40ab7...
        Session ID Length: 0
        Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CCM (0xc09f)
        Compression Method: null (0)
        Extensions Length: 9
      ▸ Extension: renegotiation_info (len=1)
```

The selected cipher suite is not one of those sent in the Client Hello, shown next:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49691 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000011	10.10.20.30	10.10.20.52	TCP	66	443 → 49691 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49691 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	3479	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=149 Ack=3426 Win=2102272 Len=0
8	0.000069	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [RST, ACK] Seq=149 Ack=3426 Win=0 Len=0

```
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 143
  ▾ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 139
      Version: TLS 1.2 (0x0303)
    ▸ Random: 5bf29443b7f6c848d37003ae50404c170a73b3c47376adfc...
      Session ID Length: 0
      Cipher Suites Length: 26
    ▾ Cipher Suites (13 suites)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
        Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Compression Methods Length: 1
```

The OS denies the connection after receiving the Server Hello, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49691 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000011	10.10.20.30	10.10.20.52	TCP	66	443 → 49691 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49691 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	3479	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [ACK] Seq=149 Ack=3426 Win=2102272 Len=0
8	0.000069	10.10.20.52	10.10.20.30	TCP	60	49691 → 443 [RST, ACK] Seq=149 Ack=3426 Win=0 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_3

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The client and server cannot communicate, because they do not possess a common algorithm
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.7.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.3** activity.

© 2019 Microsoft Corporation. All rights reserved.

18.2.8 Test 5.4

18.2.8.1 Setup

The setup is identical to the one defined for Test 1.

18.2.8.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.8.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the outgoing Key Exchange handshake message and modifies its last byte, belonging to the signature block. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_1*)(record + 1);

        if (handshake->handshake_type == TLS_SERVER_KEY_EXCH) //server key
            exchange
        {
            //find last byte
            exch = ((unsigned char*)handshake) + ntohs(record->length) - 1;

            //tweak last byte
            write_out(PRINT_INFO, "Changing last byte in server key exchange
                from from
                0x%02x to 0x%02x.", (unsigned int)*exch, (unsigned int)(*exch +
                    1));
            \*exch += 1;
            hstate->success = 1;
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 5.4
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.25 on port 60531
   I> Starting TLS handshake...
      I> Changing last byte in server key exchange from from 0xf8 to 0xf9.
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.1.1 Test 5.4
```

The effect on the Server Key Exchange message can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000017	10.10.20.25	10.10.20.30	TCP	66	60531 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	0.000021	10.10.20.30	10.10.20.25	TCP	66	443 → 60531 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7	0.000023	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8	0.000024	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
9	0.000027	10.10.20.30	10.10.20.25	TCP	60	443 → 60531 [ACK] Seq=1 Ack=147 Win=30336 Len=0
10	0.000028	10.10.20.30	10.10.20.25	TLSv1.2	2649	Server Hello, Certificate, Server Key Exchange, Server Hello Done
11	0.000034	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=147 Ack=2596 Win=262656 Len=0
12	0.000036	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [FIN, ACK] Seq=147 Ack=2596 Win=262656 Len=0
13	0.000037	10.10.20.30	10.10.20.25	TCP	60	443 → 60531 [FIN, ACK] Seq=2596 Ack=148 Win=30336 Len=0
14	0.000076	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=148 Ack=2597 Win=262656 Len=0

```
▸ Frame 10: 2649 bytes on wire (21192 bits), 2649 bytes captured (21192 bits)
▸ Ethernet II, Src: PcsCompu_f0:2a:11 (08:00:27:f0:2a:11), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.25
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60531, Seq: 1, Ack: 147, Len: 2595
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 148
    ▾ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 144
      ▾ EC Diffie-Hellman Server Params
          Curve Type: named_curve (0x03)
          Named Curve: secp256r1 (0x0017)
          Pubkey Length: 65
          Pubkey: 04594d7107803ef01479dce27c1aef5675627adbe038d618...
        ▸ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
          Signature Length: 71
          Signature: 30450221008a614b7b2b15469ac8fef359dd992fb98f5758...
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

```
09e0  1d 76 93 1c bd 60 f5 88  c2 cd e9 08 1e a6 76 66   ·v···`·· ····· vf
09f0  21 06 aa 24 84 73 81 e6  24 aa e7 91 f3 7e 1f 56   !··$·s·· $····~·V
0a00  07 e7 9e dd 0c 04 03 00  47 30 45 02 21 00 8a 61   ····· G0E·!··a
0a10  4b 7b 2b 15 46 9a c8 fe  f3 59 dd 99 2f b9 8f 57   K{+·F··· ·Y··/··W
0a20  58 7b f2 cd d3 7d 4a c2  bc ce 3e ec e6 24 02 20   X{···}J· ··>··$·
0a30  08 4f 7e 78 41 c8 60 71  28 8d 79 1f e2 f5 35 09   ·O~xA·`q (·y···5·
0a40  a5 15 d7 13 1f 2d f1 9c  02 0b c7 ba 5f 27 e9 f9   ····· ·····_'··
0a50  16 03 03 00 04 0e 00 00  00                        ····· ·
```

The OS denies the connection after the Server Key Exchange handshake message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000017	10.10.20.25	10.10.20.30	TCP	66	60531 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	0.000021	10.10.20.30	10.10.20.25	TCP	66	443 → 60531 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7	0.000023	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8	0.000024	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
9	0.000027	10.10.20.30	10.10.20.25	TCP	60	443 → 60531 [ACK] Seq=1 Ack=147 Win=30336 Len=0
10	0.000028	10.10.20.30	10.10.20.25	TLSv1.2	2649	Server Hello, Certificate, Server Key Exchange, Server Hello Done
11	0.000034	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=147 Ack=2596 Win=262656 Len=0
12	0.000036	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [FIN, ACK] Seq=147 Ack=2596 Win=262656 Len=0
13	0.000037	10.10.20.30	10.10.20.25	TCP	60	443 → 60531 [FIN, ACK] Seq=2596 Ack=148 Win=30336 Len=0
14	0.000076	10.10.20.25	10.10.20.30	TCP	60	60531 → 443 [ACK] Seq=148 Ack=2597 Win=262656 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_4

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The Local Security Authority cannot be contacted
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.8.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.4** activity demonstrate the fulfillment of the requirements

© 2019 Microsoft Corporation. All rights reserved.

established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.4** activity.

18.2.9 Test 5.5

18.2.9.1 Setup

The setup is identical to the one defined for Test 1.

18.2.9.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.9.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the server's Finished handshake message and modifies one of its bytes. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;
    if (record->record_type == TLS_CHANGE_CIPHER_SPEC)
        hstate->has_changed_cipher_spec = 1;
    else if ((record->record_type == TLS_RECORD_HANDSHAKE) &&
        (hstate->has_changed_cipher_spec))
    {   //this is the TLS Finished message according to the spec
        finished_data = ((unsigned char*)(record + 1)) + 6; //skip past the
            handshake type,
            length, and version fields

        write_out(PRINT_INFO, "Changing 6th byte in finished message from 0x%02x
            to 0x%02x.",
            (unsigned int)*finished_data, (unsigned int)(*finished_data + 1));
        \*finished_data += 1;
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 5.5
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49693
   I> Starting TLS handshake...
      I> Changing 6th byte in finished message from 0x83 to 0x84.
      I> TLS handshake complete.
   I> Connection terminated after successful SSL handshake.
   I> Attempting to gracefully shutdown TLS socket.
 > PASS: FCS_TLSC_EXT.1.1 Test 5.5
```

The effect on the server's Finished handshake message (which is encrypted) can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49693 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49693 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49693 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000029	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000032	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [FIN, ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000035	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000074	10.10.20.30	10.10.20.52	TCP	60	443 → 49693 [FIN, ACK] Seq=2766 Ack=492 Win=31360 Len=0
13	0.000076	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [RST, ACK] Seq=492 Ack=2766 Win=0 Len=0
14	0.000077	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [RST] Seq=492 Win=0 Len=0

```
▸ Frame 9: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits)
▸ Ethernet II, Src: PcsCompu_bb:22:97 (08:00:27:bb:22:97), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.52
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49693, Seq: 2638, Ack: 491, Len: 75
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 64
      Handshake Protocol: Encrypted Handshake Message
```

```
0010  00 73 e0 2d 40 00 40 06  1d f2 0a 0a 14 1e 0a 0a   ·s·-@·@· ........
0020  14 34 01 bb c2 1d cb 16  5e 13 f3 9d 25 20 50 18   ·4------ ^···% P·
0030  00 f5 55 b9 00 00 14 03  03 00 01 01 16 03 03 00   ··U····· ........
0040  40 dd 1b 6d ce fc 88 84  e5 f9 76 84 ef 10 a1 3d   @··m····  ··v····=
0050  94 94 9d 99 98 ce 78 51  ed 5b 09 53 d4 03 2e 16   ------xQ ·[·S····
0060  d6 98 bb b7 b5 28 7e 09  49 07 fa c8 aa 86 e2 8c   -----(~· I-------
0070  29 69 db b6 19 46 4b e6  fd 13 b7 b1 1a 23 f3 59   )i···FK· -----#·Y
0080  4c                                                 L
```

The OS denies the connection with RST after when it receives the modified message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49693 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49693 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49693 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000029	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000032	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [FIN, ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000035	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000074	10.10.20.30	10.10.20.52	TCP	60	443 → 49693 [FIN, ACK] Seq=2766 Ack=492 Win=31360 Len=0
13	0.000076	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [RST, ACK] Seq=492 Ack=2766 Win=0 Len=0
14	0.000077	10.10.20.52	10.10.20.30	TCP	60	49693 → 443 [RST] Seq=492 Win=0 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_5

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The specified data could not be decrypted
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.9.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.5** activity demonstrate the fulfillment of the requirements

© 2019 Microsoft Corporation. All rights reserved.

established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.5** activity.

18.2.10 Test 5.6

18.2.10.1 Setup

The setup is identical to the one defined for Test 1.

18.2.10.2 Procedure

The procedure is identical to the one defined for Test 1.

18.2.10.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the server's message after Change Cipher Spec, overriding it with random bytes. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;
    if (hstate->has_changed_cipher_spec)
    {
        write_out(PRINT_INFO, "Overriding record with random data...");
        record_len = ntohs(record->length) + sizeof(TLS_record);

        //memset(db->data, 0, record_len);
        for (i = 0; i < record_len; ++i)
            db->data[i] = rand();
        hstate->has_changed_cipher_spec = 0;
    }
    else if (record->record_type == TLS_CHANGE_CIPHER_SPEC)
        hstate->has_changed_cipher_spec = 1;

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates that the modification was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.1.1 Test 5.6
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49694
   I> Starting TLS handshake...
      I> Overriding record with random data...
      I> TLS handshake complete.
   I> Connection terminated after successful SSL handshake.
   I> Attempting to gracefully shutdown TLS socket.
 > PASS: FCS_TLSC_EXT.1.1 Test 5.6
```

The effect on this message can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49694 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000017	10.10.20.30	10.10.20.52	TCP	66	443 → 49694 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000021	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000023	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000028	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000029	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000037	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000039	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000042	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Ignored Unknown Record
10	0.000044	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000045	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000109	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [FIN, ACK] Seq=2766 Ack=491 Win=31360 Len=0
13	0.000111	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=491 Ack=2767 Win=2102272 Len=0
14	0.000113	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [FIN, ACK] Seq=491 Ack=2767 Win=2102272 Len=0
15	0.000114	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [ACK] Seq=2767 Ack=492 Win=31360 Len=0

```
▸ Frame 9: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits)
▸ Ethernet II, Src: PcsCompu_bb:22:97 (08:00:27:bb:22:97), Dst: Tp-LinkT_0d:fd:56 (d4:6e:0e:0d:fd:56)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.52
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49694, Seq: 2638, Ack: 491, Len: 75
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ▾ Ignored Unknown Record
    ▾ [Expert Info (Warning/Protocol): Ignored Unknown Record]
        [Ignored Unknown Record]
        [Severity level: Warning]
        [Group: Protocol]
```

```
0000  d4 6e 0e 0d fd 56 08 00  27 bb 22 97 08 00 45 00   ·n···V·· '"···E·
0010  00 73 3b 81 40 00 40 06  c2 9e 0a 0a 14 1e 0a 0a   ·s;·@·@· ........
0020  14 34 01 bb c2 1e f1 8d  8b b8 b6 3d 92 6b 50 18   ·4------ ---=·kP·
0030  00 f5 f6 67 00 00 14 03  03 00 01 01 ff b6 c2 65   ---g---- ----------e
0040  46 ce 6e 90 42 61 29 ae  0c ca 0e 55 80 18 d4 2a   F·n·Ba)· ···U···*
0050  c0 aa 75 6a 38 c2 d7 76  3d 20 a2 1c d7 64 81 1d   ··uj8··v = ···d··
0060  33 f0 ae 75 51 d7 23 5d  a1 32 b2 22 4a 87 4c 0a   3··uQ·#] ·2·"J·L·
0070  31 c1 74 69 83 4c df c0  6c 81 dc 43 e6 5e 61 19   1·ti·L·· l··C·^a·
0080  4e                                                 N
```

The OS denies the connection after it receives the modified message, as observed on the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49694 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000017	10.10.20.30	10.10.20.52	TCP	66	443 → 49694 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000021	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000023	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000028	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000029	10.10.20.30	10.10.20.52	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000037	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=149 Ack=2638 Win=2102272 Len=0
8	0.000039	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000042	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Ignored Unknown Record
10	0.000044	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=491 Ack=2713 Win=2102272 Len=0
11	0.000045	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000109	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [FIN, ACK] Seq=2766 Ack=491 Win=31360 Len=0
13	0.000111	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=491 Ack=2767 Win=2102272 Len=0
14	0.000113	10.10.20.52	10.10.20.30	TCP	60	49694 → 443 [FIN, ACK] Seq=491 Ack=2767 Win=2102272 Len=0
15	0.000114	10.10.20.30	10.10.20.52	TCP	60	443 → 49694 [ACK] Seq=2767 Ack=492 Win=31360 Len=0

This fact can also be seen on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.1 Test5_6

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: Received an unexpected EOF or 0 bytes from the transport stream.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

18.2.10.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.6** activity demonstrate the fulfillment of the requirements

© 2019 Microsoft Corporation. All rights reserved.

established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.6** activity.

18.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

19 FCS_TLSC_EXT.1.2

The assurance activity for the **FCS_TLSC_EXT.1.2** requirement is stated as follows:

> The evaluator will ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator will ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the OS. The evaluator will verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.
>
> The evaluator will configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:
>
> **Test 1**: The evaluator will present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator will verify that the connection fails.
>
> **Test 2**: The evaluator will present a server certificate that contains a CN that matches the reference identifier,contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator will repeat this test for each supported SAN type.
>
> **Test 3** [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be ommited.
>
> **Test 4**: The evaluator will present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator will verify that the connection succeeds.
>
> **Test 5**: The evaluator will perform the following wildcard tests with each supported type of reference identifier:
>
> - **Test 5.1**: The evaluator will present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.
> - **Test 5.2**: The evaluator will present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator will configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator will configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify

© 2019 Microsoft Corporation. All rights reserved.

that the connection fails. The evaluator will configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails.

- **Test 5.3**: The evaluator will present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator will configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator will configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails.

Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator will configure the DNS name and the service identifier. The evaluator will present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator will repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.

Test 7: [conditional] If pinned certificates are supported the evaluator will present a certificate that does not match the pinned certificate and verify that the connection fails.

19.1 Documentation Review activity

19.1.1 Findings

Section **6.2.3 Networking (TLS)** of the TSS in the ***Security Target*** document includes the following statements:

> "The reference identifier in Windows 10 for TLS is the DNS name or IP address of the remote server, which is compared against the DNS name as presented identifier in the Subject Alternative Name (SAN) or the Subject Name of the certificate. There is no configuration of the reference identifier.
>
> A certificate that uses a wildcard in the leftmost portion of the resource identifier (i.e., *.contoso.com) can be accepted for authentication, otherwise the certificate will be deemed invalid. Windows does not provide a general-purpose capability to "pin" TLS certificates."

The same information is provided in section **6.3 Managing Transport Layer Security (TLS)** of the ***Operational Guidance*** document. Both documents state that there is no configuration of the reference identifier, and the reference identifier is taken as the DNS name or the IP address of the remote server. It is also stated that DNS names are supported as SAN type and as Common Name in the certificate´s Subject Name field.

Also, it is indicated that IP addresses and wildcards are supported. Finally, it is also specified that Windows does not provide the capability to pin TLS certificates.

© 2019 Microsoft Corporation. All rights reserved.

19.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

19.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
   a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
      i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
      ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

19.2.1 Test 1

19.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = invalid.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

© 2019 Microsoft Corporation. All rights reserved.

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

19.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 2 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

19.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name and SAN identifier (DNS Name) not matching the reference identifier test.epoche.es, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

```
23 0.000035   10.10.20.30   10.10.20.52   TCP      60 443 → 49703 [ACK] Seq=1 Ack=149 Win=30336 Len=0
24 0.000036   10.10.20.30   10.10.20.52   TLSv1.2  2697 Server Hello, Certificate, Server Hello Done
25 0.000073   10.10.20.52   10.10.20.30   TCP      60 49703 → 443 [ACK] Seq=149 Ack=2644 Win=262656 Len=0
26 0.000075   10.10.20.52   10.10.20.30   TLSv1.2  396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
27 0.000078   10.10.20.30   10.10.20.52   TLSv1.2  129 Change Cipher Spec, Encrypted Handshake Message

▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2571
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2567
      Certificates Length: 2564
    ▾ Certificates (2564 bytes)
        Certificate Length: 1112
      ▾ Certificate: 308204543082023ca00302010202046ca63990300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=
        ▾ signedCertificate
            version: v3 (2)
            serialNumber: 1822833040
          ▸ signature (sha256WithRSAEncryption)
          ▸ issuer: rdnSequence (0)
          ▸ validity
          ▸ subject: rdnSequence (0)
          ▸ subjectPublicKeyInfo
          ▾ extensions: 2 items
            ▾ Extension (id-ce-subjectAltName)
                Extension Id: 2.5.29.17 (id-ce-subjectAltName)
              ▾ GeneralNames: 1 item
                ▾ GeneralName: dNSName (2)
                    dNSName: invalid.epoche.es
            ▸ Extension (id-ce-extKeyUsage)
```

The client rejects the connection, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
19	0.000021	10.10.20.52	10.10.20.30	TCP	66	49703 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
20	0.000028	10.10.20.30	10.10.20.52	TCP	66	443 → 49703 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
21	0.000030	10.10.20.52	10.10.20.30	TCP	60	49703 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
22	0.000032	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
23	0.000035	10.10.20.30	10.10.20.52	TCP	60	443 → 49703 [ACK] Seq=1 Ack=149 Win=30336 Len=0
24	0.000036	10.10.20.30	10.10.20.52	TLSv1.2	2697	Server Hello, Certificate, Server Hello Done
25	0.000073	10.10.20.52	10.10.20.30	TCP	60	49703 → 443 [ACK] Seq=149 Ack=2644 Win=262656 Len=0
26	0.000075	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
27	0.000078	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

19.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

19.2.2 Test 2

19.2.2.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

19.2.2.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 2 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

19.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents server certificates with a Common Name that matches the reference identifier test.epoche.es and a SAN identifier (DNS Name) that does not match, as shown in the following picture:

The client rejects the connection, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
16	0.000028	10.10.20.52	10.10.20.30	TCP	66	49704 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
17	0.000030	10.10.20.30	10.10.20.52	TCP	66	443 → 49704 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
18	0.000032	10.10.20.52	10.10.20.30	TCP	60	49704 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
19	0.000033	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
20	0.000036	10.10.20.30	10.10.20.52	TCP	60	443 → 49704 [ACK] Seq=1 Ack=149 Win=30336 Len=0
21	0.000037	10.10.20.30	10.10.20.52	TLSv1.2	2694	Server Hello, Certificate, Server Hello Done
22	0.000085	10.10.20.52	10.10.20.30	TCP	60	49704 → 443 [ACK] Seq=149 Ack=2641 Win=2102272 Len=0
23	0.000087	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
24	0.000089	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
25	0.000091	10.10.20.52	10.10.20.30	TCP	60	49704 → 443 [FIN, ACK] Seq=491 Ack=2716 Win=2102272 Len=0
26	0.000092	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
27	0.000094	10.10.20.30	10.10.20.52	TCP	60	443 → 49704 [FIN, ACK] Seq=2769 Ack=492 Win=31360 Len=0
28	0.000095	10.10.20.52	10.10.20.30	TCP	60	49704 → 443 [RST, ACK] Seq=492 Ack=2769 Win=0 Len=0
29	0.000096	10.10.20.52	10.10.20.30	TCP	60	49704 → 443 [RST] Seq=492 Win=0 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

19.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

19.2.3 Test 3

19.2.3.1 Setup

The setup is identical to the one defined for Test 2.

19.2.3.2 Procedure

The procedure is identical to the one defined for Test 2.

19.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name that matches the reference identifier test.epoche.es and no SAN extension, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

```
  6 0.000021   10.10.20.30   10.10.20.52   TLSv1.2   2664 Server Hello, Certificate, Server Hello Done
  7 0.000027   10.10.20.52   10.10.20.30   TCP         60 49705 → 443 [ACK] Seq=149 Ack=2611 Win=2102272 Len=0
  8 0.000029   10.10.20.52   10.10.20.30   TLSv1.2    396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  9 0.000032   10.10.20.30   10.10.20.52   TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
 10 0.000034   10.10.20.52   10.10.20.30   TLSv1.2    123 Application Data
 11 0.000035   10.10.20.30   10.10.20.52   TLSv1.2    315 Application Data

    Handshake Type: Certificate (11)
    Length: 2534
    Certificates Length: 2531
  ▾ Certificates (2531 bytes)
      Certificate Length: 1079
    ▾ Certificate: 308204333082021ba003020102020474d26e68300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.epoch
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1959947880
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▾ subject: rdnSequence (0)
          ▾ rdnSequence: 4 items (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.epoche.es,id-at-countryName=ES)
            ▸ RDNSequence item: 1 item (id-at-countryName=ES)
            ▸ RDNSequence item: 1 item (id-at-commonName=test.epoche.es)
            ▸ RDNSequence item: 1 item (id-at-organizationName=EE)
            ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=EE)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▸ Extension (id-ce-extKeyUsage)
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
        encrypted: b459b87974797af09fd156611283ba3346e5e7730bb77a8f...
      Certificate Length: 1446
```

The connection is successfully performed, as shown in the network capture:

```
No.  Time        Source       Destination  Protocol  Length Info
  1 0.000000   10.10.20.52   10.10.20.30   TCP         66 49705 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.000012   10.10.20.30   10.10.20.52   TCP         66 443 → 49705 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
  3 0.000014   10.10.20.52   10.10.20.30   TCP         60 49705 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  4 0.000016   10.10.20.52   10.10.20.30   TLSv1.2    202 Client Hello
  5 0.000019   10.10.20.30   10.10.20.52   TCP         60 443 → 49705 [ACK] Seq=1 Ack=149 Win=30336 Len=0
  6 0.000021   10.10.20.30   10.10.20.52   TLSv1.2   2664 Server Hello, Certificate, Server Hello Done
  7 0.000027   10.10.20.52   10.10.20.30   TCP         60 49705 → 443 [ACK] Seq=149 Ack=2611 Win=2102272 Len=0
  8 0.000029   10.10.20.52   10.10.20.30   TLSv1.2    396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  9 0.000032   10.10.20.30   10.10.20.52   TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
 10 0.000034   10.10.20.52   10.10.20.30   TLSv1.2    123 Application Data
 11 0.000035   10.10.20.30   10.10.20.52   TLSv1.2    315 Application Data
 12 0.000078   10.10.20.30   10.10.20.52   TLSv1.2    107 Encrypted Alert
 13 0.000081   10.10.20.52   10.10.20.30   TCP         60 49705 → 443 [ACK] Seq=560 Ack=3001 Win=2101760 Len=0
 14 0.000082   10.10.20.52   10.10.20.30   TCP         60 49705 → 443 [FIN, ACK] Seq=560 Ack=3001 Win=2101760 Len=0
 15 0.000083   10.10.20.30   10.10.20.52   TCP         60 443 → 49705 [ACK] Seq=3001 Ack=561 Win=31360 Len=0
```

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test3

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

19.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

19.2.4 Test 4

19.2.4.1 Setup

The setup is identical to the one defined for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

19.2.4.2 Procedure

The procedure is identical to the one defined for Test 1.

19.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name that does not match the reference identifier test.epoche.es and a SAN identifier (DNS Name) that matches, as shown in the following picture where the connection is successfully performed:

This fact can also be verified from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.2 Test4

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

19.2.4.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

19.2.5 Test 5.1

19.2.5.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = foo.*.test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

19.2.5.2 Procedure

The procedure is identical to the one defined for Test 1.

19.2.5.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name with value foo.*.test.epoche.es, as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

```
 9 0.000027   10.10.20.30   10.10.20.52   TLSv1.2   2670 Server Hello, Certificate, Server Hello Done
10 0.000033   10.10.20.52   10.10.20.30   TCP         60 49707 → 443 [ACK] Seq=157 Ack=2617 Win=2102272 Len=0
11 0.000035   10.10.20.52   10.10.20.30   TLSv1.2    396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12 0.000037   10.10.20.30   10.10.20.52   TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
13 0.000077   10.10.20.52   10.10.20.30   TCP         60 49707 → 443 [FIN, ACK] Seq=499 Ack=2692 Win=2102272 Len=0
14 0.000079   10.10.20.30   10.10.20.52   TLSv1.2    107 Encrypted Alert

    Version: TLS 1.2 (0x0303)
    Length: 2544
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2540
      Certificates Length: 2537
    ▾ Certificates (2537 bytes)
        Certificate Length: 1085
      ▾ Certificate: 3082043930820221a00302010202042fa9cacb300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=foo.*.test.epoc
        ▾ signedCertificate
            version: v3 (2)
            serialNumber: 799656651
          ▸ signature (sha256WithRSAEncryption)
          ▸ issuer: rdnSequence (0)
          ▸ validity
          ▾ subject: rdnSequence (0)
            ▾ rdnSequence: 4 items (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=foo.*.test.epoche.es,id-at-countryName=ES)
              ▸ RDNSequence item: 1 item (id-at-countryName=ES)
              ▸ RDNSequence item: 1 item (id-at-commonName=foo.*.test.epoche.es)
              ▸ RDNSequence item: 1 item (id-at-organizationName=EE)
              ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=EE)
          ▸ subjectPublicKeyInfo
          ▸ extensions: 1 item
```

The client rejects the connection, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000017	10.10.20.52	10.10.20.30	TCP	66	49707 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
5	0.000019	10.10.20.30	10.10.20.52	TCP	66	443 → 49707 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
6	0.000021	10.10.20.52	10.10.20.30	TCP	60	49707 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
7	0.000023	10.10.20.52	10.10.20.30	TLSv1.2	210	Client Hello
8	0.000025	10.10.20.30	10.10.20.52	TCP	60	443 → 49707 [ACK] Seq=1 Ack=157 Win=30336 Len=0
9	0.000027	10.10.20.30	10.10.20.52	TLSv1.2	2670	Server Hello, Certificate, Server Hello Done
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49707 → 443 [ACK] Seq=157 Ack=2617 Win=2102272 Len=0
11	0.000035	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.000037	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
13	0.000077	10.10.20.52	10.10.20.30	TCP	60	49707 → 443 [FIN, ACK] Seq=499 Ack=2692 Win=2102272 Len=0
14	0.000079	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
15	0.000081	10.10.20.30	10.10.20.52	TCP	60	443 → 49707 [FIN, ACK] Seq=2745 Ack=500 Win=31360 Len=0
16	0.000082	10.10.20.52	10.10.20.30	TCP	60	49707 → 443 [RST, ACK] Seq=500 Ack=2745 Win=0 Len=0
17	0.000083	10.10.20.52	10.10.20.30	TCP	60	49707 → 443 [RST] Seq=500 Win=0 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 foo.bar.test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

19.2.5.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.1** activity.

© 2019 Microsoft Corporation. All rights reserved.

19.2.6 Test 5.2

19.2.6.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = *.test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

19.2.6.2 Procedure

The procedure is identical to the one defined for Test 1.

19.2.6.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name with value *.test.epoche.es, as shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000018	10.10.20.52	10.10.20.30	TLSv1.2	206	Client Hello
5	0.000022	10.10.20.30	10.10.20.52	TCP	60	443 → 49708 [ACK] Seq=1 Ack=153 Win=30336 Len=0
6	0.000023	10.10.20.30	10.10.20.52	TLSv1.2	2666	Server Hello, Certificate, Server Hello Done
7	0.000030	10.10.20.52	10.10.20.30	TCP	60	49708 → 443 [ACK] Seq=153 Ack=2613 Win=2102272 Len=0
8	0.000031	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000034	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000036	10.10.20.52	10.10.20.30	TLSv1.2	123	Application Data
11	0.000037	10.10.20.30	10.10.20.52	TLSv1.2	315	Application Data
12	0.000080	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert

```
▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2540
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2536
      Certificates Length: 2533
    ▾ Certificates (2533 bytes)
        Certificate Length: 1081
      ▾ Certificate: 308204353082021da00302010202046fdfba74300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=*.t
        ▾ signedCertificate
            version: v3 (2)
            serialNumber: 1876933236
          ▸ signature (sha256WithRSAEncryption)
          ▸ issuer: rdnSequence (0)
          ▸ validity
          ▾ subject: rdnSequence (0)
            ▾ rdnSequence: 4 items (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=*.test.epoche.es,id-at-countryName=ES)
              ▸ RDNSequence item: 1 item (id-at-countryName=ES)
              ▸ RDNSequence item: 1 item (id-at-commonName=*.test.epoche.es)
              ▸ RDNSequence item: 1 item (id-at-organizationName=EE)
              ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=EE)
          ▸ subjectPublicKeyInfo
          ▸ extensions: 1 item
        ▸ algorithmIdentifier (sha256WithRSAEncryption)
```

When the client connects with a reference identifier with value *foo.test.epoche.es*, the connection is successful, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49708 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000014	10.10.20.30	10.10.20.52	TCP	66	443 → 49708 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000016	10.10.20.52	10.10.20.30	TCP	60	49708 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000018	10.10.20.52	10.10.20.30	TLSv1.2	206	Client Hello
5	0.000022	10.10.20.30	10.10.20.52	TCP	60	443 → 49708 [ACK] Seq=1 Ack=153 Win=30336 Len=0
6	0.000023	10.10.20.30	10.10.20.52	TLSv1.2	2666	Server Hello, Certificate, Server Hello Done
7	0.000030	10.10.20.52	10.10.20.30	TCP	60	49708 → 443 [ACK] Seq=153 Ack=2613 Win=2102272 Len=0
8	0.000031	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000034	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000036	10.10.20.52	10.10.20.30	TLSv1.2	123	Application Data
11	0.000037	10.10.20.30	10.10.20.52	TLSv1.2	315	Application Data
12	0.000080	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
13	0.000083	10.10.20.52	10.10.20.30	TCP	60	49708 → 443 [ACK] Seq=564 Ack=3003 Win=2101760 Len=0
14	0.000084	10.10.20.52	10.10.20.30	TCP	60	49708 → 443 [FIN, ACK] Seq=564 Ack=3003 Win=2101760 Len=0
15	0.000085	10.10.20.30	10.10.20.52	TCP	60	443 → 49708 [ACK] Seq=3003 Ack=565 Win=31360 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_2 (a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 foo.test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

When the client connects with a reference identifier with value *test.epoche.es*, the client rejects the connection, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49709 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49709 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.52	10.10.20.30	TCP	60	49709 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000019	10.10.20.30	10.10.20.52	TCP	60	443 → 49709 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2666	Server Hello, Certificate, Server Hello Done
7	0.000027	10.10.20.52	10.10.20.30	TCP	60	49709 → 443 [ACK] Seq=149 Ack=2613 Win=2102272 Len=0
8	0.000029	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000031	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000033	10.10.20.52	10.10.20.30	TCP	60	49709 → 443 [FIN, ACK] Seq=491 Ack=2688 Win=2102272 Len=0
11	0.000034	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000036	10.10.20.30	10.10.20.52	TCP	60	443 → 49709 [FIN, ACK] Seq=2741 Ack=492 Win=31360 Len=0
13	0.000077	10.10.20.52	10.10.20.30	TCP	60	49709 → 443 [RST, ACK] Seq=492 Ack=2741 Win=0 Len=0
14	0.000079	10.10.20.52	10.10.20.30	TCP	60	49709 → 443 [RST] Seq=492 Win=0 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_2 (b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

When the client connects with a reference identifier with value *foo.bar.test.epoche.es*, the client rejects the connection, as shown in the network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49710 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.52	TCP	66	443 → 49710 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000015	10.10.20.52	10.10.20.30	TCP	60	49710 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000016	10.10.20.52	10.10.20.30	TLSv1.2	210	Client Hello
5	0.000020	10.10.20.30	10.10.20.52	TCP	60	443 → 49710 [ACK] Seq=1 Ack=157 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.52	TLSv1.2	2666	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.52	10.10.20.30	TCP	60	49710 → 443 [ACK] Seq=157 Ack=2613 Win=2102272 Len=0
8	0.000030	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000032	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000034	10.10.20.52	10.10.20.30	TCP	60	49710 → 443 [FIN, ACK] Seq=499 Ack=2688 Win=2102272 Len=0
11	0.000035	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000037	10.10.20.30	10.10.20.52	TCP	60	443 → 49710 [FIN, ACK] Seq=2741 Ack=500 Win=31360 Len=0
13	0.000077	10.10.20.52	10.10.20.30	TCP	60	49710 → 443 [RST, ACK] Seq=500 Ack=2741 Win=0 Len=0
14	0.000079	10.10.20.52	10.10.20.30	TCP	60	49710 → 443 [RST] Seq=500 Win=0 Len=0

© 2019 Microsoft Corporation. All rights reserved.

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_2 (c)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 foo.bar.test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

19.2.6.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.2** activity.

19.2.7 Test 5.3

19.2.7.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = *.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

19.2.7.2 Procedure

The procedure is identical to the one defined for Test 1.

19.2.7.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *ee-tls-tool* presents a server certificate with a Common Name with value *.es, as shown in the following picture:

When the client connects with a reference identifier with value *epoche.es*, the client rejects the connection, as shown in the network capture:

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_3 (a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

© 2019 Microsoft Corporation. All rights reserved.

Equivalent results are also obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

When the client connects with a reference identifier with value *test.epoche.es*, the client rejects the connection, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.52	10.10.20.30	TCP	66	49712 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000013	10.10.20.30	10.10.20.52	TCP	66	443 → 49712 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000016	10.10.20.52	10.10.20.30	TCP	60	49712 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000018	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000021	10.10.20.30	10.10.20.52	TCP	60	443 → 49712 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000023	10.10.20.30	10.10.20.52	TLSv1.2	2654	Server Hello, Certificate, Server Hello Done
7	0.000029	10.10.20.52	10.10.20.30	TCP	60	49712 → 443 [ACK] Seq=149 Ack=2601 Win=2102272 Len=0
8	0.000030	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000032	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000034	10.10.20.52	10.10.20.30	TCP	60	49712 → 443 [FIN, ACK] Seq=491 Ack=2676 Win=2102272 Len=0
11	0.000035	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
12	0.000037	10.10.20.30	10.10.20.52	TCP	60	443 → 49712 [FIN, ACK] Seq=2729 Ack=492 Win=31360 Len=0
13	0.000078	10.10.20.52	10.10.20.30	TCP	60	49712 → 443 [RST, ACK] Seq=492 Ack=2729 Win=0 Len=0
14	0.000080	10.10.20.52	10.10.20.30	TCP	60	49712 → 443 [RST] Seq=492 Win=0 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.2 Test5_3 (b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are also obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

Identical results are obtained when two left-most labels are used, tested by using *foo.epoche.es* as the reference identifier.

19.2.7.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5.3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5.3** activity.

19.2.8 Test 6

This test is a conditional test and as stated in the Document Review activity:

> "The reference identifier in Windows 10 for TLS is the DNS name or IP address of the remote server, which is compared against the DNS name as presented identifier in the Subject Alternative Name (SAN) or the Subject Name of the certificate. There is no configuration of the reference identifier.""

Therefore, this test not applicable in this evaluation since the condition is not met.

19.2.9 Test 7

This test is a conditional test and as stated in the Document Review activity:

© 2019 Microsoft Corporation. All rights reserved.

> "A certificate that uses a wildcard in the leftmost portion of the resource identifier (i.e., *.contoso.com) can be accepted for authentication, otherwise the certificate will be deemed invalid. Windows does not provide a general-purpose capability to "pin" TLS certificates."

Therefore, this test not applicable in this evaluation since the condition is not met.

19.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.1.2.

© 2019 Microsoft Corporation. All rights reserved.

20 FCS_TLSC_EXT.1.3

The assurance activity for the **FCS_TLSC_EXT.1.3** requirement is stated as follows:

> The evaluator will use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following additional test:
>
> **Test 1**: The evaluator will demonstrate that a peer using a certificate without a valid certification path results in an authenticate failure. Using the administrative guidance, the evaluator will then load the trusted CA certificate(s) needed to validate the peer's certificate, and demonstrate that the connection succeeds. The evaluator then shall delete one of the CA certificates, and show that the connection fails.
>
> **Test 2**: The evaluator will demonstrate that a peer using a certificate which has been revoked results in an authentication failure.
>
> **Test 3**: The evaluator will demonstrate that a peer using a certificate which has passed its expiration date results in an authentication failure.
>
> **Test 4**: the evaluator will demonstrate that a peer using a certificate which does not have a valid identifier shall result in an authentication failure.

20.1 Documentation Review activity

20.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

20.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

20.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
    a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE

© 2019 Microsoft Corporation. All rights reserved.

b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:

   i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).

   ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

20.2.1 Test 1

20.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es

The listed certificates form a valid certification path. All certificates are available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

© 2019 Microsoft Corporation. All rights reserved.

20.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 3 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

20.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

It is worth noting that, during the course of this test, the server only sends the server certificate (CN = test.epoche.es) along with the handshake, as it can be appreciated in the following picture:

For the first part of this test, the connection failure when the full certificate chain is missing has been verified from the tool's output, which prints details about the error found in the connection process, as shown in the following screenshot:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.3 Test1 (a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate chain could not be built to a trusted root authority.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

This can also be seen in the following screenshot from the *Wireshark* capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1538	0.000969	10.10.20.52	10.10.20.30	TCP	66	49673 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1539	0.000970	10.10.20.30	10.10.20.52	TCP	66	443 → 49673 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
1540	0.000970	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
1541	0.000971	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
1542	0.000971	10.10.20.30	10.10.20.52	TCP	60	443 → 49673 [ACK] Seq=1 Ack=149 Win=30336 Len=0
1543	0.000971	10.10.20.30	10.10.20.52	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
1544	0.000975	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=149 Ack=1642 Win=262656 Len=0
1545	0.000975	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1546	0.000975	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
1547	0.000976	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=491 Ack=1717 Win=262656 Len=0
1550	0.000976	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
1551	0.000977	10.10.20.30	10.10.20.52	TCP	60	443 → 49673 [FIN, ACK] Seq=1770 Ack=491 Win=31360 Len=0
1552	0.000977	10.10.20.52	10.10.20.30	TCP	60	49673 → 443 [ACK] Seq=491 Ack=1771 Win=262656 Len=0

The client machine sends a FIN, ACK once the validation fails, and responds with RST, ACK to the application data coming from the server.

For the second part of this test, the connection is successful, as it can be seen in the following screenshot from the tool's output:

```
[+] Performing FCS_TLSC_EXT.1.3 Test1 (b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge.

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

This time, as it can be appreciated in the following network capture, the connection is successful and both client and server send application data to each other.

No.	Time	Source	Destination	Protocol	Length	Info
58	0.000130	10.10.20.52	10.10.20.30	TCP	66	49674 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
59	0.000132	10.10.20.30	10.10.20.52	TCP	66	443 → 49674 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
60	0.000133	10.10.20.52	10.10.20.30	TCP	60	49674 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
61	0.000135	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
62	0.000137	10.10.20.30	10.10.20.52	TCP	60	443 → 49674 [ACK] Seq=1 Ack=149 Win=30336 Len=0
63	0.000138	10.10.20.30	10.10.20.52	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
64	0.000163	10.10.20.52	10.10.20.30	TCP	60	49674 → 443 [ACK] Seq=149 Ack=1642 Win=2102272 Len=0
65	0.000166	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
66	0.000168	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
67	0.000170	10.10.20.52	10.10.20.30	TLSv1.2	123	Application Data
68	0.000172	10.10.20.30	10.10.20.52	TLSv1.2	315	Application Data
69	0.000173	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
70	0.000175	10.10.20.52	10.10.20.30	TCP	60	49674 → 443 [ACK] Seq=560 Ack=2032 Win=2101760 Len=0
71	0.000176	10.10.20.52	10.10.20.30	TCP	60	49674 → 443 [FIN, ACK] Seq=560 Ack=2032 Win=2101760 Len=0
72	0.000177	10.10.20.30	10.10.20.52	TCP	60	443 → 49674 [ACK] Seq=2032 Ack=561 Win=31360 Len=0

For the third part of this test, the Intermediate CA 1 certificate is removed from the chain, and the certificate connection failure is verified in the same way, as it can be observed from the tool's output:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.3 Test1 (c)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate chain could not be built to a trusted root authority.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in the network traffic capture:

No.	Time	Source	Destination	Protocol	Length	Info
21	0.000032	10.10.20.52	10.10.20.30	TCP	66	49675 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
22	0.000035	10.10.20.30	10.10.20.52	TCP	66	443 → 49675 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
23	0.000036	10.10.20.52	10.10.20.30	TCP	60	49675 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
24	0.000038	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
25	0.000041	10.10.20.30	10.10.20.52	TCP	60	443 → 49675 [ACK] Seq=1 Ack=149 Win=30336 Len=0
26	0.000042	10.10.20.30	10.10.20.52	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
27	0.000048	10.10.20.52	10.10.20.30	TCP	60	49675 → 443 [ACK] Seq=149 Ack=1642 Win=2102272 Len=0
28	0.000049	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
29	0.000092	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
31	0.000096	10.10.20.52	10.10.20.30	TCP	60	49675 → 443 [ACK] Seq=491 Ack=1717 Win=2102272 Len=0
33	0.000098	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
34	0.000099	10.10.20.30	10.10.20.52	TCP	60	443 → 49675 [FIN, ACK] Seq=1770 Ack=491 Win=31360 Len=0
35	0.000101	10.10.20.52	10.10.20.30	TCP	60	49675 → 443 [ACK] Seq=491 Ack=1771 Win=2102272 Len=0

20.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

© 2019 Microsoft Corporation. All rights reserved.

20.2.2 Test 2

20.2.2.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (revoked)

The listed certificates form a valid certification path. All certificates are available in pem format as well as pfx.

20.2.2.2 Procedure

The procedure is identical to the one defined for Test 1.

20.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The correct behaviour of certificate revocation checks is verified. As it can be appreciated, as soon as the client gets the revoked certificate, it will send back a fatal alert (Certificate Revoked) and no further packets will be accepted from the server.

No.	Time	Source	Destination	Protocol	Length	Info
10	0.000022	10.10.20.52	10.10.20.30	TCP	66	49676 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
11	0.000025	10.10.20.30	10.10.20.52	TCP	66	443 → 49676 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
12	0.000026	10.10.20.52	10.10.20.30	TCP	60	49676 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
13	0.000028	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
14	0.000031	10.10.20.30	10.10.20.52	TCP	60	443 → 49676 [ACK] Seq=1 Ack=149 Win=30336 Len=0
15	0.000032	10.10.20.30	10.10.20.52	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
16	0.000037	10.10.20.52	10.10.20.30	TCP	60	49676 → 443 [ACK] Seq=149 Ack=1642 Win=2102272 Len=0
17	0.000039	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
18	0.000041	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
19	0.000043	10.10.20.52	10.10.20.30	TCP	60	49676 → 443 [ACK] Seq=491 Ack=1717 Win=2102272 Len=0
22	0.000046	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
23	0.000086	10.10.20.30	10.10.20.52	TCP	60	443 → 49676 [FIN, ACK] Seq=1770 Ack=491 Win=31360 Len=0
24	0.000088	10.10.20.52	10.10.20.30	TCP	60	49676 → 443 [ACK] Seq=491 Ack=1771 Win=2102272 Len=0

The connection failure is verified from the output from *TlsClientTest* as well.

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.1.3 Test2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is revoked.

Chain status: The revocation function was unable to check revocation for the certificate.

Chain status: The revocation function was unable to check revocation because the revocation server was offline.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

20.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

20.2.3 Test 3

20.2.3.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (expired certificate)

The listed certificates form a valid certification path. All certificates are available in pem format.

20.2.3.2 Procedure

The procedure is identical to the one defined for Test 1.

20.2.3.3 Results

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

During the course of this test, the server only sends the expired server certificate (CN = test.epoche.es) along with the handshake. The fact that it is expired can be checked in the following picture:

The connection failure due to the lack of a valid certification path has been verified from the tool's output, which prints details about the error found in the connection process, as shown in the following screenshot:

The connection failure is verified from the output from the *Wireshark* capture as well.

The client machine sends a FIN, ACK once the validation fails, and responds with RST, ACK to the application data coming from the server.

© 2019 Microsoft Corporation. All rights reserved.

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

20.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

© 2019 Microsoft Corporation. All rights reserved.

20.2.4 Test 4

20.2.4.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificates used for the connection are different than the ones used for Test 1:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

20.2.4.2 Procedure

The procedure is identical to the one defined for Test 1.

20.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate with a Common Name and SAN identifier (DNS Name) not matching the reference identifier test.epoche.es, as shown in the following picture:

The client rejects the connection, as shown in the network capture:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
10	0.000013	10.10.20.52	10.10.20.30	TCP	66	49678 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
11	0.000021	10.10.20.30	10.10.20.52	TCP	66	443 → 49678 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
12	0.000023	10.10.20.52	10.10.20.30	TCP	60	49678 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
13	0.000024	10.10.20.52	10.10.20.30	TLSv1.2	202	Client Hello
14	0.000028	10.10.20.30	10.10.20.52	TCP	60	443 → 49678 [ACK] Seq=1 Ack=149 Win=30336 Len=0
15	0.000029	10.10.20.30	10.10.20.52	TLSv1.2	2697	Server Hello, Certificate, Server Hello Done
16	0.000035	10.10.20.52	10.10.20.30	TCP	60	49678 → 443 [ACK] Seq=149 Ack=2644 Win=2102272 Len=0
17	0.000037	10.10.20.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
18	0.000080	10.10.20.30	10.10.20.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
19	0.000083	10.10.20.52	10.10.20.30	TCP	60	49678 → 443 [FIN, ACK] Seq=491 Ack=2719 Win=2102272 Len=0
20	0.000084	10.10.20.30	10.10.20.52	TLSv1.2	107	Encrypted Alert
21	0.000086	10.10.20.30	10.10.20.52	TCP	60	443 → 49678 [FIN, ACK] Seq=2772 Ack=492 Win=31360 Len=0
22	0.000087	10.10.20.52	10.10.20.30	TCP	60	49678 → 443 [RST, ACK] Seq=492 Ack=2772 Win=0 Len=0
23	0.000088	10.10.20.52	10.10.20.30	TCP	60	49678 → 443 [RST] Seq=492 Win=0 Len=0

This fact can also be verified from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.1.3 Test4

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateNameMismatch

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

20.2.4.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

20.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.1.3.

© 2019 Microsoft Corporation. All rights reserved.

21 FCS_TLSC_EXT.2.1

The assurance activity for the **FCS_TLSC_EXT.2.1** requirement is stated as follows:

> The evaluator will verify that TSS describes support for the Supported Groups Extension and whether the required behavior is performed by default or may be configured. If the TSS indicates that support for the Supported Groups Extension must be configured to meet the requirement, the evaluator will verify that AGD guidance includes configuration instructions for the Supported Groups Extension.
>
> The evaluator will also perform the following test:
>
> The evaluator will configure a server to perform ECDHE key exchange using each of the TOE's supported curves and shall verify that the TOE successfully connects to the server.

21.1 Documentation Review activity

21.1.1 Findings

Section **6.2.3 Networking (TLS)** of the ***Security Target*** document states that Windows automatically includes the Elliptic Curve Extension as part of the Client Hello message. The elliptic curves supported by default are *secp256r1* and *secp384r1*. The elliptic curve *secp521r1* is disabled by default, but it can be enabled by following the ***Operational Guidance***.

Concretely, section **6.3.5 Configuring with group policy** of the ***Operational Guidance*** indicates that the Local Policy Editor can be used to configure the curves in the Elliptic Curve Extension by editing the ECC Curve Order list, enabling and ordering the desired cipher suites.

21.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

21.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
    a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE

© 2019 Microsoft Corporation. All rights reserved.

b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:

   i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).

   ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

21.2.1 Test 1

21.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. All certificates are available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

21.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

© 2019 Microsoft Corporation. All rights reserved.

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 4 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

21.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate using each time one of the elliptic curves selected in the ***Security Target*** (*secp256r1*, *secp384r1* and *secp521r1*). The following screenshot shows the supported elliptic curves as indicated in the Client Hello message:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60532 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.25	TCP	66	443 → 60532 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000015	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000019	10.10.20.30	10.10.20.25	TCP	60	443 → 60532 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000020	10.10.20.30	10.10.20.25	TLSv1.2	2649	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.000027	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [ACK] Seq=147 Ack=2596 Win=262656 Len=0
8	0.000029	10.10.20.25	10.10.20.30	TLSv1.2	220	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000031	10.10.20.30	10.10.20.25	TLSv1.2	145	Change Cipher Spec, Encrypted Handshake Message
10	0.000032	10.10.20.25	10.10.20.30	TLSv1.2	139	Application Data
11	0.000034	10.10.20.30	10.10.20.25	TLSv1.2	331	Application Data
12	0.000075	10.10.20.30	10.10.20.25	TLSv1.2	123	Encrypted Alert
13	0.000078	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [ACK] Seq=398 Ack=3034 Win=262144 Len=0
14	0.000079	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [FIN, ACK] Seq=398 Ack=3034 Win=262144 Len=0
15	0.000080	10.10.20.30	10.10.20.25	TCP	60	443 → 60532 [ACK] Seq=3034 Ack=399 Win=31360 Len=0

```
▾ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 137
    Version: TLS 1.2 (0x0303)
  ▸ Random: 5bfd587977d359b11a04ed0154c3a6d23ab8eafd1484dd18...
    Session ID Length: 0
    Cipher Suites Length: 24
  ▸ Cipher Suites (12 suites)
    Compression Methods Length: 1
  ▸ Compression Methods (1 method)
    Extensions Length: 72
  ▸ Extension: server_name (len=19)
  ▾ Extension: supported_groups (len=8)
      Type: supported_groups (10)
      Length: 8
      Supported Groups List Length: 6
    ▾ Supported Groups (3 groups)
        Supported Group: secp256r1 (0x0017)
        Supported Group: secp384r1 (0x0018)
        Supported Group: secp521r1 (0x0019)
  ▸ Extension: ec_point_formats (len=2)
  ▸ Extension: signature_algorithms (len=14)
```

In addition, the server sends the following certificate which uses the tested elliptic curve:

When secp256r1 elliptic curve is tested:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FCS_TLSC_EXT.2.1 Test 1 (prime256v1)
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49685
   I> Starting TLS handshake...
      I> TLS handshake complete.
   I> Handshake OK, trying to send data...
   I> SSL handshake successful!
   I> Attempting to gracefully shutdown TLS socket.
 > PASS: FCS_TLSC_EXT.2.1 Test 1 (prime256v1)
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60532 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.25	TCP	66	443 → 60532 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000014	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000015	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000019	10.10.20.30	10.10.20.25	TCP	60	443 → 60532 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000020	10.10.20.30	10.10.20.25	TLSv1.2	2649	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.000027	10.10.20.25	10.10.20.30	TCP	60	60532 → 443 [ACK] Seq=147 Ack=2596 Win=262656 Len=0
8	0.000029	10.10.20.25	10.10.20.30	TLSv1.2	220	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000031	10.10.20.30	10.10.20.25	TLSv1.2	145	Change Cipher Spec, Encrypted Handshake Message

```
    Handshake Type: Certificate (11)
    Length: 2358
    Certificates Length: 2355
  ▾ Certificates (2355 bytes)
      Certificate Length: 903
    ▾ Certificate: 308203833082016ba0030201020204704 72b1f300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1883712287
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▾ subjectPublicKeyInfo
          ▾ algorithm (id-ecPublicKey)
              Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
            ▾ ECParameters: namedCurve (0)
                namedCurve: 1.2.840.10045.3.1.7 (secp256r1)
          Padding: 0
```

When secp384r1 elliptic curve is tested:

```
 > Executing FCS_TLSC_EXT.2.1 Test 1 (secp384r1)
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49686
   I> Starting TLS handshake...
      I> TLS handshake complete.
   I> Handshake OK, trying to send data...
   I> SSL handshake successful!
   I> Attempting to gracefully shutdown TLS socket.
 > PASS: FCS_TLSC_EXT.2.1 Test 1 (secp384r1)
```

© 2019 Microsoft Corporation. All rights reserved.

```
 7 0.000022   10.10.20.30   10.10.20.25   TCP      60 443 → 60533 [ACK] Seq=1 Ack=147 Win=30336 Len=0
 8 0.000023   10.10.20.30   10.10.20.25   TLSv1.2  2742 Server Hello, Certificate, Server Key Exchange, Server Hello Done
 9 0.000030   10.10.20.25   10.10.20.30   TCP      60 60533 → 443 [ACK] Seq=147 Ack=2689 Win=262656 Len=0
10 0.000031   10.10.20.25   10.10.20.30   TLSv1.2  252 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
11 0.000033   10.10.20.30   10.10.20.25   TLSv1.2  145 Change Cipher Spec, Encrypted Handshake Message
12 0.000035   10.10.20.25   10.10.20.30   TLSv1.2  139 Application Data
13 0.000074   10.10.20.30   10.10.20.25   TLSv1.2  331 Application Data
14 0.000077   10.10.20.30   10.10.20.25   TLSv1.2  123 Encrypted Alert

    Version: TLS 1.2 (0x0303)
    Length: 2391
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2387
      Certificates Length: 2384
    ▾ Certificates (2384 bytes)
        Certificate Length: 932
      ▾ Certificate: 308203a030820188a00302010202047b30754a300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonN
        ▾ signedCertificate
            version: v3 (2)
            serialNumber: 2066773322
          ▸ signature (sha256WithRSAEncryption)
          ▸ issuer: rdnSequence (0)
          ▸ validity
          ▸ subject: rdnSequence (0)
          ▾ subjectPublicKeyInfo
            ▾ algorithm (id-ecPublicKey)
                Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
              ▾ ECParameters: namedCurve (0)
                  namedCurve: 1.3.132.0.34 (secp384r1)
            Padding: 0
            subjectPublicKey: 04f06fecae8d7e4874dbd00cef2c94370508b3d541e2e34a...
```

When secp521r1 elliptic curve is tested:

```
 > Executing FCS_TLSC_EXT.2.1 Test 1 (secp521r1)
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49687
   I> Starting TLS handshake...
      I> TLS handshake complete.
   I> Handshake OK, trying to send data...
   I> SSL handshake successful!
   I> Attempting to gracefully shutdown TLS socket.
 > PASS: FCS_TLSC_EXT.2.1 Test 1 (secp521r1)
```

```
No.  Time        Source       Destination  Protocol  Length Info
 4 0.000015   10.10.20.25   10.10.20.30   TLSv1.2   200 Client Hello
 5 0.000019   10.10.20.30   10.10.20.25   TCP       60 443 → 60534 [ACK] Seq=1 Ack=147 Win=30336 Len=0
 6 0.000021   10.10.20.30   10.10.20.25   TLSv1.2   1403 Server Hello, Certificate, Server Key Exchange, Server Hello Done
 7 0.000026   10.10.20.25   10.10.20.30   TLSv1.2   288 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
 8 0.000028   10.10.20.30   10.10.20.25   TLSv1.2   145 Change Cipher Spec, Encrypted Handshake Message
 9 0.000030   10.10.20.25   10.10.20.30   TLSv1.2   139 Application Data
10 0.000031   10.10.20.30   10.10.20.25   TLSv1.2   331 Application Data
11 0.000033   10.10.20.30   10.10.20.25   TLSv1.2   123 Encrypted Alert

  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 980
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 976
        Certificates Length: 973
      ▾ Certificates (973 bytes)
          Certificate Length: 970
        ▾ Certificate: 308203c6308201aea00302010202045442279300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonNa
          ▾ signedCertificate
              version: v3 (2)
              serialNumber: 1430528633
            ▸ signature (sha256WithRSAEncryption)
            ▸ issuer: rdnSequence (0)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▾ subjectPublicKeyInfo
              ▾ algorithm (id-ecPublicKey)
                  Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
                ▾ ECParameters: namedCurve (0)
                    namedCurve: 1.3.132.0.35 (secp521r1)
              Padding: 0
              subjectPublicKey: 04015c5067908cb35309037b5439156d55a009dc45cf6e83...
            ▸ extensions: 2 items
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
```

© 2019 Microsoft Corporation. All rights reserved.

For all cases the connection is successfully established, as it can be observed in the Application Data sent in each screenshot.

And from the output obtained using a web browser like Edge:

And in Internet Explorer for Windows Server:

The connection is always performed via CLI for each curve test:

© 2019 Microsoft Corporation. All rights reserved.

```
When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

21.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

21.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.2.1.

© 2019 Microsoft Corporation. All rights reserved.

22 FCS_TLSC_EXT.3.1

The assurance activity for the **FCS_TLSC_EXT.3.1** requirement is stated as follows:

> The evaluator will verify that TSS describes the signature_algorithm extension and whether the required behavior is performed by default or may be configured. If the TSS indicates that the signature_algorithm extension must be configured to meet the requirement, the evaluator will verify that AGD guidance includes configuration of the signature_algorithm extension.
>
> The evaluator will also perform the following test:
>
> The evaluator will configure the server to send a certificate in the TLS connection that is not supported according to the Client's HashAlgorithm enumeration within the signature_algorithms extension (for example, send a certificate with a SHA-1 signature). The evaluator will verify that the OS disconnects after receiving the server's Certificate handshake message.

22.1 Documentation Review activity

22.1.1 Findings

Section **6.2.3 Networking (TLS)** of the ***Security Target*** document states that Windows uses the following signature algorithms by default in the "Client Hello" message:

- SHA256
- SHA384
- SHA512

It is also stated that this list of accepted algorithms can be modified by editing a registry key. Section **6.3.7 Managing signature algorithms with the Windows registry** of the ***Operational Guidance**** document provides the concrete key where the status and order of these algorithms can be modified, which is:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00010003

Therefore, this allows the user to set up the evaluated configuration by disabling SHA1 for TLS connections.

22.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

22.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
   a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
      i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
      ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

22.2.1 Test 1

22.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es
- CN = EE Client (client certificate with friendly name "Fake PKCS12 Certificate")

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

© 2019 Microsoft Corporation. All rights reserved.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

22.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 5 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

22.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a server certificate using SHA1, a signature hash algorithm that is not supported. In order to do so, the tool will modify the incoming Client Hello to include the unsupported algorithm on the first position, so that the server believes it is actually supported. The output that informs about this fact is shown next:

```
 > Executing FCS_TLSC_EXT.3.1 Test 1
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.52 on port 49688
   I> Starting TLS handshake...
      I> Overwriting first signature_algorithm in Client Hello with 0x0201 (SHA1 with RSA)
      E> Socket recv error: Connection reset by peer
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FCS_TLSC_EXT.3.1 Test 1
```

The following screenshot shows the supported hash algorithms in the actual Client Hello, configured according to the TSS:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.86	10.10.20.30	TCP	66	49713 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.86	TCP	66	443 → 49713 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000015	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000016	10.10.20.86	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000020	10.10.20.30	10.10.20.86	TCP	60	443 → 49713 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.86	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0

```
> Cipher Suites (13 suites)
  Compression Methods Length: 1
> Compression Methods (1 method)
  Extensions Length: 72
> Extension: server_name (len=19)
> Extension: supported_groups (len=8)
> Extension: ec_point_formats (len=2)
v Extension: signature_algorithms (len=14)
    Type: signature_algorithms (13)
    Length: 14
    Signature Hash Algorithms Length: 12
  v Signature Hash Algorithms (6 algorithms)
    v Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
        Signature Hash Algorithm Hash: SHA256 (4)
        Signature Hash Algorithm Signature: RSA (1)
    v Signature Algorithm: rsa_pkcs1_sha384 (0x0501)
        Signature Hash Algorithm Hash: SHA384 (5)
        Signature Hash Algorithm Signature: RSA (1)
    v Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
        Signature Hash Algorithm Hash: SHA256 (4)
        Signature Hash Algorithm Signature: ECDSA (3)
    v Signature Algorithm: ecdsa_secp384r1_sha384 (0x0503)
        Signature Hash Algorithm Hash: SHA384 (5)
        Signature Hash Algorithm Signature: ECDSA (3)
    v Signature Algorithm: rsa_pkcs1_sha512 (0x0601)
        Signature Hash Algorithm Hash: SHA512 (6)
        Signature Hash Algorithm Signature: RSA (1)
    v Signature Algorithm: ecdsa_secp521r1_sha512 (0x0603)
        Signature Hash Algorithm Hash: SHA512 (6)
        Signature Hash Algorithm Signature: ECDSA (3)
  Extension: sessionTicket TLS (len=0)
```

In contrast, the server sends the following certificate which uses the unsupported algorithm mentioned, causing the mismatch:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000020	10.10.20.30	10.10.20.86	TCP	60	443 → 49713 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.86	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0
8	0.000030	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0

```
  v Certificate: 3082044e30820236a0030201020201471635f34300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,ic
    v signedCertificate
        version: v3 (2)
        serialNumber: 1902337844
      v signature (sha1WithRSAEncryption)
          Algorithm Id: 1.2.840.113549.1.1.5 (sha1WithRSAEncryption)
      > issuer: rdnSequence (0)
```

The OS disconnects as soon as it receives the Certificate handshake message, as shown in the next network capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.86	10.10.20.30	TCP	66	49713 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000012	10.10.20.30	10.10.20.86	TCP	66	443 → 49713 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000015	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000016	10.10.20.86	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000020	10.10.20.30	10.10.20.86	TCP	60	443 → 49713 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000021	10.10.20.30	10.10.20.86	TLSv1.2	2691	Server Hello, Certificate, Server Hello Done
7	0.000028	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [ACK] Seq=149 Ack=2638 Win=262656 Len=0
8	0.000030	10.10.20.86	10.10.20.30	TCP	60	49713 → 443 [RST, ACK] Seq=149 Ack=2638 Win=0 Len=0

The fact that the connection failed can also be verified from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.3.1 Test1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: A call to SSPI failed, see inner exception.

Inner exception: The token supplied to the function is invalid
Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

22.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements es-

© 2019 Microsoft Corporation. All rights reserved.

tablished in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

22.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.3.1.

© 2019 Microsoft Corporation. All rights reserved.

23 FCS_TLSC_EXT.4.1

The assurance activity for the **FCS_TLSC_EXT.4.1** requirement is stated as follows:

> The evaluator will ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.
>
> The evaluator will verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.
>
> The evaluator will also perform the following test:
>
> **Test 1**: The evaluator shall establish a connection to a peer server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes the negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
>
> **Test 2**: The evaluator shall establish a connection to a peer server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty client's Certificate message (type 11) and Certificate Verify (type 15) message.

23.1 Documentation Review activity

23.1.1 Findings

Section **6.2.3 Networking (TLS)** of the ***Security Target*** document states that:

"The TOE implements TLS to enable a trusted network path that is used for client and server authentication, as well as HTTPS"

Also, section **6.2.1 Client certificates and Certificate Authorities** of the ***Operational Guidance*** document includes the indicated steps needed to import a certificate, valid for client certificates.

23.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

23.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

© 2019 Microsoft Corporation. All rights reserved.

1. Software acting as a TLS server:
   a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
      i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
      ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

23.2.1 Test 1

23.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es

The listed certificates form a valid certification path. The root certificate is available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

© 2019 Microsoft Corporation. All rights reserved.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

23.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 6 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

23.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* does not send the Certificate Request message.

The effect on the certificate is shown as follows:

```
No.  Time      Source        Destination   Protocol  Length Info
  4 0.000015   10.10.20.25   10.10.20.30   TLSv1.2    200 Client Hello
  5 0.000018   10.10.20.30   10.10.20.25   TCP         60 443 → 60535 [ACK] Seq=1 Ack=147 Win=30336 Len=0
  6 0.000020   10.10.20.30   10.10.20.25   TLSv1.2   2836 Server Hello, Certificate, Server Hello Done
  7 0.000026   10.10.20.25   10.10.20.30   TCP         60 60535 → 443 [ACK] Seq=147 Ack=2783 Win=262656 Len=0
  8 0.000028   10.10.20.25   10.10.20.30   TLSv1.2    396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  9 0.000030   10.10.20.30   10.10.20.25   TLSv1.2    129 Change Cipher Spec, Encrypted Handshake Message
 10 0.000032   10.10.20.25   10.10.20.30   TLSv1.2    123 Application Data
 11 0.000071   10.10.20.30   10.10.20.25   TLSv1.2    315 Application Data
▸ Frame 6: 2836 bytes on wire (22688 bits), 2836 bytes captured (22688 bits)
▸ Ethernet II, Src: PcsCompu_f0:2a:11 (08:00:27:f0:2a:11), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.25
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60535, Seq: 1, Ack: 147, Len: 2782
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 2710
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 2706
        Certificates Length: 2703
      ▾ Certificates (2703 bytes)
          Certificate Length: 824
        ▾ Certificate: 308203343082021ca00302010202041dd2cf37300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonN
          ▸ signedCertificate
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: 9f84ca7e7b64dc77c436ca010efcbe4281d454679be83825...
          Certificate Length: 794
        ▸ Certificate: 308203163082201fea0030201020204200f7c85300d06092a... (id-at-organizationalUnitName=EE 1,id-at-organizationName=EE 1,id-at-com
          Certificate Length: 1076
        ▸ Certificate: 3082043030820218a00302010202041ad88caa300d06092a... (id-at-organizationalUnitName=EE 0,id-at-organizationName=EE 0,id-at-com
```

This fact can also be appreciated from the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FCS_TLSC_EXT.4.1 Test1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge. The user connects to the server:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

23.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

23.2.2 Test 2

23.2.2.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = test.epoche.es
- CN = EE Client (client certificate with friendly name "EE Client Certificate")

The listed certificates form a valid certification path. The root certificate is available in pem format, the client certificate in pfx format, and the other one will be sent by the server.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

© 2019 Microsoft Corporation. All rights reserved.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

23.2.2.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 6 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

23.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* presents a Certificate Request message and the client sends its certificate that is verified with the Certificate Verify message.

The effect on the certificate is shown as follows:

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000003	10.10.20.25	10.10.20.30	TCP	66	60536 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.000005	10.10.20.30	10.10.20.25	TCP	66	443 → 60536 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.000005	10.10.20.25	10.10.20.30	TCP	60	60536 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
6	0.000006	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
7	0.000006	10.10.20.30	10.10.20.25	TCP	60	443 → 60536 [ACK] Seq=1 Ack=147 Win=30336 Len=0
8	0.000007	10.10.20.30	10.10.20.25	TLSv1.2	3068	Server Hello, Certificate, Certificate Request, Server Hello Done
9	0.000008	10.10.20.25	10.10.20.30	TCP	60	60536 → 443 [ACK] Seq=147 Ack=2921 Win=262656 Len=0
10	0.000009	10.10.20.25	10.10.20.30	TLSv1.2	1754	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
11	0.000021	10.10.20.30	10.10.20.25	TCP	60	443 → 60536 [ACK] Seq=3015 Ack=1847 Win=33792 Len=0
12	0.000021	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
13	0.000022	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
14	0.000022	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data

TlsClientTest provides the client certificate during the handshake by accessing the local key store for the current user without explicit user intervention. As it can be observed in the next network capture, the connection is accepted as soon as the client sends its client certificate.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000003	10.10.20.25	10.10.20.30	TCP	66	60536 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.000005	10.10.20.30	10.10.20.25	TCP	66	443 → 60536 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.000005	10.10.20.25	10.10.20.30	TCP	60	60536 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
6	0.000006	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
7	0.000006	10.10.20.30	10.10.20.25	TCP	60	443 → 60536 [ACK] Seq=1 Ack=147 Win=30336 Len=0
8	0.000007	10.10.20.30	10.10.20.25	TLSv1.2	3068	Server Hello, Certificate, Certificate Request, Server Hello Done
9	0.000008	10.10.20.25	10.10.20.30	TCP	60	60536 → 443 [ACK] Seq=147 Ack=2921 Win=262656 Len=0
10	0.000009	10.10.20.25	10.10.20.30	TLSv1.2	1754	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
11	0.000021	10.10.20.30	10.10.20.25	TCP	60	443 → 60536 [ACK] Seq=3015 Ack=1847 Win=33792 Len=0
12	0.000021	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
13	0.000022	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
14	0.000022	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data

© 2019 Microsoft Corporation. All rights reserved.

This fact can also be appreciated from the output of the tool used to perform the connection:

```
[+] Performing FCS_TLSC_EXT.4.1 Test2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge. The user is prompted for the certificate and, once it is selected, the connection succeeds:

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

23.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

23.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FCS_TLSC_EXT.4.1.

© 2019 Microsoft Corporation. All rights reserved.

24 FDP_ACF_EXT.1.1

The assurance activity for the **FDP_ACF_EXT.1.1** requirement is stated as follows:

> The evaluator will confirm that the TSS comprehensively describes the access control policy enforced by the OS. The description must include the rules by which accesses to particular files and directories are determined for particular users. The evaluator will inspect the TSS to ensure that it describes the access control rules in such detail that given any possible scenario between a user and a file governed by the OS the access control decision is unambiguous.
>
> The evaluator will create two new standard user accounts on the system and conduct the following tests:
>
> - **Test 1:** The evaluator will authenticate to the system as the first user and create a file within that user's home directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to read the file created in the first user's home directory. The evaluator will ensure that the read attempt is denied.
> - **Test 2:** The evaluator will authenticate to the system as the first user and create a file within that user's home directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to modify the file created in the first user's home directory. The evaluator will ensure that the modification is denied.
> - **Test 3:** The evaluator will authenticate to the system as the first user and create a file within that user's user directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to delete the file created in the first user's home directory. The evaluator will ensure that the deletion is denied.
> - **Test 4:** The evaluator will authenticate to the system as the first user. The evaluator will attempt to create a file in the second user's home directory. The evaluator will ensure that the creation of the file is denied.
> - **Test 5:** The evaluator will authenticate to the system as the first user and attempt to modify the file created in the first user's home directory. The evaluator will ensure that the modification of the file is accepted.
> - **Test 6:** The evaluator will authenticate to the system as the first user and attempt to delete the file created in the first user's directory. The evaluator will ensure that the deletion of the file is accepted.

© 2019 Microsoft Corporation. All rights reserved.

24.1 Documentation Review activity

24.1.1 Findings

The evaluator has reviewed the section **6.3.1. Discretionary Access Control** of the ***Security Target*** document, which is related to the access control policy and how these policies are applied to files and folders in the operating system.

This section is composed by the following main subsections:

- 6.3.1.1 Subject DAC Attributes
- 6.3.1.2 Object DAC Attributes
- 6.3.1.3 DAC Enforcement Algorithm
- 6.3.1.4 Default DAC Protection

The first and second subsections include a brief description about the security attributes for a subject and for an object.

The third one includes a step-by-step description about the algorithm used to determine whether a user has access to one file or not. This section also includes information regarding the kind of permissions which are checked at each step of the algorithm.

The fourth one includes information about the default access rules assigned to all new objects. This section also includes information about how the inheritance rules work.

Finally, the following subsections are also included in the **6.3.1. Discretionary Access Control** section.

- 6.3.1.5 DAC Management
- 6.3.1.6 Reference Mediation

In these subsections it is described the functions that manages the DACL and how the system obtains a handle for the objects.

24.1.2 Verdict

The evaluator has reviewed the information provided in the TSS section **6.3.1. Discretionary Access Control** and considers that the access control policies enforced by the operating system are properly described. This description contains enough information to allow the evaluator determine how the access control policies are applied to folders and files in the operating system, allowing or denying the access.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

24.2 Test Activity

Throughout the following tests, attempts to create, read, write and delete a test file will be performed using different users and permissions.

24.2.1 Test 1, Test 2, Test 3, Test 4, Test 5 and Test 6

24.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- User accounts with user names *userFDP1* and *userFDP2* shall not exist.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

24.2.1.2 Procedure

In order to complete these tests, the following steps shall be performed:

- Run as administrator, the script *FDP_ACF_EXT.1.ps1*. The script behaviour is explained as follows:
  - *FDP_ACF_EXT.1.ps1*: This script creates a new empty file in the *userFDP1* home directory. After that, it invokes two scripts that will perform all the test cases (First Test 1 to 3, then Test 4 to 6). The script source code is as follows:

```
. .\common\UserAndGroupManagement.ps1

#
# Test FDP_ACF_EXT.1
#

#Variables
$groupName = "Users"
$user1= "userFDP1"
$user2 = "userFDP2"
$pass = "p@ss1234"

$filePath = "Documents\fileFDP.txt"
$fullFilePathUser1 = "$env:HOMEDRIVE\Users\$user1\$filePath"
$fullFilePathUser2 = "$env:HOMEDRIVE\Users\$user2\$filePath"

# Create users for the test
Write-Host "[I]: Creating user account..."
$SecurePassword = ConvertTo-SecureString $pass -AsPlainText -Force
AddLocalUserAccount $user1 $pass
AddLocalUserAccount $user2 $pass
AddUserToGroup $user1 $groupname $False
AddUserToGroup $user2 $groupname $False
```

© 2019 Microsoft Corporation. All rights reserved.

```
#Creating a new file in user1 HOME directory
Write-Host "[I]: Creating file in user1 home directory..."
$user1Cred = New-Object System.Management.Automation.PSCredential($user1,
$SecurePassword)
Start-Process powershell -ArgumentList "-Command New-Item
$fullFilePathUser1 -type file; pause" -Credential $user1Cred
-Wait -WorkingDirectory "C:\Windows\System32"
Write-Host "[I]: Done!"

#Setup, this will copy the scripts to C directory, this way it will
inherit the everyone permissions and the scripts will run

# Script1
$inheritPath = 'C:\testFDP'
$fullInheritPath1 = $inheritPath + '\Tests1-3.ps1'
$1Path = ls '.\Tests1-3.ps1'
$script1Path = $1Path.FullName
#Script2
$fullInheritPath2 = $inheritPath + '\Tests4-6.ps1'
$2Path = ls '.\Tests4-6.ps1'
$script2Path = $2Path.FullName

mkdir C:\testFDP
Copy-Item $script1Path $inheritPath
Copy-Item $script2Path $inheritPath

#Test 1-3
Write-Host "[I]: Running Test 1 to 3 (view the new opened window)..."
$user2Cred = New-Object
System.Management.Automation.PSCredential($user2,$SecurePassword)
Start-Process powershell -Credential $user2Cred -ArgumentList
"-File $fullInheritPath1 -path $fullFilePathUser1" -Wait
-WorkingDirectory "C:\Windows\System32"
Write-Host "[I]: Done!"

#Test 4-6
Write-Host "[I]: Running Test 4 to 6 (view the new opened window)..."
Start-Process powershell -Credential $user1Cred -ArgumentList
"-File $fullInheritPath2 -pathUser1 $fullFilePathUser1 -pathUser2
$fullFilePathUser2" -Wait -WorkingDirectory "C:\Windows\System32"
Write-Host "[I]: Done!"

#Delete the temp file and folders
#Start-Process powershell -ArgumentList "-Command Remove-Item
$fullFilePathUser1; pause" -Credential $user1Cred -Wait
-WorkingDirectory "C:\Windows\System32"
Remove-Item -Recurse $inheritPath
#Remove-Item -Recurse $filePathUser1

#Remove-Item -Recurse $filePathUser2
#Start-Process powershell -ArgumentList "-Command Remove-Item -Recurse
$filePathUser1; pause" -Credential $user1Cred -Wait
```

© 2019 Microsoft Corporation. All rights reserved.

```
-WorkingDirectory "C:\Windows\System32"
#Start-Process powershell -ArgumentList "-Command Remove-Item -Recurse
 $filePathUser2; pause" -Credential $user1Cred -Wait
-WorkingDirectory "C:\Windows\System32"

#Delete the temp users
Write-Host "[I]: Deleting the user account..."
DeleteLocalUserAccount $user1
DeleteLocalUserAccount $user2

# If (-NOT
([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]
    ::GetCurrent()).IsInRole(
        [Security.Principal.WindowsBuiltInRole]
    "Administrator")) {
#     $arguments = "& '" + $myinvocation.mycommand.definition + "'"
#     Start-Process powershell -Verb runAs -ArgumentList $arguments
#     Break
# }

# Remove-Item $filePathUser1 -Recurse -Force -Confirm:$false
# Remove-Item $filePathUser2 -Recurse -Force -Confirm:$false
```

- *Test1-3.ps1*: It carries out the tests 1 to 3. All of these operations are performed using the *userFDP2* credentials. The following code block shows the source code of the script:

```
param([string]$path)
#Test 1 - Read file
Write-Host "[I]: Executing Test 1"
Write-Host "[I]: User2 attempts to read a file in user1's home directory..."
try{
    Get-Content $path -ErrorAction Stop
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}

#Test 2 - Modify file
Write-Host "[I]: Executing Test 2"
Write-Host "[I]: User2 attempts to modify a file in user1's home
    directory..."
try{
    Add-Content $path "Modified" -ErrorAction Stop
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}

#Test 3 - Delete file
Write-Host "[I]: Executing Test 3"
Write-Host "[I]: User2 attempts to delete a file in user1's home
    directory..."
```

© 2019 Microsoft Corporation. All rights reserved.

```
try{
    Remove-Item $path -ErrorAction Stop
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}
pause
```

- *Test4-6.ps1*: It carries out the tests 4 to 6. All of these operations are performed using the *userFDP1* credentials. The following image shows the source of the script:

```
param([string]$pathUser1, [string]$pathUser2)

#Test 4 - Create file
Write-Host "[I]: Executing Test 4"
Write-Host "[I]: User1 attempts to create a new file in user2's home
    directory..."
try{
    New-Item $pathUser2 -ErrorAction Stop
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}

#Test 5 - Modify a correct file
Write-Host "[I]: Executing Test 5"
Write-Host "[I]: User1 attempts to modify a file in user1's home
    directory..."
try{
    Add-Content $pathUser1 "Modified" -ErrorAction Stop
    Write-Host "[I]: Done!"
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}

#Test 6 - Delete a correct file
Write-Host "[I]: Executing Test 6"
Write-Host "[I]: User1 attempts to delete a file in user1's home
    directory..."
try{
    Remove-Item $pathUser1 -ErrorAction Stop
    Write-Host "[I]: Done!"
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}
pause
```

- To execute the script, type the following command in a *PowerShell* terminal:

  .\FDP_ACF_EXT1.ps1

- Observe the results.

24.2.1.3 Results

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

During the execution of tests 1, 2 and 3, the evaluator has attempted to read, modify and delete a file stored in *userFDP1´s* home directory. *UserFDP2* does not have access to files stored in the *userFDP1's* home directory. Therefore *userFDP2* cannot read, modify or delete them; as it can be shown in the following image:

```
[I]: Executing Test 1
[I]: User2 attempts to read a file in user1's home directory...
WARNING: Access is denied
[I]: Executing Test 2
[I]: User2 attempts to modify a file in user1's home directory...
WARNING: Access is denied
[I]: Executing Test 3
[I]: User2 attempts to delete a file in user1's home directory...
WARNING: Access is denied
Press Enter to continue...:
```

During the execution of tests 4, 5 and 6, the evaluator has attempted to read, modify and delete a file stored in *userFDP2´s* home directory. *UserFDP1* does not have permission to create a file in *userFDP2's* home directory, but has permissions to modify or delete files stored in its home directory. As it can be shown in the following picture:

```
[I]: Executing Test 4
[I]: User1 attempts to create a new file in user2's home directory...
WARNING: Access to the path 'C:\Users\userFDP2\Documents\fileFDP.txt' is denied.
[I]: Executing Test 5
[I]: User1 attempts to modify a file in user1's home directory...
[I]: Done!
[I]: Executing Test 6
[I]: User1 attempts to delete a file in user1's home directory...
[I]: Done!
Press Enter to continue...:
```

© 2019 Microsoft Corporation. All rights reserved.

These behaviours match with the ones described in the assurance activity section.

24.2.1.4 Verdict

As it can be observed in the above section, the obtained behaviours match with the ones described in the assurance activity section. Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1, Test 2, Test 3, Test4, Test 5 and Test 6** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1, Test 2, Test 3, Test 4, Test 5 and Test 6**.

24.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FDP_ACF_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

25 FDP_IFC_EXT.1.1

The assurance activity for the **FDP_IFC_EXT.1.1** requirement is stated as follows:

> The evaluator will verify that the TSS section of the ST describes the routing of IP traffic when a VPN client is enabled. The evaluator will ensure that the description indicates which traffic does not go through the VPN and which traffic does, and that a configuration exists for each in which only the traffic identified by the ST author as necessary for establishing the VPN connection (IKE traffic and perhaps HTTPS or DNS traffic) is not encapsulated by the VPN protocol (IPsec). The evaluator will perform the following test:
>
> - **Test 1**:
> - **Step 1**: The evaluator will enable a network connection. The evaluator will sniff packets while performing running applications that use the network such as web browsers and email clients. The evaluator will verify that the sniffer captures the traffic generated by these actions, turn off the sniffing tool, and save the session data.
> - **Step 2**: The evaluator will configure an IPsec VPN client that supports the routing specified in this requirement. The evaluator will turn on the sniffing tool, establish the VPN connection, and perform the same actions with the device as performed in the first step. The evaluator will verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.
> - **Step 3**: The evaluator will examine the traffic from both step one and step two to verify that all non-excepted Data Plane traffic in Step 2 is encapsulated by IPsec. The evaluator will examine the Security Parameter Index (SPI) value present in the encapsulated packets captured in Step 2 from the TOE to the Gateway and shall verify this value is the same for allactions used to generate traffic through the VPN. Note that it is expected that the SPI value for packets from the Gateway to the TOE is different than the SPI value for packets from the TOE to the Gateway.
> - **Step 4**: The evaluator will perform a ping on the TOE host on the local network and verify that no packets sent are captured with the sniffer. The evaluator will attempt to send packets to the TOE outside the VPN tunnel (i.e. not through the VPN gateway), including from the local network, and verify that the TOE discards them.

25.1 Documentation Review activity

25.1.1 Findings

The evaluator has reviewed the section **6.3.2 VPN Client** of the ***Security Target*** document. This section states the following:

© 2019 Microsoft Corporation. All rights reserved.

The Windows IPsec VPN client can be configured by the device local administrator. The administrator can also configure the IPsec VPN client that all IP traffic is routed through the IPsec tunnel except for:

- *IKE traffic used to establish the VPN tunnel.*
- *IPv4 ARP traffic for resolution of local network layer addresses and to establish a local address.*
- *IPv6 NDP traffic for resolution of local network layer addresses and to establish a local address.*

The components responsible for routing IP traffic through the VPN client:

- *The **IPv4/IPv6 network stack** in the kernel processes ingoing and outgoing network traffic.*
- *The **IPsec** and **IKE and AuthIP Keying Modules** service which hosts the IKE and Authenticated Internet Protocol (AuthIP) keying modules. These keying modules are used for authentication and key exchange in Internet Protocol security (IPsec).*
- *The **Remote Access Service** device driver in the kernel, which is used primarily for VPN connections; known as the "RAS IPsec VPN" or "RAS VPN".*
- *The **IPsec Policy Agent** service which enforces IPsec policies.*

*Universal Windows App developers can implement their own VPN client if authorized by Microsoft to use the **networkingVpnProvider** capability, which includes setting the policy to lockdown networking traffic as described above.*

This section does not state conformance with any native VPN client, it only provides the available interfaces for applications can use to establish a secure tunnel. Moreover, the TSS contains a pointer to the MSDN website describing all the available interfaces and information about VPN configurations: Windows.Networking.Vpn Namespace

25.1.2 Verdict

The evaluator considers that the information provided in the TSS describes in detail how to configure a VPN channel for external applications taking into account its different phases and providing the set of available interfaces required for doing that.

Also, once the VPN channel is established, the traffic that does not go through the VPN and the traffic that goes through it, is the same as the one described in the TSS section.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

25.2 Test Activity

25.2.1 Test 1 - Information Flow Control

The evaluator shall ensure which traffic does go through the VPN and which traffic does not.

25.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A network protocol analyzer (e.g. *Wireshark*) is available.
- Command Prompt (*cmd.exe*).
- Microsoft Edge (*MicrosoftEdge.exe*).
- *Netcat* computer networking utility (*nc.exe*) is available.
- VPN Server with L2TP/IPsec protocol enabled shall be available.
- *Testing machine* connected to the TOE local network.

25.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using administrator account.
- Open *Wireshark* and configure it to listen through the active network interface.
- Run *MicrosoftEdge.exe* and navigate to *Microsoft Outlook*.
- Then, verify the captured network traffic in *Wireshark*. A secure connection between the TOE and the server shall be established over a secure and trusted channel.
- Save the captured packages and close *Wireshark* session.
- Go to *Settings->Network & Internet ->VPN*. Then click in *Add a VPN connection*.
- Configure the VPN connection as shown in the image below (*Pre-shared key and Password: vpn*):

© 2019 Microsoft Corporation. All rights reserved.

- Open *Wireshark* and configure it to listen through the active network interface.
- Go to *Settings->Network & Internet ->VPN*. Then click on *VPN-Test* and *Connect*.
- Run *MicrosoftEdge.exe* and navigate to *Microsoft Outlook*.
- After that, verify the captured network traffic in *Wireshark* and observe how the communication is encapsulated by Internet Protocol Security (*IPsec*).
- Save captured packages.
- Review the last captured packages and verify whether *Security Parameter Index (SPI)* value present in the encapsulated packages is the same for packets from the *Gateway* to the TOE and it is different for packets from the TOE to the *Gateway*.
- Close *Wireshark* session.
- Start *Wireshark* session again and configure it to listen through the active network interface.
- Open a *Command Prompt* and ping to the *Testing machine* located in the same local network.
- Then, attempt to send packets to the TOE outside the VPN tunnel.
- Open a *Command Prompt* and use *netcat* to open a TCP network tunnel.
- Go to Testing machine, open a command line terminal and establish a network connection using the opened TCP tunnel. Once the communication is establised, attempt to send data information to the TOE.

© 2019 Microsoft Corporation. All rights reserved.

- Go back to the TOE and review captured packages in *Wireshark*, save session data and close *Wireshark* application.

25.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

In the first part of Test 1, the evaluator navigate to *Microsoft Outlook* using *MicrosoftEdge.exe* browser and capturing the network communication through *Wireshark* as shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
62	1.156368	40.77.226.250	192.168.0.111	TLSv1.2	746	[TCP Spurious Retransmission] , Application Data
64	1.156524	204.79.197.200	192.168.0.111	TLSv1.2	174	[TCP Spurious Retransmission] , Application Data
66	1.156576	192.168.0.111	40.77.226.250	TLSv1.2	286	Client Hello
73	1.204850	40.77.226.250	192.168.0.111	TLSv1.2	1212	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
75	1.217390	192.168.0.111	40.77.226.250	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
77	1.299237	192.168.0.111	40.77.226.250	TLSv1.2	112	Application Data
78	1.302691	192.168.0.111	13.107.42.11	TLSv1.2	277	Application Data
80	1.332459	40.77.226.250	192.168.0.111	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
83	1.355984	13.107.42.11	192.168.0.111	TLSv1.2	141	Application Data
85	1.356168	13.107.42.11	192.168.0.111	TLSv1.2	92	Application Data
88	1.363450	40.77.226.250	192.168.0.111	TLSv1.2	490	Application Data
90	1.403648	192.168.0.111	40.77.226.250	TLSv1.2	1048	Application Data

Then, for the second part of Test 1, the evaluator has configured a VPN connection between TOE and VPN Server with *L2TP/IPsec* protocol enabled.

© 2019 Microsoft Corporation. All rights reserved.

Once the VPN connection is established, the evaluator has performed the same action as the first part, capturing network packets with *Wireshark*, but in this case the communication is being encrypted by Internet Protocol Security (*IPsec*) as shown below:

No.	Time	Source	Destination	Protocol	Length	Info
18	4.865984	192.168.0.111	212.237.211.44	ESP	206	ESP (SPI=0x12175947)
19	4.943333	212.237.211.44	192.168.0.111	ESP	190	ESP (SPI=0x1923b494)
20	4.943582	192.168.0.111	212.237.211.44	ESP	110	ESP (SPI=0x12175947)
21	4.943716	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
22	5.024586	212.237.211.44	192.168.0.111	ESP	110	ESP (SPI=0x1923b494)
23	5.024588	212.237.211.44	192.168.0.111	ESP	126	ESP (SPI=0x1923b494)
24	5.024807	192.168.0.111	212.237.211.44	ESP	142	ESP (SPI=0x12175947)
25	5.033138	192.168.0.111	212.237.211.44	ESP	126	ESP (SPI=0x12175947)
26	5.110199	212.237.211.44	192.168.0.111	ESP	110	ESP (SPI=0x1923b494)
27	5.110421	212.237.211.44	192.168.0.111	ESP	110	ESP (SPI=0x1923b494)
28	5.110682	192.168.0.111	212.237.211.44	ESP	110	ESP (SPI=0x12175947)
29	5.111145	212.237.211.44	192.168.0.111	ESP	126	ESP (SPI=0x1923b494)
30	5.111457	192.168.0.111	212.237.211.44	ESP	110	ESP (SPI=0x12175947)
31	5.189151	212.237.211.44	192.168.0.111	ESP	110	ESP (SPI=0x1923b494)
32	5.189153	212.237.211.44	192.168.0.111	ESP	110	ESP (SPI=0x1923b494)

In the third part of Test 1, review the last captured packets and find the *Security Parameter Index (SPI)* parameter present in the encapsulated packets:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
176	15.000093	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
177	15.012457	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
178	15.018220	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
179	15.188430	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
180	15.189922	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
181	15.189923	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
182	15.189924	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
183	15.189925	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
184	15.190357	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
185	15.190361	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
186	15.190362	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
187	15.190644	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
188	15.190819	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
189	15.190938	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
190	15.190939	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
191	15.191179	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x1923b494)
192	15.191180	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
193	15.203182	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
194	15.203951	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)
195	15.203953	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0x1923b494)

No.	Time	Source	Destination	Protocol	Length	Info
232	16.272052	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
233	16.272350	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
234	16.274840	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
235	16.275105	192.168.0.111	212.237.211.44	ESP	190	ESP (SPI=0x12175947)
236	16.275471	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
237	16.275691	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
238	16.276043	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
239	16.276266	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
240	16.276612	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
241	16.277073	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
242	16.277305	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
243	16.277659	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
244	16.277984	192.168.0.111	212.237.211.44	ESP	254	ESP (SPI=0x12175947)
246	16.279381	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)
247	16.279993	192.168.0.111	212.237.211.44	ESP	158	ESP (SPI=0x12175947)

As shown in the images above the *Security Parameter Index (SPI)* value is the same for packets that are being sending from the *Gateway* to the TOE, and it is different for packets from the TOE to the *Gateway*.

Finally, for the last part of the Test 1, the evaluator perform a ping request to the Testing machine. No packets sent are captured by *Wireshark* as shown in the image below:

No.	Time	Source	Destination	Protocol	Length	Info
372…	2358.042283	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0xb5400864)
372…	2358.042759	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0xb5400864)
372…	2358.043706	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0xb5400864)
372…	2358.043708	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0xb5400864)
372…	2358.043709	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0xb5400864)
372…	2358.043710	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.044141	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.044143	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.044503	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.044710	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.045944	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.045952	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.045953	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.045954	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.046377	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.046378	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.046613	212.237.211.44	192.168.0.111	ESP	398	ESP (SPI=0xb5400864)
372…	2358.046614	212.237.211.44	192.168.0.111	ESP	126	ESP (SPI=0xb5400864)
372…	2358.046615	212.237.211.44	192.168.0.111	UDPENCAP	43	NAT-keepalive
372…	2358.047379	192.168.0.111	212.237.211.44	ESP	126	ESP (SPI=0x2aa36fe4)
372…	2358.294068	192.168.0.106	224.0.0.251	MDNS	130	Standard query 0x0000 PTR _companion-link._tcp.local, "QU" question P
372…	2359.319051	192.168.0.106	224.0.0.251	MDNS	112	Standard query 0x0000 PTR _sleep-proxy._udp.local, "QU" question OPT
372…	2360.342179	192.168.0.106	224.0.0.251	MDNS	112	Standard query 0x0000 PTR _sleep-proxy._udp.local, "QM" question OPT
372…	2361.900578	212.237.211.44	192.168.0.111	ESP	414	ESP (SPI=0xb5400864)

© 2019 Microsoft Corporation. All rights reserved.

After that the evaluator attempt to send packets to the TOE outside the VPN tunnel (i.e. from local network).

Open a TCP tunnel between TOE and Testing machine using *netcat*, then try to send information from Testing machine to the TOE through established TCP tunnel:

Command Prompt - nc.exe -lvp 4443

```
C:\Users\evaluador>nc.exe -lvp 4443
listening on [any] 4443 ...
```

root@kali: ~ 66x32

```
root@kali:~# nc 192.168.0.111 4443
Testing
```

No.	Time	Source	Destination	Protocol	Length	Info
134…	1571.855607	192.168.0.111	212.237.211.44	ESP	110	ESP (SPI=0x78049953)
134…	1573.024741	D-LinkIn_35:3b:42	Broadcast	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
134…	1573.035841	212.237.211.44	192.168.0.111	ESP	126	ESP (SPI=0x45312af2)
134…	1573.036444	192.168.0.111	212.237.211.44	ESP	126	ESP (SPI=0x78049953)
134…	1574.048879	D-LinkIn_35:3b:42	Broadcast	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
134…	1574.825391	D-LinkIn_35:3b:42	Broadcast	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
134…	1574.825393	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x45312af2)
134…	1575.273735	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x45312af2)
134…	1575.291083	212.237.211.44	192.168.0.111	ESP	446	ESP (SPI=0x45312af2)
134…	1576.622570	D-LinkIn_35:3b:42	Broadcast	ARP	42	Who has 192.168.0.102? Tell 192.168.0.1
134…	1576.622575	192.168.0.1	239.255.255.250	SSDP	311	NOTIFY * HTTP/1.1
134…	1576.622576	192.168.0.1	239.255.255.250	SSDP	374	NOTIFY * HTTP/1.1
134…	1576.622576	192.168.0.1	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
134…	1576.622577	192.168.0.1	239.255.255.250	SSDP	350	NOTIFY * HTTP/1.1
134…	1576.622578	192.168.0.1	239.255.255.250	SSDP	382	NOTIFY * HTTP/1.1
134…	1576.629800	212.237.211.44	192.168.0.111	ESP	430	ESP (SPI=0x45312af2)
134…	1576.679312	192.168.0.1	239.255.255.250	SSDP	302	NOTIFY * HTTP/1.1
134…	1576.679313	192.168.0.1	239.255.255.250	SSDP	374	NOTIFY * HTTP/1.1
134…	1576.679314	192.168.0.1	239.255.255.250	SSDP	370	NOTIFY * HTTP/1.1
134…	1576.679315	192.168.0.1	239.255.255.250	SSDP	350	NOTIFY * HTTP/1.1
134…	1576.679316	192.168.0.1	239.255.255.250	SSDP	382	NOTIFY * HTTP/1.1
134…	1576.679317	192.168.0.1	239.255.255.250	SSDP	364	NOTIFY * HTTP/1.1

There is no data information in the *Wireshark* capture since TOE discard them.

© 2019 Microsoft Corporation. All rights reserved.

25.2.2 Verdict

The evaluator considers that the information provided in the TSS describes in detail how to configure a VPN channel for external applications, taking into account its different phases and providing the set of available interfaces required for doing that.

Also, once the VPN channel is established, the traffic that does not go through the VPN and the traffic that goes through it, is the same as the one described in the TSS section.

When the VPN is connected, all the described traffic is encrypted using the Internet Protocol Security (*IPsec*) and no data is sent in plain text.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

25.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FDP_IFC_EXT.1.1 requirement.

© 2019 Microsoft Corporation. All rights reserved.

26 FIA_AFL.1.1

The assurance activity for the **FIA_AFL.1.1** requirement is stated as follows:

> The evaluator will set an administrator-configurable threshold for failed attempts, or note the ST-specified assignment. The evaluator will then (per selection) repeatedly attempt to authenticate with an incorrect password, PIN, or certificate until the number of attempts reaches the threshold. Note that the authentication attempts and lockouts must also be logged as specified in FAU_GEN.1.

26.1 Documentation Review activity

26.1.1 Findings

The related assurance activity does not define any action for this requirement.

26.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

26.2 Test Activity

The assurance activity states that a threshold for failed authentication attempts shall be configured by the OS administrator. After that, the evaluator will then repeatedly attempt to authenticate with an incorrect password until the threshold is reached.

This test is done at the same time as *FIA_AFL1.2* requirement.

26.2.1 Test

26.2.1.1 Setup

The applicable setup for this test is covered in the Test Activity section for *FIA_AFL.1.2* requirement.

26.2.1.2 Procedure

This test is done at the same time as *FIA_AFL1.2*.

26.2.1.3 Results

The results for this test are included in the Test Activity section for *FIA_AFL.1.2* requirement.

© 2019 Microsoft Corporation. All rights reserved.

26.2.1.4 Verdict

The verdict for this test is assigned in the Test Activity section for FIA_AFL.1.2 requirement.

26.3 Final Verdict

The final verdict for this requirement is assigned in Test Activity section for the FIA_AFL.1.2 requirement.

© 2019 Microsoft Corporation. All rights reserved.

27 FIA_AFL.1.2

The assurance activity for the **FIA_AFL.1.2** requirement is stated as follows:

> **Test 1:** The evaluator will attempt to authenticate repeatedly to the system with a known bad password. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.
>
> **Test 2:** The evaluator will attempt to authenticate repeatedly to the system with a known bad certificate. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.
>
> **Test 3:** The evaluator will attempt to authenticate repeatedly to the system using both a bad password and a bad certificate. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.

27.1 Documentation Review activity

27.1.1 Findings

The related assurance activity does not define any action for this requirement.

27.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

27.2 Test Activity

During this test, the evaluator will attempt to authenticate repeatedly to the system with a known bad password. Once the threshold for failed authentication attempts limit is reached, the evaluator will ensure that the account is locked. In addition, the audit events will be analyzed to demonstrate the fulfilment of the requirement *FAU_GEN.1* related with lock and unlock of user accounts.

© 2019 Microsoft Corporation. All rights reserved.

27.2.1 Test 1

27.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A user account with user name *userFIA* shall not exist.
- An administrator account shall be available.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.
- Additionally, for OE-protected storage, Windows 10 can authenticate users with a Windows Hello PIN which is backed by a TPM. Platforms with TPM must be configured with Windows Hello PIN authentication.

27.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into a user account with administrator rights.
- Run the script *FIA_AFL.1.2 - Setup.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script is stored and type the following command:

```
.\FIA_AFL.1.2 - Setup.ps1
```

- The script source code is as follows:

```
#
# Test FIA_AFL.1
#
$pass = "p@ss1234"

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Logon" /success:enable /failure:enable
auditpol /set /subcategory:"User Account Management" /success:enable
/failure:enable
auditpol /set /subcategory:"Account Lockout" /success:enable /failure:enable
wevtutil cl security

Write-Host "[I]: Configuring authentication attemps threshold..."
net accounts /lockoutthreshold:3 /lockoutwindow:15 /lockoutduration:15

Write-Host "[I]: Setting up user account 'userFIA' for the test"
$SecurePassword = ConvertTo-SecureString $pass -AsPlainText -Force
New-LocalUser -Name "userFIA" -Password $SecurePassword
Add-LocalGroupMember -Group "Users" -Member "userFIA"

Write-Host "[I]: In order to perform this test, the current user account
will be locked, please log in as 'userFIA' using an invalid password three
times to lock the account"
Write-Host "[I]: Afterwards, log into an administrator account, but now with
a valid password ..."
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Please, read the instructions carefully"
pause
logoff
```

- The script will set the audit rules necessary for the test and configure the threshold for unsuccessful authentication attempts to 3 attempts.
- Once the threshold has been configured by the script, the current session account will be locked and the login screen will be shown. Now, the evaluator shall attempt to perform a login using the user account created by the script (*userFIA*) using a known bad password three times.

- After three failed authentication attempts, the threshold is reached and an error message shall be shown.
- Then, the evaluator shall perform a valid login attempt using a user account with administrator rights.
- To unlock the *userFIA* account, the Computer Management tool (*WIN + X, then 'g' key*) shall be opened. Then, go to *System Tools -> Local Users and Groups -> Users*. Select the user account which has been locked (*userFIA*).
- Ensure that this account is locked. To do that, verify that the *'Account is locked out'* check box is checked.

© 2019 Microsoft Corporation. All rights reserved.

- Uncheck the checkbox to unlock the user account.
- Finally, run the script *FIA_AFL.1.2 - Audit&Restore.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

```
.\FIA_AFL.1.2 — Audit&Restore.ps1
```

- The script source code is as follows:

```
#
# Test FIA_AFL.1
#
Write-Host "[I]: Getting current state of userFIA to unlock it..."
Write-Host "[I]: Please, go 'Local Users and Groups' menu and then to
'Users' Folder. Select the userFIA, right click on it and then check lock
status and unlock it"
Write-Host "[I]: Please, read the instructions carefully..."
pause
Start-Process compmgmt.msc
Write-Host "[I]: Press enter if the previous operations have been already
    performed..."
pause
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Getting audit log entries..."
Get-EventLog Security -InstanceId 4740 -Newest 1 | Format-List *
Get-EventLog Security -InstanceId 4767 -Newest 1 | Format-List *
Get-EventLog Security -InstanceId 4625 -Newest 1 | Format-List *

Write-Host "[I]: Restoring audit rules..."
auditpol /set /subcategory:"Logon" /success:disable /failure:disable
auditpol /set /subcategory:"User Account Management" /success:disable
    /failure:disable
auditpol /set /subcategory:"Account Lockout" /success:disable /failure:disable

Write-Host "[I]: Removing test user account..."
Remove-LocalUser -Name userFIA
```

- Finally, the previous steps shall be repeated, but this time, the userFIA user account must be configured with Windows Hello PIN. The necessary steps to set up this authentication method are described in the FIA_UAU.5.1 requirement.
- Once the user account is enabled and configured with Windows Hello PIN, the evaluator shall attempt to perform a login using the user account created by the script (*userFIA*) using a known bad PIN three times.
- After three failed authentication attempts, the threshold is reached and an error message shall be shown.
- Then, the evaluator shall perform a valid login attempt using a user account with administrator rights.
- To unlock the *userFIA* account, the Computer Management tool (*WIN + X, then 'g' key*) shall be opened. Then, go to *System Tools -> Local Users and Groups -> Users*. Select the user account which has been locked (*userFIA*).
- Ensure that this account is locked. To do that, verify that the *'Account is locked out'* check box is checked.

© 2019 Microsoft Corporation. All rights reserved.

- Uncheck the checkbox to unlock the user account.
- Finally, run the script *FIA_AFL.1.2 - Audit&Restore.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

```
.\FIA_AFL.1.2 — Audit&Restore.ps1
```

- Observe the results.

27.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

During the test, the evaluator has configured the threshold for failed authentication attempts to three attempts and has attempted three times to login using a known bad password/PIN. Once the threshold had been reached, the user account was locked, as it can be observed in the following images:

© 2019 Microsoft Corporation. All rights reserved.

Once the user account has been locked, the evaluator has logged into the computer using a user account with administrator rights and has executed the script *FIA_AFL.1.2 - Audit&Restore.ps1* to obtain the generated audit events. For each unsuccessful authentication attempt an audit event like the following has been generated, indicating that an invalid authentication attempt was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : An account failed to log on.

                       Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3E7

                       Logon Type:                      2

                       Account For Which Logon Failed:
                        Security ID:            S-1-0-0
                        Account Name:           userFIA
                        Account Domain:         SURFACEBOOK2

                       Failure Information:
                        Failure Reason:         Account locked out.
                        Status:                 0xC0000234
                        Sub Status:             0x0

                       Process Information:
                        Caller Process ID:      0x894
                        Caller Process Name:    C:\Windows\System32\svchost.exe

                       Network Information:
                        Workstation Name:       SURFACEBOOK2
                        Source Network Address: 127.0.0.1
                        Source Port:            0

                       Detailed Authentication Information:
                        Logon Process:          User32
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                       This event is generated when a logon request fails. It is generated on the computer where
                       access was attempted.

                       The Subject fields indicate the account on the local system which requested the logon. This is
                       most commonly a service such as the Server service, or a local process such as Winlogon.exe or
                       Services.exe.

                       The Logon Type field indicates the kind of logon that was requested. The most common types are
                       2 (interactive) and 3 (network).

                       The Process Information fields indicate which account and process on the system requested the
                       logon.

                       The Network Information fields indicate where a remote logon request originated. Workstation
                       name is not always available and may be left blank in some cases.

                       The authentication information fields provide detailed information about this specific logon
                       request.
                        - Transited services indicate which intermediate services have participated in this logon
                       request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session
                       key was requested.
Id                   : 4625
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12546
Opcode               : 0
Keywords             : -9218868437227405312
RecordId             : 14059
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 852
ThreadId             : 952
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 10:40:15 AM
ActivityId           : 6141e5e7-8bfe-0000-6de6-4161fe8bd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Account Lockout
KeywordsDisplayNames : {Audit Failure}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

Once the configured threshold has been reached and the user account has been locked, the following audit event has been generated showing this fact:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A user account was locked out.

                       Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3E7

                       Account That Was Locked Out:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1003
                        Account Name:           userFIA

                       Additional Information:
                        Caller Computer Name:   SURFACEBOOK2
Id                   : 4740
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13824
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 14055
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 852
ThreadId             : 952
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 10:40:14 AM
ActivityId           : 6141e5e7-8bfe-0007-0dee-4161fe8bd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : User Account Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

Finally, the evaluator has unlocked the user account, which generates the following audit event:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A user account was unlocked.

                       Subject:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x1C218D

                       Target Account:
                        Security ID:            S-1-5-21-1598544264-1886666874-1118049692-1003
                        Account Name:           userFIA
                        Account Domain:         SURFACEBOOK2
Id                   : 4767
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13824
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 14302
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 852
ThreadId             : 8396
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/4/2018 10:42:41 AM
ActivityId           : 6141e5e7-8bfe-0000-6de6-4161fe8bd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : User Account Management
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

27.2.1.4 Verdict

The evaluator has performed invalid authentication attempts using a known bad password/PIN until the configured threshold has been reached. Once the user account has been locked, the evaluator has unlocked this account using a user account with administrator rights.

As the above images state, the user account has been locked successfully after the configured threshold is reached. Additionally, both invalid authentication attempts and the account lockout events have been audited as specified in *FAU_GEN.1*.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

27.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FIA_AFL.1.1 and FIA_AFL.1.2.

© 2019 Microsoft Corporation. All rights reserved.

28 FIA_UAU.5.1

The assurance activity for the **FIA_UAU.5.1** requirement is stated as follows:

> If user name and password authentication is selected, the evaluator will configure the OS with a known user name and password and conduct the following tests:
>
> - **Test 1:** The evaluator will attempt to authenticate to the OS using the known user name and password. The evaluator will ensure that the authentication attempt is successful.
> - **Test 2:** The evaluator will attempt to authenticate to the OS using the known user name but an incorrect password. The evaluator will ensure that the authentication attempt is unsuccessful.
>
> If user name and PIN that releases an asymmetric key is selected, the evaluator will examine the TSS for guidance on supported protected storage and will then configure the TOE or OE to establish a PIN which enables release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the OS can interface. The evaluator will then conduct the following tests:
>
> - **Test 1:** The evaluator will attempt to authenticate to the OS using the known user name and PIN. The evaluator will ensure that the authentication attempt is successful.
> - **Test 2:** The evaluator will attempt to authenticate to the OS using the known user name but an incorrect PIN. The evaluator will ensure that the authentication attempt is unsuccessful.
>
> If X.509 certificate authentication is selected, the evaluator will generate an X.509v3 certificate for a user with the Client Authentication Enhanced Key Usage field set. The evaluator will provision the OS for authentication with the X.509v3 certificate. The evaluator will ensure that the certificates are validated by the OS as per FIA_X509_EXT.1.1 and then conduct the following tests:
>
> - **Test 1:** The evaluator will attempt to authenticate to the OS using the X.509v3 certificate. The evaluator will ensure that the authentication attempt is successful.
> - **Test 2:** The evaluator will generate a second certificate identical to the first except for the public key and any values derived from the public key. The evaluator will attempt to authenticate to the OS with this certificate. The evaluator will ensure that the authentication attempt is unsuccessful.

28.1 Documentation Review activity

28.1.1 Findings

The ***Security Target*** document, defines the available authentication mechanisms available in its section **5.1.4.2 Multiple Authentication Mechanisms (FIA_UAU.5)**.

© 2019 Microsoft Corporation. All rights reserved.

FIA_UAU.5.1 The OS shall provide the following authentication mechanisms:
[
- ***Authentication based on user name and password,***
- ***authentication based on user name and a PIN that releases an asymmetric key stored in OE-protected storage***[10]

] to support user authentication.

The evaluator has reviewed the respective TSS section (**6.4 Identification and Authentication**) of the ***Security Target*** document, where it is described which authentication mechanism is supported by each Windows OS edition. From the previous section, the following information can be extracted:

- All Windows editions supports authentication based on username and password.
- Additionally, for OE-protected storage, Windows 10 can authenticate users with a Windows Hello PIN which is backed by a TPM.
- For Windows 10 and Windows Server, a physical or virtual smart card, can be also used.

In addition, the ***Operational Guidance*** document, includes also in its sections 6.8 and 6.9 the necessary information and supporting documentation to set up or configure the environment for each authentication mechanism.

28.1.2 Verdict

The evaluator has reviewed the ***Security Target*** document and has reviewed each authentication mechanism available and the information on the TSS about the supported protected storage. Moreover, the ***Operational Guidance*** document, has been also analyzed, checking the setup and configuration procedures for each authentication mechanism.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

28.2 Test Activity

The evaluator will attempt authenticate to the OS using a known user name and the correct password or PIN. Then the evaluator will repeat the process using a known user name but an incorrect password or PIN. The evaluator will ensure that the authentication is successful in the first test and fails in the second.

28.2.1 Test 1 & Test 2 - Username and password

28.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

© 2019 Microsoft Corporation. All rights reserved.

- A standard user account with user name *userFIA* shall not exist. This account shall belong to default *Users* group.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

28.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Run as administrator, the script *FIA_UAU.5.1 - Test 1&2.ps1*. The script behaviour is explained as follows:
  - *FIA_UAU.5.1 - Test 1&2.ps1*: This script creates a new user with name *userFIA*. Afterwards, the script lockouts the current user. Then, a manual login with valid credentials of the recently created user is needed. Finally, the same operation shall be performed, but this time the login should be done with an invalid password. The script source code is as follows:

```
#
# Test FIA_UAU.5.1
#
$pass = "p@ss1234"

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Logon" /success:enable /failure:enable
wevtutil cl security

Write-Host "[I]: Setting up user account 'userFIA' for the test"
$SecurePassword = ConvertTo-SecureString $pass -AsPlainText -Force
New-LocalUser -Name "userFIA" -Password $SecurePassword
Add-LocalGroupMember -Group "Users" -Member "userFIA"

Write-Host "[I]: In order to perform this test, please log in as'userFIA' using
    an valid password"
Write-Host "[I]: Please, read the instructions carefully"
pause

logoff
Write-Host "[I]: Getting audit log entries..."
pause
Get-EventLog Security -InstanceId 4624 -Newest 1 | Format-List *

Write-Host "[I]: In order to perform this test, please log in as'userFIA' using
    an invalid password, afterwards introduce a validpassword"
Write-Host "[I]: Please, read the instructions carefully"
pause

logoff
Write-Host "[I]: Getting audit log entries..."
pause
Get-EventLog Security -InstanceId 4625 -Newest 1 | Format-List *
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Removing test user account..."
Remove-LocalUser -Name userFIA

Write-Host "[I]: Cleaning up audit rules..."
auditpol /set /subcategory:"Logon" /success:disable /failure:disable
```

- Observe the results

28.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to login using a known correct user name and password. The login was successful and the next image shows the generated audit event:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID            : 4624
MachineName        : SURFACEBOOK2
Data               : {}
Index              : 19795
Category           : (12544)
CategoryNumber     : 12544
EntryType          : SuccessAudit
Message            : An account was successfully logged on.

                     Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3e7

                     Logon Information:
                        Logon Type:             5
                        Restricted Admin Mode:  -
                        Virtual Account:        %%1843
                        Elevated Token:         %%1842

                     Impersonation Level:       %%1833

                     New Logon:
                        Security ID:            S-1-5-18
                        Account Name:           SYSTEM
                        Account Domain:         NT AUTHORITY
                        Logon ID:               0x3e7
                        Linked Logon ID:        0x0
                        Network Account Name:   -
                        Network Account Domain: -
                        Logon GUID:             {00000000-0000-0000-0000-000000000000}

                     Process Information:
                        Process ID:             0x338
                        Process Name:           C:\Windows\System32\services.exe

                     Network Information:
                        Workstation Name:       -
                        Source Network Address: -
                        Source Port:            -

                     Detailed Authentication Information:
                        Logon Process:          Advapi
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                     This event is generated when a logon session is created. It is generated on the computer that was accessed.

                     The subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server
                     service, or a local process such as Winlogon.exe or Services.exe.

                     The logon type field indicates the kind of logon that occurred. The most common types are 2 (interactive) and 3 (network).

                     The New Logon fields indicate the account for whom the new logon was created, i.e. the account that was logged on.

                     The network fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank in some cases.

                     The impersonation level field indicates the extent to which a process in the logon session can impersonate.

                     The authentication information fields provide detailed information about this specific logon request.
                        - Logon GUID is a unique identifier that can be used to correlate this event with a KDC event.
                        - Transited services indicate which intermediate services have participated in this logon request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session key was requested.
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-18, SURFACEBOOK2$, WORKGROUP, 0x3e7...}
InstanceId         : 4624
TimeGenerated      : 12/5/2018 10:37:38 AM
TimeWritten        : 12/5/2018 10:37:38 AM
UserName           :
Site               :
Container          :
```

After that, the evaluator has performed an authentication attempt using a correct user name but an invalid password. The login was unsuccessful as can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID            : 4625
MachineName        : SURFACEBOOK2
Data               : {}
Index              : 19801
Category           : (12544)
CategoryNumber     : 12544
EntryType          : FailureAudit
Message            : An account failed to log on.

                     Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3e7

                     Logon Type:                       2

                     Account For Which Logon Failed:
                        Security ID:            S-1-0-0
                        Account Name:           userFIA
                        Account Domain:         SURFACEBOOK2

                     Failure Information:
                        Failure Reason:         %%2313
                        Status:                 0xc000006d
                        Sub Status:             0xc000006a

                     Process Information:
                        Caller Process ID:      0x950
                        Caller Process Name:    C:\Windows\System32\svchost.exe

                     Network Information:
                        Workstation Name:       SURFACEBOOK2
                        Source Network Address: 127.0.0.1
                        Source Port:            0

                     Detailed Authentication Information:
                        Logon Process:          User32
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                     This event is generated when a logon request fails. It is generated on the computer where access was attempted.

                     The Subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server
                     service, or a local process such as Winlogon.exe or Services.exe.

                     The Logon Type field indicates the kind of logon that was requested. The most common types are 2 (interactive) and 3 (network).

                     The Process Information fields indicate which account and process on the system requested the logon.

                     The Network Information fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank
                     in some cases.

                     The authentication information fields provide detailed information about this specific logon request.
                        - Transited services indicate which intermediate services have participated in this logon request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session key was requested.
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-18, SURFACEBOOK2$, WORKGROUP, 0x3e7...}
InstanceId         : 4625
TimeGenerated      : 12/5/2018 10:37:58 AM
TimeWritten        : 12/5/2018 10:37:58 AM
UserName           :
Site               :
Container          :
```

28.2.1.4 Verdict

The evaluator has performed an authentication attempt using a valid user name and password that has been completed successfully. Afterwards, the evaluator has performed another authentication attempt but this time, using an invalid password, ensuring that this authentication attempt failed.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfillment of the **Test 1 and Test 2 - Username and password** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1 and Test 2 - Username and password**.

28.2.2 Test 1 & Test 2 - Username and a PIN that releases an asymmetric key stored in OE-protected storage (TPM)

28.2.2.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A standard user account with user name *userFIA* shall exist. This account shall belong to default *Users* group.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

© 2019 Microsoft Corporation. All rights reserved.

28.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- First, log into the user account with username *userFIA*. Then, open the Settings application (*WIN + I*).

- After that, select *Accounts* option in the menu and then *Sign-in options* in the left menu. In the PIN section, press the '*Change*' button to add a new PIN:

© 2019 Microsoft Corporation. All rights reserved.

- A prompt will be opened asking for the actual password of the user that will set the new PIN.

- Click *OK* button and in the next window type two times the chosen PIN code.

- The user should have now configured the PIN as an authentication mechanism.

© 2019 Microsoft Corporation. All rights reserved.

- Run as administrator, the script *FIA_UAU.5.1 - Test 1&2-PIN.ps1*. The script behaviour is explained as follows:
  - *FIA_UAU.5.1 - Test 1&2-PIN.ps1*: This script lockouts the current user. Now, a manual login with valid credentials (username and PIN) of the recently created user is needed. Afterwards, the same operation shall be performed, but this time, the login should be done with an invalid PIN. The script source code is as follows:

```
#
# Test FIA_UAU.5.1 - PIN
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Logon" /success:enable /failure:enable
wevtutil cl security

Write-Host "[I]: In order to perform this test, please log in as'userFIA' using
    an valid PIN"
Write-Host "[I]: Please, read the instructions carefully"
pause

logoff
Write-Host "[I]: Getting audit log entries..."
pause
Get-EventLog Security -InstanceId 4624 -Newest 1 | Format-List *

Write-Host "[I]: In order to perform this test, please log in as'userFIA' using
    an invalid PIN,
    afterwards introduce a valid PIN"
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Please, read the instructions carefully"
pause

logoff
Write-Host "[I]: Getting audit log entries..."
pause
Get-EventLog Security -InstanceId 4625 -Newest 1 | Format-List *

Write-Host "[I]: Cleaning up audit rules..."
auditpol /set /subcategory:"Logon" /success:disable /failure:disable
```

- Observe the results

28.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

After setting the sign-in option to username and PIN. The evaluator has attempted to login using a known correct user name and PIN. The login was successful and the next image shows the generated audit event:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID             : 4624
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 20091
Category            : (12544)
CategoryNumber      : 12544
EntryType           : SuccessAudit
Message             : An account was successfully logged on.

                      Subject:
                         Security ID:            S-1-5-18
                         Account Name:           SURFACEBOOK2$
                         Account Domain:         WORKGROUP
                         Logon ID:               0x3e7

                      Logon Information:
                         Logon Type:             2
                         Restricted Admin Mode:  -
                         Virtual Account:                %%1843
                         Elevated Token:         %%1842

                      Impersonation Level:               %%1833

                      New Logon:
                         Security ID:            S-1-5-21-1598544264-1886666874-1118049692-500
                         Account Name:           Administrator
                         Account Domain:         SURFACEBOOK2
                         Logon ID:               0x4efebb
                         Linked Logon ID:                0x0
                         Network Account Name:   -
                         Network Account Domain: -
                         Logon GUID:             {00000000-0000-0000-0000-000000000000}

                      Process Information:
                         Process ID:             0x97c
                         Process Name:           C:\Windows\System32\svchost.exe

                      Network Information:
                         Workstation Name:       SURFACEBOOK2
                         Source Network Address: 127.0.0.1
                         Source Port:            0

                      Detailed Authentication Information:
                         Logon Process:          User32
                         Authentication Package: Negotiate
                         Transited Services:     -
                         Package Name (NTLM only):       -
                         Key Length:             0

                      This event is generated when a logon session is created. It is generated on the computer that was
                      accessed.

                      The subject fields indicate the account on the local system which requested the logon. This is
                      most commonly a service such as the Server service, or a local process such as Winlogon.exe or
                      Services.exe.

                      The logon type field indicates the kind of logon that occurred. The most common types are 2
                      (interactive) and 3 (network).

                      The New Logon fields indicate the account for whom the new logon was created, i.e. the account
                      that was logged on.

                      The network fields indicate where a remote logon request originated. Workstation name is not
                      always available and may be left blank in some cases.

                      The impersonation level field indicates the extent to which a process in the logon session can
                      impersonate.

                      The authentication information fields provide detailed information about this specific logon
                      request.
                         - Logon GUID is a unique identifier that can be used to correlate this event with a KDC event.
                         - Transited services indicate which intermediate services have participated in this logon
                      request.
                         - Package name indicates which sub-protocol was used among the NTLM protocols.
                         - Key length indicates the length of the generated session key. This will be 0 if no session key
                      was requested.
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {S-1-5-18, SURFACEBOOK2$, WORKGROUP, 0x3e7...}
InstanceId          : 4624
TimeGenerated       : 12/5/2018 10:59:42 AM
TimeWritten         : 12/5/2018 10:59:42 AM
UserName            :
Site                :
Container           :
```

After that, the evaluator has performed an authentication attempt using a correct user name but an invalid PIN. The login was unsuccessful as can be observed in the following audit event image:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID            : 4625
MachineName        : SURFACEBOOK2
Data               : {}
Index              : 20097
Category           : (12544)
CategoryNumber     : 12544
EntryType          : FailureAudit
Message            : An account failed to log on.

                     Subject:
                        Security ID:            S-1-5-18
                        Account Name:           SURFACEBOOK2$
                        Account Domain:         WORKGROUP
                        Logon ID:               0x3e7

                     Logon Type:                      2

                     Account For Which Logon Failed:
                        Security ID:            S-1-0-0
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2

                     Failure Information:
                        Failure Reason:         %%2313
                        Status:                 0xc000006d
                        Sub Status:             0xc000006a

                     Process Information:
                        Caller Process ID:      0x97c
                        Caller Process Name:    C:\Windows\System32\svchost.exe

                     Network Information:
                        Workstation Name:       SURFACEBOOK2
                        Source Network Address: 127.0.0.1
                        Source Port:            0

                     Detailed Authentication Information:
                        Logon Process:          User32
                        Authentication Package: Negotiate
                        Transited Services:     -
                        Package Name (NTLM only):       -
                        Key Length:             0

                     This event is generated when a logon request fails. It is generated on the computer where access was attempted.

                     The Subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server
                     service, or a local process such as Winlogon.exe or Services.exe.

                     The Logon Type field indicates the kind of logon that was requested. The most common types are 2 (interactive) and 3 (network).

                     The Process Information fields indicate which account and process on the system requested the logon.

                     The Network Information fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank
                     in some cases.

                     The authentication information fields provide detailed information about this specific logon request.
                        - Transited services indicate which intermediate services have participated in this logon request.
                        - Package name indicates which sub-protocol was used among the NTLM protocols.
                        - Key length indicates the length of the generated session key. This will be 0 if no session key was requested.
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-18, SURFACEBOOK2$, WORKGROUP, 0x3e7...}
InstanceId         : 4625
TimeGenerated      : 12/5/2018 10:59:58 AM
TimeWritten        : 12/5/2018 10:59:58 AM
UserName           :
Site               :
Container          :
```

28.2.2.4 Verdict

The evaluator has performed an authentication attempt using the user name and the correct PIN and it has been completed successfully. Afterwards, the evaluator has performed other authentication attempt, but this time using an invalid PIN. The evaluator has ensured that the authentication attempt is invalid.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfillment of the **Test 1 and Test 2 - Username and a PIN (TPM)** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1 and Test 2 - Username and a PIN (TPM)**.

28.2.3 Test 1 & Test 2 - Username and a PIN that releases an asymmetric key stored in OE-protected storage (Smart Card)

28.2.3.1 Setup

To do this test, the evaluator shall have set a Public Key Infrastructure as well as access to the PIV/CAC smart cards.

Due to this fact, this test shall be performed at Microsoft facilities.

© 2019 Microsoft Corporation. All rights reserved.

To perform this test, the evaluated platforms shall be joined to the domain controller. Due to this feature is not included in Windows 10 Home, this test is not applicable for the following evaluated platforms:

- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).

The setup for these tests is described in *Configuring Domain Controller and Certificate Authority (13 Apr 2017)* document.

Before the test execution, the following setup condition must be fulfilled:

- The evaluated platform must be joined to a domain. The steps required to join a computer to a domain are described in Test 13 for *FMT_MOF_EXT.1* requirement.

Once the evaluated platform has been joined to the domain controller, the following steps must be performed according to the authentication method used (physical smart card or virtual smart card).

28.2.3.1.1 Authentication using a physical smart card

The following steps must be carried out by the evaluator in order to configure the authentication using physical smart cards:

- Login the user account that will be used and connect the smart card reader to the computer. To check that the drivers are correctly installed go to *Device Manager* (*WIN + X, then press m key*). The information about the smart card reader and smart card shall be as follows:

© 2019 Microsoft Corporation. All rights reserved.

> **Note:** Physical Smart card issues may arise related to the drivers for the smart card or the smart card itself.
>
> If this is the case, go to the device vendor website or the *Microsoft Update Catalog* to download ans install the latest drivers.
>
> For example, for a *Gemalto IdPrime* card, the drivers can be found in the *Microsoft Update Catalog* using the device manufacture (*Gemalto*) as the query parameter.;
>
> Microsoft Update Catalog for Gemalto
>
> The files downloaded with .cab extension must be extracted and used to install the drivers for *'Unknown smart card'* in Computer Managment/Device Manager. After that, the Smart card shall be immediately available for log in on TOE.

28.2.3.1.2 Authentication using a virtual smart card

The following steps must be carried out by the evaluator to configure the authentication using virtual smart cards:

- Open a Command Line terminal with administrator rights and type the following command to create a virtual smart card. This command only works in platforms with a valid TPM. The default PIN for the virtual smart card is *123456768*.

```
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\evaluador>tpmvscmgr.exe create /name tpmsvc /pin default /adminkey random /generate
Using default PIN: 12345678
Creating TPM Smart Card...
Initializing the Virtual Smart Card component...
Creating the Virtual Smart Card component...
Initializing the Virtual Smart Card Simulator...
Creating the Virtual Smart Card Simulator...
Initializing the Virtual Smart Card Reader...
Creating the Virtual Smart Card Reader...
Waiting for TPM Smart Card Device...
Authenticating to the TPM Smart Card...
Generating filesystem on the TPM Smart Card...
TPM Smart Card created.
Smart Card Reader Device Instance ID = ROOT\SMARTCARDREADER\0000
```

- Enroll the certificate to the virtual smart card with the help of the *Certificate Manager* tool (*WIN* + *R* and then type *certmgr.msc*).
- Then right-click over *Personal*, and select *All Task -> Request New Certificate*.

© 2019 Microsoft Corporation. All rights reserved.

- Follow the wizard, clicking the *'Next'* button twice, on the *'Request Certificates'* screen select *'TPM Virtual Smart Card Logon'* and enter the PIN when it is asked.

© 2019 Microsoft Corporation. All rights reserved.

- If the process has been completed successfully, the following screen shall be shown.

© 2019 Microsoft Corporation. All rights reserved.

- To check that the virtual smart card ready is correctly deployed, go to the *Device Manager* tool (*WIN + X, then press m key*).. The information about the virtual smart card reader shall be as follows:

© 2019 Microsoft Corporation. All rights reserved.

- Sign out to go back the login screen.

28.2.3.2 Procedure

The evaluator shall attempt to authenticate to the operating system using a valid Smart Card PIN. The evaluator shall carry out the following steps, according to the authentication method used (physical smart card or virtual smart card). The steps are defined as follows:

Authentication using a physical smart card

- On the login screen, choose *Other user* option and login using a valid username in the domain controller. Click in sign-in option and select the smart card option. After that, enter the correct smart card PIN. The user shall login correctly.

Authentication using a virtual smart card

© 2019 Microsoft Corporation. All rights reserved.

- On the login screen, choose *Other user* option and login using a valid username in the domain controller. Click in sign-in option and select the security device option. After that, enter the correct virtual smart card PIN. The user shall login correctly

28.2.3.3 Results

The evaluator has performed these tests on the following platforms:

- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

After setting the sign-in option to smart card, the evaluator has attempted to login using the smart card credentials (*User name and Smart Card PIN*). The login was successful and the next image shows the generated audit event:

Then, the evaluator has checked the audit event to verify that the smart card holder verification status was successful during the authentication process:

© 2019 Microsoft Corporation. All rights reserved.

Moreover, the evaluator has ensured that the Smart Card certificate has been involved in the login process. First of all, the evaluator has extracted the certificate information from the smart card with the following command:

> certutil -scinfo

Obtaining the following certificate issuer:

```
CertContext[0][0]: dwInfoStatus=102 dwErrorStatus=0
  Issuer: CN=Physical Smart Card - MSIT3
  NotBefore: 10/29/2018 8:25 AM
  NotAfter: 10/29/2019 8:25 AM
  Subject: CN=[redacted], OU=UserAccounts, DC=redmond, DC=corp, DC=microsoft, DC=com
  Serial: 50000088402b66c3493230c038000000008840
  SubjectAltName: Other Name:Principal Name=[redacted]
  Template: 1.3.6.1.4.1.311.21.8.16155509.8105089.5391003.2969441.12400096.221.9680820.15042049
  Cert: 32c6d8e9aec65e2c963b96cbb6165c40a97b15b7
  Element.dwInfoStatus = CERT_TRUST_HAS_KEY_MATCH_ISSUER (0x2)
  Element.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
    CRL 015b:
    Issuer: CN=Physical Smart Card - MSIT3
    ThisUpdate: 12/10/2018 11:03 AM
    NextUpdate: 12/12/2018 11:23 AM
    CRL: a2bba60527adfad8ca0a029b10b4b6e67268a476
  Issuance[0] = 1.3.6.1.4.1.311.42.1.6 Msit-smartcard-logon
  Issuance[1] = 1.3.6.1.4.1.311.42.1.24
  Application[0] = 1.3.6.1.5.5.7.3.2 Client Authentication
  Application[1] = 1.3.6.1.4.1.311.20.2.2 Smart Card Logon
  Application[2] = 1.3.6.1.4.1.311.42.2.1 ITG Smartcard
```

© 2019 Microsoft Corporation. All rights reserved.

As shown in the following image, this certificate issuer is the same as the certificate installed on the local machine.

In the *Certification Path* tab, it can be seen that the smart card certificate depends on a root certificate:

© 2019 Microsoft Corporation. All rights reserved.

This root certificate was properly installed when the computer was joined to the domain. Due to this, the authentication can be performed and the verification will be successful.

© 2019 Microsoft Corporation. All rights reserved.

In addition, the evaluator has performed an authentication attempt using an incorrect smart card credential (a valid user name but an invalid PIN). The login was unsuccessful as can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

Then, the evaluator has checked the audit event to verify that the smart card holder verification status was unsuccessful during the authentication process:

© 2019 Microsoft Corporation. All rights reserved.

Finally, the same results were obtained when virtual smart cards were used.

28.2.3.4 Verdict

The evaluator has performed an authentication attempt using the correct smart card credentials (*User name and PIN*) and it has been completed successfully. Afterwards, the evaluator has performed other authentication attempt, but this time using an incorrect smart card credential (*Invalid PIN*). The evaluator has ensured that the authentication attempt is invalid.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfillment of the **Test 1 and Test 2 - Username and a PIN (Smart Card)** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1 and Test 2 - Username and a PIN (Smart Card)**.

28.2.4 Test 1 & Test 2 - X.509 Certificates

28.2.4.1 Setup

These tests are not applicable in this evaluation due to the assignment made for the SFR definition in the final version of the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

28.2.4.2 Procedure

These tests are not applicable in this evaluation due to the assignment made for the SFR definition in the final version of the ***Security Target*** document.

28.2.4.3 Results

These tests are not applicable in this evaluation due to the assignment made for the SFR definition in the final version of the ***Security Target*** document.

28.2.4.4 Verdict

These tests are not applicable in this evaluation due to the assignment made for the SFR definition in the final version of the ***Security Target*** document.

28.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FIA_UAU.5.1.

© 2019 Microsoft Corporation. All rights reserved.

29 FIA_UAU.5.2

The assurance activity for the **FIA_UAU.5.2** requirement is stated as follows:

> The evaluator will ensure that the TSS describes each mechanism provided to support user authentication and the rules describing how the authentication mechanism(s) provide authentication.
>
> The evaluator will verify that configuration guidance for each authentication mechanism is addressed in the AGD guidance.
>
> - **Test 1:** For each authentication mechanism selected, the evaluator will enable that mechanism and verify that it can be used to authenticate the user at the specified authentication factor interfaces.
> - **Test 2:** For each authentication mechanism rule, the evaluator will ensure that the authentication mechanism(s) behave as documented in the TSS.

29.1 Documentation Review activity

29.1.1 Findings

The evaluator has reviewed the ***Security Target*** document, section **6.4 Identification and Authentication**, which contains the following information:

> *Password-based authentication to Windows succeeds when the credential provided by the user matches the stored protected representation of the password; Windows Hello and smart cards both use PIN-based authentication to unlock a protected resource, a private key, the stored representation of the PIN is protected by the Secure Kernel.*
>
> *Password authentication can be used for interactive, service, and network logons and to initiate the "change password" screen; the Windows Hello PIN, physical and virtual smart cards can be used for interactive logons; and the Windows Hello PIN is used to re-authenticate the user when the user chooses to change their PIN.*
>
> *When the authentication succeeds, the user will be logged onto their desktop, their screen unlocked, or their authentication factors changed depending whether the user logged onto the computer, the display was locked, or the PIN or password was to be changed.*

As it can be observed, the provided information includes a description about all the authentication mechanisms supported by the TOE. This section also contains a brief description about the rules describing how the authentication mechanisms work as well as the results of a successful authentication.

Moreover, the ***Operational Guidance*** document, includes the following sections, in which the configuration guidance is addressed for each authentication mechanisms supported by the TOE:

- Section 6.7. Managing passwords and password policy

© 2019 Microsoft Corporation. All rights reserved.

- Section 6.8. Managing smart card logon
- Section 6.9. Managing Windows Hello

29.1.2 Verdict

The evaluator has reviewed the ***Security Target*** document and has reviewed the information provided within section **6.4 Identification and Authentication**. This section contains enough information to allow the evaluator to understand how the authentication mechanisms work, as well as, the results of a successful authentication. Moreover, the ***Operational Guidance*** document, also includes configuration guidance for each authentication mechanisms supported by the TOE.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfillment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

29.2 Test Activity

The evaluator will ensure that each authentication mechanism supported by the OS authenticates the user and behaves as described in the AGD guidance.

29.2.1 Test 1

29.2.1.1 Setup

The applicable setup for this test is covered in the Test Activity section for *FIA_UAU.5.1* requirement.

29.2.1.2 Procedure

This test is done at the same time as *FIA_UAU.5.1*.

29.2.1.3 Results

The results for this test are included in the Test Activity section for *FIA_UAU.5.1* requirement.

29.2.1.4 Verdict

The verdict for this test is assigned in the Test Activity section for *FIA_UAU.5.1* requirement.

29.2.2 Test 2

29.2.2.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

Before the test execution, the following setup condition must be fulfilled:

- A standard user account with user name *userTest* shall exist. This account shall belong to default *Users* group.

29.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

Each authentication mechanism supported by the OS has been already tested in the FIA_UAU_5.1 requirement. To verify that the user can modify its password as described in the TSS, the following procedure shall be performed:

- Open the *Settings* application.

- After that, select *Accounts* option and then *Sign-in options* on the left menu:

© 2019 Microsoft Corporation. All rights reserved.

- To change the current password, click on the *Change* button in the Password section. A new prompt asking for the previous password will be shown:

- After typing the current password, the next window will ask for the new password credentials that will be set.

© 2019 Microsoft Corporation. All rights reserved.

- Click on *Next* and *Finish* in the last window. The password will be changed.

- This password change will be applied the next time the user signs in.

© 2019 Microsoft Corporation. All rights reserved.

If the user has set a Windows Hello PIN as an authentication method (*Only available for some of the platforms*), the following steps shall be performed:

- Open the *Settings* application.

- After that, select *Accounts* option and then *Sign-in options* in the left menu:

© 2019 Microsoft Corporation. All rights reserved.

- To change the current Windows Hello PIN, click on the *Change* button in the PIN section. A new prompt asking for the previous PIN will be shown:

- After typing the current PIN, it will asks for the new PIN credentials that will be set.

© 2019 Microsoft Corporation. All rights reserved.

- Click the *OK* button. The PIN will be changed.
- This change will be applied the next time the user sign in.

The evaluator has also verified that the session can be locked (*WIN* + *L*). After locking the current session, it is also possible to unlock it by typing the current password or PIN.

29.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has verified that each authentication attempt works as described in its respective TSS section. For each method, the operation guidance includes information about how to manage each of this methods. As the previous section states, all the authentication mechanisms can be properly managed as described in the documentation.

29.2.2.4 Verdict

The evaluator has managed each authentication method supported. All of them have been properly configured as described in the ***Operational Guidance*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfillment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1.**

29.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FIA_UAU.5.2 requirement.

© 2019 Microsoft Corporation. All rights reserved.

30 FIA_X509_EXT.1.1

The assurance activity for the **FIA_X509_EXT.1.1** requirement is stated as follows:

> The evaluator will ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.
>
> The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA.
>
> **Test 1**: The evaluator will demonstrate that validating a certificate without a valid certification path results in the function failing. The evaluator will then load a certificate or certificates as trusted CAs needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.
>
> **Test 2**: The evaluator will demonstrate that validating an expired certificate results in the function failing.
>
> **Test 3**: The evaluator will test that the OS can properly handle revoked certificates—conditional on whether CRL, OCSP, or OCSP stapling is selected; if multiple methods are selected, then a test shall be performed for each method. The evaluator will test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). The evaluator will ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.
>
> **Test 4**: If either OCSP option is selected, the evaluator will configure the OCSP server or use a man-in-the- middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator will configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.
>
> **Test 5**: The evaluator will modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate should fail to parse correctly.)
>
> **Test 6**: The evaluator will modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate should not validate.)

© 2019 Microsoft Corporation. All rights reserved.

> **Test 7**: The evaluator will modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate should not validate.)

30.1 Documentation Review activity

30.1.1 Findings

The evaluator has reviewed the section **6.4.1 X.509 Certificate Validation and Generation** of the ***Security Target***, where it is stated that all components in the system use the same subcomponent for performing certificate validation. More details are given on the provided URL, where the API used to check the validity of the X509 certificates is described. Concretely, this task is performed by Crypt32.lib and Wincrypt.h.

The TSS also provides information about the certification path algorithm which, according to this document, follows RFC 5280 section **6.2.4 Certificate validation**.

30.1.2 Verdict

The evaluator considers that the evidences defined above and obtained during the documentation review demonstrate the fulfillment of the requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the documentation review activity.

30.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
    a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
    b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
        i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
        ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

© 2019 Microsoft Corporation. All rights reserved.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

30.2.1 Test 1

30.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es

The listed certificates form a valid certification path. All certificates are available in pem format.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

30.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 7 when the menu is prompt.

© 2019 Microsoft Corporation. All rights reserved.

- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

30.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

It is worth noting that, during the course of this test, the server only sends the server certificate (CN = test.epoche.es) along with the handshake, as it can be appreciated in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	60538 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000022	10.10.20.30	10.10.20.25	TCP	66	443 → 60538 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000025	10.10.20.25	10.10.20.30	TCP	60	60538 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000027	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
5	0.000030	10.10.20.30	10.10.20.25	TCP	60	443 → 60538 [ACK] Seq=1 Ack=147 Win=30336 Len=0
6	0.000031	10.10.20.30	10.10.20.25	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
7	0.000037	10.10.20.25	10.10.20.30	TCP	60	60538 → 443 [ACK] Seq=147 Ack=1642 Win=262656 Len=0
8	0.000039	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000041	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000043	10.10.20.25	10.10.20.30	TCP	60	60538 → 443 [ACK] Seq=489 Ack=1717 Win=262656 Len=0
11	0.000044	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
12	0.000046	10.10.20.30	10.10.20.25	TCP	60	443 → 60538 [FIN, ACK] Seq=1770 Ack=489 Win=31360 Len=0
13	0.000047	10.10.20.25	10.10.20.30	TCP	60	60538 → 443 [ACK] Seq=489 Ack=1771 Win=262656 Len=0

```
▸ Frame 6: 1695 bytes on wire (13560 bits), 1695 bytes captured (13560 bits)
▸ Ethernet II, Src: PcsCompu_f0:2a:11 (08:00:27:f0:2a:11), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.25
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60538, Seq: 1, Ack: 147, Len: 1641
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1569
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1565
        Certificates Length: 1562
      ▾ Certificates (1562 bytes)
          Certificate Length: 1559
        ▾ Certificate: 30820613308203fba003020102020210003 00d06092a8648... (id-at-commonName=test.epoche.es,id-at-organizationalUnitName=ee,id-at-organizati
          ▸ signedCertificate
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: 18521da4750a5625ca33a9ba668b4cfa96133b71b2115b64...
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

For the first part of this test, the certificate validation failure when the full certificate chain is missing has been verified from the tool's output, which prints details about the error found in the connection process, as shown in the following screenshot:

```
[+] Performing FIA_X509_EXT.1.1 Test1 (a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate chain could not be built to a trusted root authority.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

© 2019 Microsoft Corporation. All rights reserved.

This results on the client denying the connection, as it can be seen in the following screenshot from the *Wireshark* capture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.77	10.10.20.30	TCP	66	49788 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 S
2	0.000022	10.10.20.30	10.10.20.77	TCP	66	443 → 49788 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=14
3	0.000024	10.10.20.77	10.10.20.30	TCP	60	49788 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000025	10.10.20.77	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000028	10.10.20.30	10.10.20.77	TCP	60	443 → 49788 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000030	10.10.20.30	10.10.20.77	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
7	0.000035	10.10.20.77	10.10.20.30	TCP	60	49788 → 443 [ACK] Seq=149 Ack=1642 Win=262656 Len=0
8	0.000037	10.10.20.77	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handsh
9	0.000038	10.10.20.30	10.10.20.77	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000040	10.10.20.77	10.10.20.30	TCP	60	49788 → 443 [FIN, ACK] Seq=491 Ack=1717 Win=262656 Len=0
11	0.000041	10.10.20.30	10.10.20.77	TLSv1.2	107	Encrypted Alert
12	0.000042	10.10.20.30	10.10.20.77	TCP	60	443 → 49788 [FIN, ACK] Seq=1770 Ack=492 Win=31360 Len=0
13	0.000043	10.10.20.77	10.10.20.30	TCP	60	49788 → 443 [RST, ACK] Seq=492 Ack=1770 Win=0 Len=0

The client machine sends a FIN, ACK once the validation fails, and responds with RST, ACK to the application data coming from the server.

For the second part of this test, the certificate validation success when there is a valid certification path in the system has been verified in the same way, and the evidence is shown next:

```
[+] Performing FIA_X509_EXT.1.1 Test1 (b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000020	10.10.20.25	10.10.20.30	TCP	66	60539 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7	0.000022	10.10.20.30	10.10.20.25	TCP	66	443 → 60539 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
8	0.000024	10.10.20.25	10.10.20.30	TCP	60	60539 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
9	0.000026	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
10	0.000028	10.10.20.30	10.10.20.25	TCP	60	443 → 60539 [ACK] Seq=1 Ack=147 Win=30336 Len=0
11	0.000029	10.10.20.30	10.10.20.25	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
12	0.000035	10.10.20.25	10.10.20.30	TCP	60	60539 → 443 [ACK] Seq=147 Ack=1642 Win=262656 Len=0
13	0.000036	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14	0.000039	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
15	0.000040	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
16	0.000042	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data
17	0.000082	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
18	0.000084	10.10.20.25	10.10.20.30	TCP	60	60539 → 443 [ACK] Seq=558 Ack=2032 Win=262400 Len=0
19	0.000086	10.10.20.25	10.10.20.30	TCP	60	60539 → 443 [FIN, ACK] Seq=558 Ack=2032 Win=262400 Len=0
20	0.000087	10.10.20.30	10.10.20.25	TCP	60	443 → 60539 [ACK] Seq=2032 Ack=559 Win=31360 Len=0

This time, as it can be appreciated, the validation is successful and both client and server send application data to each other.

© 2019 Microsoft Corporation. All rights reserved.

Equivalent result is obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

For the third part of this test, the Intermediate CA 1 certificate is removed from the chain, and the certificate validation failure is verified in the same way. The following evidence is obtained using the command line tool:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.1 Test1 (c)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate chain could not be built to a trusted root authority.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

30.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements es-

© 2019 Microsoft Corporation. All rights reserved.

tablished in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

30.2.2 Test 2

30.2.2.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (expired certificate)

The listed certificates form a valid certification path. All certificates are available in pem format.

30.2.2.2 Procedure

The procedure is identical to the one defined for Test 1.

30.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

During the course of this test, the server only sends the expired server certificate (CN = test.epoche.es) along with the handshake. The fact that it is expired can be checked in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

The certificate validation failure due to the lack of a valid certification path has been verified from the tool's output, which prints details about the error found in the connection process, as shown in the following screenshot:

```
[+] Performing FIA_X509_EXT.1.1 Test2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A required certificate is not within its validity period when verifying against the current system clock or the timestamp in the signed file.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

This results on the client denying the connection, as it can be seen in the following screenshot from the *Wireshark* capture:

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000005	10.10.20.25	10.10.20.30	TCP	66	60541 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	0.000006	10.10.20.30	10.10.20.25	TCP	66	443 → 60541 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	0.000006	10.10.20.25	10.10.20.30	TCP	60	60541 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
5	0.000007	10.10.20.25	10.10.20.30	TLSv1.2	200	Client Hello
6	0.000008	10.10.20.30	10.10.20.25	TCP	60	443 → 60541 [ACK] Seq=1 Ack=147 Win=30336 Len=0
7	0.000008	10.10.20.30	10.10.20.25	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
8	0.000010	10.10.20.25	10.10.20.30	TCP	60	60541 → 443 [ACK] Seq=147 Ack=1642 Win=262656 Len=0
9	0.000011	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.000011	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
11	0.000012	10.10.20.25	10.10.20.30	TCP	60	60541 → 443 [FIN, ACK] Seq=489 Ack=1717 Win=262656 Len=0
12	0.000012	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
13	0.000013	10.10.20.30	10.10.20.25	TCP	60	443 → 60541 [FIN, ACK] Seq=1770 Ack=490 Win=31360 Len=0
14	0.000013	10.10.20.25	10.10.20.30	TCP	60	60541 → 443 [RST, ACK] Seq=490 Ack=1770 Win=0 Len=0

The client machine sends a FIN, ACK once the validation fails, and responds with RST, ACK to the application data coming from the server. Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

30.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

30.2.3 Test 3

30.2.3.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1 (non-revoked)
- CN = EE Test Intermediate CA 1 (revoked)
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es (non-revoked)
- CN = test.epoche.es (revoked)

The certificates listed form a valid certification path. All certificates are available in pem format as well as pfx.

Additionally the following OCSP signing certificates are added:

- CN = ocsp.test.epoche.es (signed by the root CA)
- CN = ocsp.test.epoche.es (signed by intermediate CA 1)

© 2019 Microsoft Corporation. All rights reserved.

- CN = ocsp.test.epoche.es (signed by intermediate CA 2)

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine, CRL Server (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

For OCSP and OCSP stapling tests, the following machine is added:

- Web Server, OCSP Server (Windows Server 2016)

These machines are all in the same network with the following configuration:

- Server Machine/CRL Server, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*
- Web/OCSP Server, IP = 10.10.20.40

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS. Furthermore, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

Moreover, the following additional scripts are needed to perform the test case:

- For CRL (on CRL server machine):
  - *1_startCRLServer_norevocation.sh*: This script starts the CRL server in which there is no revoked certificate.
  - *2_startCRLServer_servercert_revocation.sh*: This script starts the CRL server in which the server certificate is revoked.
  - *3_startCRLserver_intermediate1_revocation.sh*: This script starts the CRL server in which a intermediate certificate is revoked.
- For OCSP (on OCSP Server machine):
  - *configureServer.ps1:* Powershell script which shall be executed on the OCSP Server machine. This scripts automatically modifies the revocation configuration depending on the performed test (no revocation, server certificate revoked or intermediate certificate revoked).

Moreover, the Web/OCSP server machine shall have the following Windows Server roles installed:

- Internet Information Services (web server)
- Online Responder (OCSP server)

If they are not installed, the provided script *configureWindowsServerForOCSPandOCSPStapling.ps1* automatically installs and configures the IIS role along with the correct certificates.

© 2019 Microsoft Corporation. All rights reserved.

However, the Online Responder role requires some degree of manual configuration, indicated next.

The Online Responder role shall first be installed as follows:

- After running *configureWindowsServerForOCSPandOCSPStapling.ps1* on the Windows Server machine, open the Server Manager from the task bar.
- From the Server Manager Dashboard select "Manage" and then "Add Roles and Features".
- Click next until the "Select server roles" screen.
- Inside "Active Directory Certificate Services", select "Online Responder".
- Click next until the installation is finished, accepting the suggested extra elements in the process.
- Configure the Online Responder if prompted in the results screen (no input is needed to do so).

Before configuring the Online Responder, the certificate revocation lists (CRLs) need to be made available:

- Copy the contents from "ee-tls-tool/certs/X509_EXT.1.1/ca" (it includes the CRLs) inside "C:\inetpub\wwwroot".
- From the Server Manager Dashboard select "Tools" and then "Internet Information Services (IIS) Manager".
- On the left panel, click on "Default Web Site" below the system's name and "Sites".
- On the central panel, double click on "MIME Types".
- On the right panel, click "Add…".
- Under "File name extension", type ".pem".
- Under "MIME type", type "application/pki-crl" and click "OK".

Read permission for the OCSP private keys needs to be granted to the "NETWORK SERVICE" account as follows:

- Open mmc.exe (e.g. by opening the start menu, typing "mmc" and then pressing enter).
- On the "File" menu, click "Add/Remove Snap-in…".
- In the "Available snap-ins" panel, select "Certificates".
- Press "Add".
- In the "Certificates snap-in" dialog box, select "Computer Account" and click next.
- Select "Local Computer" and press "Finish".
- Press "OK".
- In the console tree, open "Personal" and then "Certificates". The three certificates with CN = ocsp.test.epoche.es will appear inside the presented list.
- Right click on one of them and select "All tasks" and then "Manage Private Keys…".
- In the "Security" tab presented, click "Add…".
- Enter "NETWORK SERVICE" as the object name.
- Check that the "NETWORK SERVICE" group now appears with read access.
- Repeat the process with the two remaining OCSP signing certificates.

The Online Responder service can now be configured as shown next:

© 2019 Microsoft Corporation. All rights reserved.

- From the Server Manager Dashboard select "Tools" and then "Online Responder Management".
- On the left panel, right click on "Revocation Configuration" and then click on "Add Revocation Configuration".
- Click "Next" on the first screen.
- On the "Name the Revocation Configuration" screen, type "EE Root CA" (could be any name) and click "Next".
- On the "Select CA Certificate Location" screen, select "Select a Certificate from the Local certificate store" and click "Next".
- On the "Choose CA Certificate" screen, click "Browse..." and select the root CA (CN = EE Test Root CA). Then click "Next".
- On the "Select Signing Certificate" screen, select "Manually select a signing certificate" and click "Next". An error might appear; it can be ignored.
- On the "Revocation Provider" screen, click "Provider...".
- Below the "Base CRLs" list, click "Add...".
- Type "http://localhost/crl/ca_before_revocation.crl.pem" and click "OK".
- Repeat from step 2, for the intermediate certificate 1. Type "EE Intermediate CA 1" instead, select "EE Test Intermediate CA 1" and type "http://localhost/intermediate1/crl/intermediate1.crl.pem".
- Repeat from step 2, for the intermediate certificate 2. Type "EE Intermediate CA 2" instead, select "EE Test Intermediate CA 2" and type "http://localhost/intermediate1/intermediate2/crl/intermediate2_before_revocation.crl.pem".
- On the left panel, select the system under "Array configuration".
- On the centre panel, for every one of the three configurations created, click them and then select "Assign Signing Certificate" on the right panel, selecting the corresponding OCSP signing certificate (e.g. for configuration "EE Intermediate CA 2" select the one with CN = ocsp.test.epoche.es that was signed by "EE Test Intermediate CA 2").
- On the left panel, right click on "Array configuration" and then "Refresh Revocation Data".
- Click on the top-most element in the left panel (starting with "Online Responder:").
- The centre panel should show the status of the three configurations added, and they all should be shown as "Working".

Both on the client machine and the Windows Server machine, "C:\Windows\System32\drivers\etc\hosts" shall be modified so that it contains a line "10.10.20.30 test.epoche.es" and another with "10.10.20.40 ocsp.test.epoche.es". Finally, reboot the Windows Server machine for OCSP stapling to be enabled.

30.2.3.2 Procedure

The following steps shall be performed to complete this test assurance activity for revocation via CRL:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.

© 2019 Microsoft Corporation. All rights reserved.

- On the same machine, run the appropriate script to start the CRL server (*1_startCRLServer_norevocation.sh, 2_startCRLServer_servercert_revocation.sh or 3_startCRLserver_intermediate1_revocation.sh* depending on the test case)
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 7 when the menu is prompt.
- Once the TLS server is waiting for connection and the CRL server is properly configured, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

Next, revocation via OCSP shall be tested as follows:

- Stop the CRL server from the previous tests.
- On the OCSP server machine, run the script *configureServer.ps1* to configure the OCSP server depending on the test case will be executed. The script will provide some instructions that should be followed on the OCSP server machine in order to ensure that the revocation list is properly refreshed.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 7 when the menu is prompt.
- Once the TLS server is waiting for connection and the OSCP server is properly configured, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

Next, revocation via OCSP stapling shall be tested as follows:

- Shut down the "ee-tls-tool" since the TLS server is the Web Server running on the Windows Server machine this time.
- Change hosts file accordingly in the client machine and the Windows Server machine, so that it contains "10.10.20.40 test.epoche.es" and with "10.10.20.40 ocsp.test.epoche.es".
- Change hosts file in the Windows Server machine so that it contains 127.0.0.1 ocsp.test.epoche.es-
- Execute the following powershell command to clear the revocation cache: certutil ——urlcache * delete
- Ensuring that both the web server and the Online Responder at the Windows Server machine are running,.
- On the client machine, open a web browser (Edge or Internet Explorer) and navigate to "https://test.epoche.es". This is done since they both support OCSP stapling, unlike *TlsClientTest*.
- Verify that validation is successful both from the browser window and the *Wireshark* capture.
- On the Windows Server Machine, open the "Online Responder Manager" from the Server Manager Dashboard ("Tools" menu). On the left panel, select "Revocation Configuration". Then, on the middle panel, select "EE Intermediate CA 2". Then, on the right panel se-

© 2019 Microsoft Corporation. All rights reserved.

lect "Edit Properties". On the "Revocation Provider" tab, click on "Provider...". Edit the existing base CRL so that it points to "http://localhost/intermediate1/intermediate2/crl/intermediate2_after_revocation". On the left panel, right click on "Array configuration" and then "Refresh Revocation Data". The server certificate should now be revoked.

- Execute the following powershell command to clear the revocation cache: `certutil --urlcache * delete`
- Restart the Windows Server machine to force OCSP status to be checked again from the web server.
- Ensuring that both the web server and the Online Responder at the Windows Server machine are running, open a web browser like Edge or Internet explorer on the client machine and navigate to https://test.epoche.es.
- Verify that validation fails both from the browser window and the *Wireshark* capture.

Note that for OCSP stapling, only test revocation of the node (server) certificate can be tested, and not that of the intermediate 1 certificate, since only revocation status of the server certificate is transmitted by the server during the handshake as stated in RFC 6066.

30.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For every scenario defined in the test assurance activity, the correct behaviour of certificate revocation checks is verified. To do so, the certificates presented by the server, generated by the scripts, will have the *cRLDistributionPoints* and *authorityInfoAccessSyntax* extensions indicating the URIs where the certificate status can be checked by using CRL or OCSP respectively. A picture from a network capture (server's "Certificate" message during the handshake) where this can be appreciated is shown next.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
15	4.882562393	10.10.10.52	10.10.20.30	TLSv1.2	229	Client Hello
17	4.884413468	10.10.20.30	10.10.10.52	TLSv1.2	1514	Server Hello
19	4.884970724	10.10.20.30	10.10.10.52	TLSv1.2	235	Certificate, Server Hello Done
23	4.882588733	10.10.20.25	10.10.20.30	TLSv1.2	229	Client Hello
25	4.883588311	10.10.20.30	10.10.20.25	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
127	28.151797342	10.10.10.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
128	28.152087684	10.10.10.52	10.10.20.30	TLSv1.2	475	Application Data
130	28.153626713	10.10.20.30	10.10.10.52	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
132	28.154075246	10.10.20.30	10.10.10.52	TLSv1.2	368	Application Data, Encrypted Alert
157	28.151817258	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
158	28.152096602	10.10.20.25	10.10.20.30	TLSv1.2	475	Application Data
160	28.153423838	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
161	28.153862680	10.10.20.30	10.10.20.25	TLSv1.2	368	Application Data, Encrypted Alert

```
Extension (id-ce-authorityKeyIdentifier)
Extension (id-ce-keyUsage)
Extension (id-ce-extKeyUsage)
Extension (id-ce-cRLDistributionPoints)
    Extension Id: 2.5.29.31 (id-ce-cRLDistributionPoints)
  CRLDistPointsSyntax: 1 item
    DistributionPoint
      distributionPoint: fullName (0)
        fullName: 1 item
          GeneralName: uniformResourceIdentifier (6)
            uniformResourceIdentifier: http://crl.epoche.es:8000/intermediate1/intermediate2/crl/intermediate2.crl.pem
Extension (id-pe-authorityInfoAccessSyntax)
    Extension Id: 1.3.6.1.5.5.7.1.1 (id-pe-authorityInfoAccessSyntax)
  AuthorityInfoAccessSyntax: 1 item
    AccessDescription
      accessMethod: 1.3.6.1.5.5.7.48.1 (id-pkix.48.1)
      accessLocation: 6
        uniformResourceIdentifier: http://ocsp.test.epoche.es/ocsp
algorithmIdentifier (sha256WithRSAEncryption)
Padding: 0
encrypted: 18521da4750a5625ca33a9ba668b4cfa96133b71b2115b64...
```

```
03e0  07 03 01 30 60 06 03 55  1d 1f 04 59 30 57 30 55   ···0`··U ···Y0W0U
03f0  a0 53 a0 51 86 4f 68 74  74 70 3a 2f 2f 63 72 6c   ·S·Q·Oht tp://crl
0400  2e 65 70 6f 63 68 65 2e  65 73 3a 38 30 30 30 2f   .epoche. es:8000/
0410  69 6e 74 65 72 6d 65 64  69 61 74 65 31 2f 69 6e   intermed iate1/in
0420  74 65 72 6d 65 64 69 61  74 65 32 2f 63 72 6c 2f   termedia te2/crl/
0430  69 6e 74 65 72 6d 65 64  69 61 74 65 32 2e 63 72   intermed iate2.cr
0440  6c 2e 70 65 6d 30 3b 06  08 2b 06 01 05 05 07 01   l.pem0;· ·+·····
0450  01 04 2f 30 2d 30 2b 06  08 2b 06 01 05 05 07 30   ··/0-0+· ·+·····0
```

Note that the shown ones belong to the server certificate. The intermediate certificates present in the client machine also have their own URIs for revocation check, where the CRL files are different (but in our test scenario present on the same server) and the OCSP server URL stays the same (since the Windows Server machine's Online Responder is set up to respond to OCSP requests for all certificates in the chain).

The first scenario in the test assurance activity tests CRL with no revoked certificates. In the following picture from the network traffic the client requests to the CRL server are shown, and both client and server machines send application data to each other after the CRL responses arrive.

No.	Time	Source	Destination	Protocol	Length	Info
98	26.327807724	10.10.20.25	10.10.20.30	TCP	54	49717 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
99	26.327900182	10.10.20.25	10.10.20.30	HTTP	187	GET /crl/ca.crl.pem HTTP/1.1
100	26.328120206	10.10.20.30	10.10.20.25	TCP	60	8000 → 49717 [ACK] Seq=1 Ack=134 Win=30336 Len=0
101	26.328733823	10.10.20.30	10.10.20.25	TCP	71	8000 → 49717 [PSH, ACK] Seq=1 Ack=134 Win=30336 Len=17 [TCP segment of a reassembled PDU]
102	26.329040286	10.10.20.30	10.10.20.25	HTTP	1283	HTTP/1.0 200 OK
103	26.330106039	10.10.20.25	10.10.20.30	TCP	54	49717 → 8000 [ACK] Seq=134 Ack=1248 Win=261376 Len=0
104	26.330281995	10.10.20.25	10.10.20.30	TCP	54	49717 → 8000 [FIN, ACK] Seq=134 Ack=1248 Win=261376 Len=0
105	26.330503475	10.10.20.30	10.10.20.25	TCP	60	8000 → 49717 [ACK] Seq=1248 Ack=135 Win=30336 Len=0
106	28.067829754	10.10.20.40	10.10.10.52	DNS	79	Standard query response 0x2e48 Server failure A ocsp.test.epoche.es
107	28.091124432	10.10.10.52	10.10.20.30	TCP	66	49718 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
108	28.091565900	10.10.20.30	10.10.10.52	TCP	66	8000 → 49718 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
109	28.092210408	10.10.10.52	10.10.20.30	TCP	60	49718 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
110	28.092432815	10.10.10.52	10.10.20.30	HTTP	212	GET /intermediate1/crl/intermediate1.crl.pem HTTP/1.1
111	28.092778265	10.10.20.30	10.10.10.52	TCP	54	8000 → 49718 [ACK] Seq=1 Ack=159 Win=30336 Len=0
112	28.093019895	10.10.20.30	10.10.10.52	TCP	71	8000 → 49718 [PSH, ACK] Seq=1 Ack=159 Win=30336 Len=17 [TCP segment of a reassembled PDU]
113	28.093168117	10.10.20.30	10.10.10.52	HTTP	1295	HTTP/1.0 200 OK
114	28.093682779	10.10.10.52	10.10.20.30	TCP	60	49718 → 8000 [ACK] Seq=159 Ack=1260 Win=261376 Len=0
115	28.093879924	10.10.10.52	10.10.20.30	TCP	60	49718 → 8000 [FIN, ACK] Seq=159 Ack=1260 Win=261376 Len=0
116	28.094152948	10.10.20.30	10.10.10.52	TCP	54	8000 → 49718 [ACK] Seq=1260 Ack=160 Win=30336 Len=0
117	28.130889963	10.10.10.52	10.10.20.30	TCP	66	49719 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
118	28.132035635	10.10.20.30	10.10.10.52	TCP	66	8000 → 49719 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
119	28.132534547	10.10.10.52	10.10.20.30	TCP	60	49719 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
120	28.132738976	10.10.10.52	10.10.20.30	HTTP	226	GET /intermediate1/intermediate2/crl/intermediate2.crl.pem HTTP/1.1
121	28.133675772	10.10.20.30	10.10.10.52	TCP	54	8000 → 49719 [ACK] Seq=1 Ack=173 Win=30336 Len=0
122	28.134496900	10.10.20.30	10.10.10.52	TCP	71	8000 → 49719 [PSH, ACK] Seq=1 Ack=173 Win=30336 Len=17 [TCP segment of a reassembled PDU]
123	28.134949200	10.10.20.30	10.10.10.52	HTTP	1295	HTTP/1.0 200 OK
124	28.135440794	10.10.10.52	10.10.20.30	TCP	60	49719 → 8000 [ACK] Seq=173 Ack=1260 Win=261376 Len=0
125	28.135470900	10.10.10.52	10.10.20.30	TCP	60	49719 → 8000 [FIN, ACK] Seq=173 Ack=1260 Win=261376 Len=0
126	28.136028270	10.10.20.30	10.10.10.52	TCP	54	8000 → 49719 [ACK] Seq=1260 Ack=174 Win=30336 Len=0
127	28.151797342	10.10.10.52	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
128	28.152087684	10.10.10.52	10.10.20.30	TLSv1.2	475	Application Data

The correct connection to the server is also verified from the output from *TlsClientTest*.

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.1 Test3 (CRL a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

The next step, testing a revoked server certificate using CRL, is performed after serving the corresponding CRL file with the revocation information. As it can be appreciated, as soon as the client gets the CRL from the intermediate 2 CA, the connection is cancelled so that no application data is sent from the client.

No.	Time	Source	Destination	Protocol	Length	Info
34	0.000103	10.10.20.25	10.10.20.30	TCP	60	49724 → 443 [ACK] Seq=176 Ack=1642 Win=261888 Len=0
35	0.000104	10.10.20.25	10.10.20.30	TCP	66	49725 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
36	0.000105	10.10.20.30	10.10.20.25	TCP	66	8000 → 49725 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
37	0.000107	10.10.20.25	10.10.20.30	TCP	60	49725 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
38	0.000108	10.10.20.25	10.10.20.30	HTTP	187	GET /crl/ca.crl.pem HTTP/1.1
39	0.000110	10.10.20.30	10.10.20.25	TCP	60	8000 → 49725 [ACK] Seq=1 Ack=134 Win=30336 Len=0
40	0.000111	10.10.20.30	10.10.20.25	TCP	71	8000 → 49725 [PSH, ACK] Seq=1 Ack=134 Win=30336 Len=17 [TCP segment of a reassembled PDU]
41	0.000112	10.10.20.30	10.10.20.25	HTTP	1283	HTTP/1.0 200 OK
42	0.000116	10.10.20.25	10.10.20.30	TCP	60	49725 → 8000 [ACK] Seq=134 Ack=1248 Win=261376 Len=0
43	0.000117	10.10.20.25	10.10.20.30	TCP	60	49725 → 8000 [FIN, ACK] Seq=134 Ack=1248 Win=261376 Len=0
44	0.000118	10.10.20.30	10.10.20.25	TCP	60	8000 → 49725 [ACK] Seq=1248 Ack=135 Win=30336 Len=0
45	0.000119	10.10.20.25	10.10.20.30	TCP	66	49726 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
46	0.000121	10.10.20.30	10.10.20.25	TCP	66	8000 → 49726 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
47	0.000122	10.10.20.25	10.10.20.30	TCP	60	49726 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
48	0.000123	10.10.20.25	10.10.20.30	HTTP	212	GET /intermediate1/crl/intermediate1.crl.pem HTTP/1.1
49	0.000125	10.10.20.30	10.10.20.25	TCP	60	8000 → 49726 [ACK] Seq=1 Ack=159 Win=30336 Len=0
50	0.000126	10.10.20.30	10.10.20.25	TCP	71	8000 → 49726 [PSH, ACK] Seq=1 Ack=159 Win=30336 Len=17 [TCP segment of a reassembled PDU]
51	0.000127	10.10.20.30	10.10.20.25	HTTP	1295	HTTP/1.0 200 OK
52	0.000150	10.10.20.25	10.10.20.30	TCP	60	49726 → 8000 [ACK] Seq=159 Ack=1260 Win=261376 Len=0
53	0.000152	10.10.20.25	10.10.20.30	TCP	60	49726 → 8000 [FIN, ACK] Seq=159 Ack=1260 Win=261376 Len=0
54	0.000153	10.10.20.30	10.10.20.25	TCP	60	8000 → 49726 [ACK] Seq=1260 Ack=160 Win=30336 Len=0
55	0.000154	10.10.20.25	10.10.20.30	TCP	66	49727 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
56	0.000155	10.10.20.30	10.10.20.25	TCP	66	8000 → 49727 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
57	0.000157	10.10.20.25	10.10.20.30	TCP	60	49727 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
58	0.000158	10.10.20.25	10.10.20.30	HTTP	226	GET /intermediate1/intermediate2/crl/intermediate2.crl.pem HTTP/1.1
59	0.000160	10.10.20.30	10.10.20.25	TCP	60	8000 → 49727 [ACK] Seq=1 Ack=173 Win=30336 Len=0
60	0.000161	10.10.20.30	10.10.20.25	TCP	71	8000 → 49727 [PSH, ACK] Seq=1 Ack=173 Win=30336 Len=17 [TCP segment of a reassembled PDU]
61	0.000162	10.10.20.30	10.10.20.25	HTTP	1327	HTTP/1.0 200 OK
62	0.000166	10.10.20.25	10.10.20.30	TCP	60	49727 → 8000 [ACK] Seq=173 Ack=1292 Win=261376 Len=0
63	0.000167	10.10.20.25	10.10.20.30	TCP	60	49727 → 8000 [FIN, ACK] Seq=173 Ack=1292 Win=261376 Len=0
64	0.000168	10.10.20.30	10.10.20.25	TCP	60	8000 → 49727 [ACK] Seq=1292 Ack=174 Win=30336 Len=0

The failure during validation is verified from the output from *TlsClientTest* as well.

```
[+] Performing FIA_X509_EXT.1.1 Test3 (CRL b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is revoked.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

© 2019 Microsoft Corporation. All rights reserved.

Equivalent results are obtained when using a web browser like Edge.

The same is verified for the case where the intermediate 1 certificate is revoked, as shown next:

No.	Time	Source	Destination	Protocol	Length	Info
3369	0.016323	10.10.20.25	10.10.20.30	TCP	66	49728 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
3370	0.016324	10.10.20.30	10.10.20.25	TCP	66	443 → 49728 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3371	0.016326	10.10.20.25	10.10.20.30	TCP	60	49728 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
3372	0.016327	10.10.20.25	10.10.20.30	TLSv1.2	229	Client Hello
3373	0.016329	10.10.20.30	10.10.20.25	TCP	60	443 → 49728 [ACK] Seq=1 Ack=176 Win=30336 Len=0
3374	0.016330	10.10.20.30	10.10.20.25	TLSv1.2	1695	Server Hello, Certificate, Server Hello Done
3375	0.016343	10.10.20.25	10.10.20.30	TCP	60	49728 → 443 [ACK] Seq=176 Ack=1461 Win=262144 Len=0
3376	0.016345	10.10.20.25	10.10.20.30	TCP	60	49728 → 443 [ACK] Seq=176 Ack=1642 Win=261888 Len=0
3381	0.016350	10.10.20.25	10.10.20.30	TCP	66	49729 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3382	0.016351	10.10.20.30	10.10.20.25	TCP	66	8000 → 49729 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3383	0.016353	10.10.20.25	10.10.20.30	TCP	60	49729 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
3384	0.016354	10.10.20.25	10.10.20.30	HTTP	187	GET /crl/ca.crl.pem HTTP/1.1
3385	0.016355	10.10.20.30	10.10.20.25	TCP	60	8000 → 49729 [ACK] Seq=1 Ack=134 Win=30336 Len=0
3386	0.016357	10.10.20.30	10.10.20.25	TCP	71	8000 → 49729 [PSH, ACK] Seq=1 Ack=134 Win=30336 Len=17 [TCP segment of a reassembled PDU]
3387	0.016358	10.10.20.30	10.10.20.25	HTTP	1311	HTTP/1.0 200 OK
3388	0.016361	10.10.20.25	10.10.20.30	TCP	60	49729 → 8000 [ACK] Seq=134 Ack=1276 Win=261376 Len=0
3389	0.016362	10.10.20.25	10.10.20.30	TCP	60	49729 → 8000 [FIN, ACK] Seq=134 Ack=1276 Win=261376 Len=0
3390	0.016363	10.10.20.30	10.10.20.25	TCP	60	8000 → 49729 [ACK] Seq=1276 Ack=135 Win=30336 Len=0
3391	0.016364	10.10.20.25	10.10.20.30	TCP	60	49728 → 443 [FIN, ACK] Seq=176 Ack=1642 Win=261888 Len=0
3392	0.016365	10.10.20.30	10.10.20.25	TCP	60	443 → 49728 [FIN, ACK] Seq=1642 Ack=177 Win=30336 Len=0

And in the output from *TlsClientTest* as well.

```
[+] Performing FIA_X509_EXT.1.1 Test3 (CRL c)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is revoked.

Chain status: The revocation function was unable to check revocation for the certificate.

Chain status: The revocation function was unable to check revocation because the revocation server was offline.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

In the case the connection is refused by the client as soon as it gets the CRL from the root CA.

The next tests are performed with OCSP, so the scenario is changed as explained in the previous section. For the test with no revocations, next picture shows how the client sends OCSP requests to the server, one for every certificate.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.10.20.52	10.10.20.30	TCP	68	49735 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000414011	10.10.20.30	10.10.20.52	TCP	68	443 → 49735 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000837374	10.10.20.52	10.10.20.30	TCP	62	49735 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.010118992	10.10.20.52	10.10.20.30	TLSv1.2	204	Client Hello
5	0.010619693	10.10.20.30	10.10.20.52	TCP	62	443 → 49735 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.011135006	10.10.20.30	10.10.20.52	TLSv1.2	1516	Server Hello
7	0.011234204	10.10.20.30	10.10.20.52	TLSv1.2	237	Certificate, Server Hello Done
8	0.011578138	10.10.20.52	10.10.20.30	TCP	62	49735 → 443 [ACK] Seq=149 Ack=1642 Win=2102272 Len=0
9	0.013839452	10.10.20.52	10.10.20.30	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.019362949	10.10.20.30	10.10.20.52	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
11	0.074728285	10.10.20.52	10.10.20.30	TCP	62	49735 → 443 [ACK] Seq=491 Ack=1717 Win=2102272 Len=0
12	1.034605403	10.10.20.30	10.10.20.52	TLSv1.2	109	Encrypted Alert
13	1.034829589	10.10.20.30	10.10.20.52	TCP	62	443 → 49735 [FIN, ACK] Seq=1770 Ack=491 Win=31360 Len=0
14	1.035271909	10.10.20.52	10.10.20.30	TCP	62	49735 → 443 [ACK] Seq=491 Ack=1771 Win=2102272 Len=0
15	11.792276097	10.10.20.52	10.10.20.40	TCP	68	49736 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
16	11.792711648	10.10.20.40	10.10.20.52	TCP	68	80 → 49736 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
17	11.793124602	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
18	11.793319705	10.10.20.52	10.10.20.40	HTTP	277	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBT2Ccwj0sogAJDmm5nPT2C%2FgPHEtgQUlEOfgCR14SKnwkrl
19	11.795150518	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=1 Ack=222 Win=525568 Len=1460 [TCP segment of a reassembled PDU]
20	11.795164915	10.10.20.40	10.10.20.52	OCSP	1193	Response
21	11.795568668	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=222 Ack=2598 Win=2102272 Len=0
22	11.826238021	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBTgc2kYnVqyR7nclS0mZZyR4cdhigQUBBGr9Zu3TJdXx0KtX
23	11.827978990	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=2598 Ack=441 Win=525312 Len=1460 [TCP segment of a reassembled PDU]
24	11.827993307	10.10.20.40	10.10.20.52	OCSP	1203	Response
25	11.828412396	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=441 Ack=5205 Win=2102272 Len=0
26	11.851830092	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBS0YNFlpGsTx4EbOEIpySbeFyb8XgQUcsnWd4bc9Tqi0A7ia
27	11.853617136	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=5205 Ack=660 Win=525056 Len=1460 [TCP segment of a reassembled PDU]
28	11.853630284	10.10.20.40	10.10.20.52	OCSP	1203	Response
29	11.854058549	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=660 Ack=7812 Win=2102272 Len=0
30	11.875701955	10.10.20.52	10.10.20.30	TLSv1.2	125	Application Data

All responses indicate that the certificate´s status is *good*, like the one shown next, which belongs to the first response above:

No.	Time	Source	Destination	Protocol	Length	Info
16	11.792711648	10.10.20.40	10.10.20.52	TCP	68	80 → 49736 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
17	11.793124602	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
18	11.793319705	10.10.20.52	10.10.20.40	HTTP	277	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBT2Ccwj0sogAJDmm5nPT2C%2FgPHEtgQUlEOfgCR14SKnwkrFla0UhcEKeGECAhAA HTTP/1.1
19	11.795150518	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=1 Ack=222 Win=525568 Len=1460 [TCP segment of a reassembled PDU]
20	11.795164915	10.10.20.40	10.10.20.52	OCSP	1193	Response
21	11.795568668	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=222 Ack=2598 Win=2102272 Len=0
22	11.826238021	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBTgc2kYnVqyR7nclS0mZZyR4cdhigQUBBGr9Zu3TJdXx0KtXcm3Z0XKiEECAhAA HTTP/1.1
23	11.827978990	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=2598 Ack=441 Win=525312 Len=1460 [TCP segment of a reassembled PDU]
24	11.827993307	10.10.20.40	10.10.20.52	OCSP	1203	Response
25	11.828412396	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=441 Ack=5205 Win=2102272 Len=0
26	11.851830092	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBS0YNFlpGsTx4EbOEIpySbeFyb8XgQUcsnWd4bc9Tqi0A7ia9EpavtY6KQCAhAA HTTP/1.1
27	11.853617136	10.10.20.40	10.10.20.52	TCP	1516	80 → 49736 [ACK] Seq=5205 Ack=660 Win=525056 Len=1460 [TCP segment of a reassembled PDU]
28	11.853630284	10.10.20.40	10.10.20.52	OCSP	1203	Response
29	11.854058549	10.10.20.52	10.10.20.40	TCP	62	49736 → 80 [ACK] Seq=660 Ack=7812 Win=2102272 Len=0

```
> Frame 24: 1203 bytes on wire (9624 bits), 1203 bytes captured (9624 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
> Transmission Control Protocol, Src Port: 80, Dst Port: 49736, Seq: 4058, Ack: 441, Len: 1147
> [2 Reassembled TCP Segments (2607 bytes): #23(1460), #24(1147)]
> Hypertext Transfer Protocol
v Online Certificate Status Protocol
    responseStatus: successful (0)
  v responseBytes
      ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
    v BasicOCSPResponse
      v tbsResponseData
        > responderID: byName (1)
          producedAt: 2018-11-19 15:35:56 (UTC)
        v responses: 1 item
          v SingleResponse
            > certID
            > certStatus: good (0)
              thisUpdate: 2018-10-25 08:16:31 (UTC)
              nextUpdate: 2028-10-22 08:16:31 (UTC)
      > signatureAlgorithm (sha1WithRSAEncryption)
        Padding: 0
        signature: 6cd197fdc03b91208d9a1b588c68062e2ec638aad207a87e...
      > certs: 1 item
```

The following images show the output from the *TlsClientTest* tool and Microsoft Edge as well:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.1 Test3 (OCSP a)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

When the server certificate is revoked, the first OCSP response indicates that the certificate is *revoked*:

```
25 0.198356234   10.10.20.40   10.10.20.52   OCSP   1220 Response
26 0.198772887   10.10.20.52   10.10.20.40   TCP    62 49740 → 80 [ACK] Seq=660 Ack=7829 Win=262656 Len=0

▸ Frame 25: 1220 bytes on wire (9760 bits), 1220 bytes captured (9760 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.52
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 49740, Seq: 6665, Ack: 660, Len: 1164
▸ [2 Reassembled TCP Segments (2624 bytes): #24(1460), #25(1164)]
▸ Hypertext Transfer Protocol
▾ Online Certificate Status Protocol
    responseStatus: successful (0)
  ▾ responseBytes
      ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
    ▾ BasicOCSPResponse
      ▾ tbsResponseData
        ▸ responderID: byName (1)
          producedAt: 2018-11-19 15:42:15 (UTC)
        ▾ responses: 1 item
          ▾ SingleResponse
            ▸ certID
            ▸ certStatus: revoked (1)
              thisUpdate: 2018-10-25 08:16:55 (UTC)
              nextUpdate: 2028-10-22 08:16:55 (UTC)
      ▸ signatureAlgorithm (sha1WithRSAEncryption)
        Padding: 0
        signature: 0425ae814022851e1f17e1484c99c8e793616fad327e18e3...
      ▸ certs: 1 item
```

The client rejects the connection before sending application data:

No.	Time	Source	Destination	Protocol	Length	Info
11	0.060112859	10.10.20.52	10.10.20.30	TCP	62	49739 → 443 [ACK] Seq=491 Ack=1717 Win=2102272 Len=0
12	0.070538674	10.10.20.52	10.10.20.40	TCP	68	49740 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
13	0.070702980	10.10.20.40	10.10.20.52	TCP	68	80 → 49740 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
14	0.071148466	10.10.20.52	10.10.20.40	TCP	62	49740 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
15	0.071159510	10.10.20.52	10.10.20.40	HTTP	277	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBT2Ccwj0sogAJDmm5nPT2C%2FgPHEtgQUlEOfgCR14SKnwkrFla0UhcEKeGECAhAA HTTP/1.1
16	0.097486673	10.10.20.40	10.10.20.52	TCP	1516	80 → 49740 [ACK] Seq=1 Ack=222 Win=525568 Len=1460 [TCP segment of a reassembled PDU]
17	0.097492385	10.10.20.40	10.10.20.52	OCSP	1193	Response
18	0.097932930	10.10.20.52	10.10.20.40	TCP	62	49740 → 80 [ACK] Seq=222 Ack=2598 Win=262656 Len=0
19	0.124642051	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBTgc2kYnVqyR7ncls0mZZyR4cdhigQUBBGr9Zu3TJdXx0KtXcm3Z0XKiEECAhAA HTTP/1.1
20	0.141411990	10.10.20.40	10.10.20.52	TCP	1516	80 → 49740 [ACK] Seq=2598 Ack=441 Win=525312 Len=1460 [TCP segment of a reassembled PDU]
21	0.141419111	10.10.20.40	10.10.20.52	OCSP	1203	Response
22	0.141918892	10.10.20.52	10.10.20.40	TCP	62	49740 → 80 [ACK] Seq=441 Ack=5205 Win=262656 Len=0
23	0.168343773	10.10.20.52	10.10.20.40	HTTP	275	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBS8YNFlpGsTx4EbOEIpySbeFyb8XgQUcsnWd4bc9Tqi0A7ia9EpavtY6KQCAhAA HTTP/1.1
24	0.198346890	10.10.20.40	10.10.20.52	TCP	1516	80 → 49740 [ACK] Seq=5205 Ack=660 Win=525056 Len=1460 [TCP segment of a reassembled PDU]
25	0.198356234	10.10.20.40	10.10.20.52	OCSP	1220	Response
26	0.198772887	10.10.20.52	10.10.20.40	TCP	62	49740 → 80 [ACK] Seq=660 Ack=7829 Win=262656 Len=0
27	0.248466962	10.10.20.52	10.10.20.30	TCP	62	49739 → 443 [FIN, ACK] Seq=491 Ack=1717 Win=2102272 Len=0
28	0.248789654	10.10.20.30	10.10.20.52	TLSv1.2	109	Encrypted Alert
29	0.248841727	10.10.20.30	10.10.20.52	TCP	62	443 → 49739 [FIN, ACK] Seq=1770 Ack=492 Win=31360 Len=0
30	0.249064118	10.10.20.52	10.10.20.30	TCP	62	49739 → 443 [RST, ACK] Seq=492 Ack=1770 Win=0 Len=0
31	0.249305516	10.10.20.52	10.10.20.30	TCP	62	49739 → 443 [RST] Seq=492 Win=0 Len=0
32	0.252754893	10.10.20.52	10.10.20.40	TCP	62	49740 → 80 [RST, ACK] Seq=660 Ack=7829 Win=0 Len=0

When the intermediate 1 is revoked, the connection is also rejected, but the client directly does it before even sending any other OCSP request, as shown in the following screenshot:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
14	13.377004336	10.10.20.30	10.10.20.52	TCP	62	443 → 49741 [FIN, ACK] Seq=1770 Ack=491 Win=31360 Len=0
15	13.377425864	10.10.20.52	10.10.20.30	TCP	62	49741 → 443 [ACK] Seq=491 Ack=1771 Win=2102272 Len=0
16	16.660785341	10.10.20.52	10.10.20.40	TCP	68	49742 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
17	16.660942492	10.10.20.40	10.10.20.52	TCP	68	80 → 49742 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
18	16.661369947	10.10.20.52	10.10.20.40	TCP	62	49742 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
19	16.661534430	10.10.20.52	10.10.20.40	HTTP	277	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBT2Ccwj0sogAJDmm5nPT2C%2FgPHEtgQUlEOfgCR14SKnwkrFla0UhcEKeGECAhAA HTTP/1.1
20	16.689348191	10.10.20.40	10.10.20.52	TCP	1516	80 → 49742 [ACK] Seq=1 Ack=222 Win=525568 Len=1460 [TCP segment of a reassembled PDU]
21	16.689354312	10.10.20.40	10.10.20.52	OCSP	1210	Response
22	16.689769717	10.10.20.52	10.10.20.40	TCP	62	49742 → 80 [ACK] Seq=222 Ack=2615 Win=2102272 Len=0
23	16.750028128	10.10.20.52	10.10.20.30	TCP	62	49741 → 443 [RST, ACK] Seq=491 Ack=1771 Win=0 Len=0
24	16.756392743	10.10.20.52	10.10.20.40	TCP	62	49742 → 80 [RST, ACK] Seq=222 Ack=2615 Win=0 Len=0

The following images show the output from the *TlsClientTest* tool:

```
[+] Performing FIA_X509_EXT.1.1 Test3 (OCSP b)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is revoked.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

```
[+] Performing FIA_X509_EXT.1.1 Test3 (OCSP c)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The certificate is revoked.

Chain status: The revocation function was unable to check revocation for the certificate.

Chain status: The revocation function was unable to check revocation because the revocation server was offline.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Output from Microsoft Edge is the following when the connection is rejected:

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

To test OCSP stapling, the scenario is changed as explained in the previous section. This time, validation will be verified with help from a browser, be it Microsoft Edge or Internet Explorer.

With no certificate revocation, the connection is successfully performed, as shown in the following network traffic picture:

No.	Time	Source	Destination	Protocol	Length	Info
20	16.283822	10.10.20.25	10.10.20.40	TLSv1.2	239	Client Hello
21	16.286425	10.10.20.40	10.10.20.25	TLSv1.2	7191	Server Hello, Certificate, Certificate Status, Server Hello Done
22	16.288347	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=1461 Win=262144 Len=0
23	16.289133	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=2921 Win=262144 Len=0
24	16.289980	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=4381 Win=262144 Len=0
25	16.290801	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=5841 Win=262144 Len=0
26	16.291672	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=7138 Win=260608 Len=0
57	28.503039	10.10.20.25	10.10.20.40	TLSv1.2	412	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
58	28.503484	10.10.20.25	10.10.20.40	TLSv1.2	171	Application Data
59	28.503516	10.10.20.40	10.10.20.25	TCP	54	443 → 49678 [ACK] Seq=7138 Ack=661 Win=65024 Len=0
60	28.503682	10.10.20.25	10.10.20.40	TLSv1.2	331	Application Data
61	28.506599	10.10.20.40	10.10.20.25	TLSv1.2	145	Change Cipher Spec, Encrypted Handshake Message
62	28.506952	10.10.20.40	10.10.20.25	TLSv1.2	155	Application Data
63	28.507507	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=938 Ack=7229 Win=260608 Len=0
64	28.507530	10.10.20.40	10.10.20.25	TLSv1.2	123	Application Data
65	28.507871	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=938 Ack=7330 Win=262144 Len=0
66	28.508341	10.10.20.25	10.10.20.40	TLSv1.2	123	Application Data
67	28.508424	10.10.20.40	10.10.20.25	TLSv1.2	939	Application Data
68	28.508594	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=1007 Ack=7399 Win=261888 Len=0
69	28.509334	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=1007 Ack=8284 Win=261120 Len=0

It can be appreciated how the server now sends a Certificate Status message during the handshake process. It will only refer to the server certificate, so the revocation status of the two intermediate ones is checked independently via OCSP requests from the client. This Certificate Status message is shown next, where the status of the certificate is indicated to be good.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
20	16.283822	10.10.20.25	10.10.20.40	TLSv1.2	239	Client Hello
21	16.286425	10.10.20.40	10.10.20.25	TLSv1.2	7191	Server Hello, Certificate, Certificate Status, Server Hello Done
22	16.288347	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=1461 Win=262144 Len=0
23	16.289133	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=2921 Win=262144 Len=0
24	16.289980	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=4381 Win=262144 Len=0
25	16.290801	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=5841 Win=262144 Len=0
26	16.291672	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=186 Ack=7138 Win=260608 Len=0
57	28.503039	10.10.20.25	10.10.20.40	TLSv1.2	412	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
58	28.503484	10.10.20.25	10.10.20.40	TLSv1.2	171	Application Data
59	28.503516	10.10.20.40	10.10.20.25	TCP	54	443 → 49678 [ACK] Seq=7138 Ack=661 Win=65024 Len=0
60	28.503682	10.10.20.25	10.10.20.40	TLSv1.2	331	Application Data
61	28.506599	10.10.20.40	10.10.20.25	TLSv1.2	145	Change Cipher Spec, Encrypted Handshake Message
62	28.506952	10.10.20.40	10.10.20.25	TLSv1.2	155	Application Data
63	28.507507	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=938 Ack=7229 Win=260608 Len=0
64	28.507530	10.10.20.40	10.10.20.25	TLSv1.2	123	Application Data
65	28.507871	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=938 Ack=7330 Win=262144 Len=0
66	28.508341	10.10.20.25	10.10.20.40	TLSv1.2	123	Application Data
67	28.508424	10.10.20.40	10.10.20.25	TLSv1.2	939	Application Data
68	28.508594	10.10.20.25	10.10.20.40	TCP	60	49678 → 443 [ACK] Seq=1007 Ack=7399 Win=261888 Len=0

```
Certificate Status Type: OCSP (1)
OCSP Response Length: 2270
OCSP Response
  responseStatus: successful (0)
  responseBytes
    ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
    BasicOCSPResponse
      tbsResponseData
        responderID: byName (1)
        producedAt: 2018-11-02 07:07:39 (UTC)
        responses: 1 item
          SingleResponse
            certID
            certStatus: good (0)
            thisUpdate: 2018-10-25 08:16:32 (UTC)
            nextUpdate: 2028-10-22 08:16:32 (UTC)
```

This fact can also be verified from the browser where the test was performed, where the test web page was correctly retrieved under an HTTPS connection, for Microsoft Edge:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

When the server certificate is revoked, this Certificate Status message indicates *revoked*, as shown next:

```
No.  Time      Source       Destination  Protocol Length Info
  1 0.000000   10.10.20.55  10.10.20.40  TCP        66 49696 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.000064   10.10.20.40  10.10.20.55  TCP        66 443 → 49696 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  3 0.000283   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
  4 0.000584   10.10.20.55  10.10.20.40  TLSv1.2   239 Client Hello
  5 0.004006   10.10.20.40  10.10.20.55  TLSv1.2  7208 Server Hello, Certificate, Certificate Status, Server Hello Done
  6 0.004244   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [ACK] Seq=186 Ack=7155 Win=262144 Len=0
  7 0.080394   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [FIN, ACK] Seq=186 Ack=7155 Win=262144 Len=0
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49696, Seq: 1, Ack: 186, Len: 7154
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 7149
    ▸ Handshake Protocol: Server Hello
    ▸ Handshake Protocol: Certificate
    ▾ Handshake Protocol: Certificate Status
        Handshake Type: Certificate Status (22)
        Length: 2291
        Certificate Status Type: OCSP (1)
        OCSP Response Length: 2287
      ▾ OCSP Response
          responseStatus: successful (0)
        ▾ responseBytes
            ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
          ▾ BasicOCSPResponse
            ▾ tbsResponseData
              ▸ responderID: byName (1)
                producedAt: 2018-11-28 07:29:33 (UTC)
              ▾ responses: 1 item
                ▾ SingleResponse
                  ▸ certID
                  ▾ certStatus: revoked (1)
                    ▸ revoked
                    thisUpdate: 2018-10-25 08:16:55 (UTC)
                    nextUpdate: 2028-10-22 08:16:55 (UTC)
            ▸ signatureAlgorithm (sha1WithRSAEncryption)
```

This makes the client reject the connection, as shown in the next screenshot:

```
No.  Time      Source       Destination  Protocol Length Info
  1 0.000000   10.10.20.55  10.10.20.40  TCP        66 49696 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.000064   10.10.20.40  10.10.20.55  TCP        66 443 → 49696 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  3 0.000283   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
  4 0.000584   10.10.20.55  10.10.20.40  TLSv1.2   239 Client Hello
  5 0.004006   10.10.20.40  10.10.20.55  TLSv1.2  7208 Server Hello, Certificate, Certificate Status, Server Hello Done
  6 0.004244   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [ACK] Seq=186 Ack=7155 Win=262144 Len=0
  7 0.080394   10.10.20.55  10.10.20.40  TCP        60 49696 → 443 [FIN, ACK] Seq=186 Ack=7155 Win=262144 Len=0
  8 0.080453   10.10.20.40  10.10.20.55  TCP        54 443 → 49696 [RST, ACK] Seq=7155 Ack=187 Win=0 Len=0
```

This can also be verified from the browser, where validation errors are shown for Microsoft Edge:

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

30.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

30.2.4 Test 4

30.2.4.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1

© 2019 Microsoft Corporation. All rights reserved.

- CN = EE Test Intermediate CA 2 (no cRLSign key usage bit)
- CN = test.epoche.es

The certificates listed form a valid certification path. All certificates are available in pem format as well as pfx.

Additionally, the following OCSP signing certificates are added:

- CN = ocsp.test.epoche.es (signed by the root CA)
- CN = ocsp.test.epoche.es (signed by intermediate CA 1)
- CN = ocsp.test.epoche.es (signed by intermediate CA 2)

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine, CRL Server (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

For the OCSP test, the following machines are added:

- OCSP Server (Windows Server 2016)
- MITM Machine (Kali Linux)

These machines are all in the same network with the following configuration:

- Server Machine/CRL Server, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*
- OCSP Server, IP = 10.10.20.40

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS. Furthermore, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

Moreover, the following additional script is needed to perform the test case:

- For CRL (on CRL server machine):
  - *startCRLServer_intermediate2_noCRLSignKeyUsageBit.sh*: This script starts the CRL server in which a CRL signed by a CA without cRLsign key usage bit is provided.

The MITM machine shall act as a switch between the OCSP Server and the rest of the network, which means that its IP address does not play any role. It shall contain the *Rehtse* tool, a tool developed by the evaluator which is capable of modifying packets being routed through the machine on the fly, according to some criteria defined by regular expressions and BPF filters.

The Windows Server machine shall have the Online Responder role installed. To install it and configure it, see the Setup section for Test 3.

© 2019 Microsoft Corporation. All rights reserved.

Lastly, both on the client machine and the Windows Server machine, "C:\Windows\System32\drivers\etc\hosts" shall be modified so that it contains a line 10.10.10.10 test.epoche.es and another with 10.10.20.40 ocsp.test.epoche.es.

30.2.4.2 Procedure

The following steps shall be performed to complete this test assurance activity for CRLs signed with a certificate without cRLsign key usage bit set:

- Ensure that there is no communication between the client machine and the OCSP server since otherwise OCSP responses will have priority
- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the same machine, run the script *startCRLServer_intermediate2_noCRLSignKeyUsageBit.sh* to start the CRL server.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 7 when the menu is prompt.
- Once the TLS server is waiting for connection and the CRL server is properly configured, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

Next, revocation via OCSP shall be tested as follows:

- Stop the CRL server from the previous tests.
- Execute the following powershell command to clear the revocation cache: certutil ——urlcache * delete
- Restore the hosts file so that the Windows Server machine is reachable from ocsp.test.epoche.es. Only if the browser is not being used for testing, comment out the test.epoche.es line so that the CRL server can't be resolved when it falls back as the OCSP validation fails, which would clutter the network captures with failed connection attempts. The reference identifier test.epoche.es is set anyway in *TlsClientTest*.
- Configure the *Rehtse* tool with a *config.json* file that replaces the byte sequence *2b 06 01 05 05 07 03 09* by *2b 06 01 05 05 07 03 01* (switching from OCSP signing purpose to server authentication purpose).
- Ensuring the tool at the server is again waiting for connections, use the *TlsClientTest* tool to connect in the same way.
- Verify that validation fail both from the client's output and the *Wireshark* capture.

30.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

For the scenario with CRL signed by a certificate without cRLsign key usage bit, it has been verified that validation fails as expected by observing the network traffic. An excerpt is shown next:

17	23.634033556	10.10.10.52	10.10.20.30	TCP	60	49687 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
18	23.634216478	10.10.10.52	10.10.20.30	HTTP	187	GET /crl/ca.crl.pem HTTP/1.1
19	23.634732436	10.10.20.30	10.10.10.52	TCP	54	8000 → 49687 [ACK] Seq=1 Ack=134 Win=30336 Len=0
20	23.635013959	10.10.20.30	10.10.10.52	TCP	71	8000 → 49687 [PSH, ACK] Seq=1 Ack=134 Win=30336 Len=17 [TCP segment of a reassembled PDU]
21	23.635194349	10.10.20.30	10.10.10.52	HTTP	1283	HTTP/1.0 200 OK
22	23.635710182	10.10.10.52	10.10.20.30	TCP	60	49687 → 8000 [ACK] Seq=134 Ack=1248 Win=261376 Len=0
23	23.635910744	10.10.10.52	10.10.20.30	TCP	60	49687 → 8000 [FIN, ACK] Seq=134 Ack=1248 Win=261376 Len=0
24	23.636254562	10.10.20.30	10.10.10.52	TCP	54	8000 → 49687 [ACK] Seq=1248 Ack=135 Win=30336 Len=0
25	35.405826441	10.10.10.52	10.10.20.30	TCP	66	49688 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
26	35.406323075	10.10.20.30	10.10.10.52	TCP	66	8000 → 49688 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
27	35.407127570	10.10.10.52	10.10.20.30	TCP	60	49688 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
28	35.407268029	10.10.10.52	10.10.20.30	HTTP	212	GET /intermediate1/crl/intermediate1.crl.pem HTTP/1.1
29	35.407616153	10.10.20.30	10.10.10.52	TCP	54	8000 → 49688 [ACK] Seq=1 Ack=159 Win=30336 Len=0
30	35.407966269	10.10.20.30	10.10.10.52	TCP	71	8000 → 49688 [PSH, ACK] Seq=1 Ack=159 Win=30336 Len=17 [TCP segment of a reassembled PDU]
31	35.408136874	10.10.20.30	10.10.10.52	HTTP	1295	HTTP/1.0 200 OK
32	35.408751737	10.10.10.52	10.10.20.30	TCP	60	49688 → 8000 [ACK] Seq=159 Ack=1260 Win=261376 Len=0
33	35.408953570	10.10.10.52	10.10.20.30	TCP	60	49688 → 8000 [FIN, ACK] Seq=159 Ack=1260 Win=261376 Len=0
34	35.409180798	10.10.20.30	10.10.10.52	TCP	54	8000 → 49688 [ACK] Seq=1260 Ack=160 Win=30336 Len=0
35	47.518014719	10.10.10.52	10.10.20.30	TCP	66	49689 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
36	47.519313531	10.10.20.30	10.10.10.52	TCP	66	8000 → 49689 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
37	47.519959255	10.10.10.52	10.10.20.30	TCP	60	49689 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
38	47.520293207	10.10.10.52	10.10.20.30	HTTP	226	GET /intermediate1/intermediate2/crl/intermediate2.crl.pem HTTP/1.1
39	47.521098449	10.10.20.30	10.10.10.52	TCP	54	8000 → 49689 [ACK] Seq=1 Ack=173 Win=30336 Len=0
40	47.522142085	10.10.20.30	10.10.10.52	TCP	71	8000 → 49689 [PSH, ACK] Seq=1 Ack=173 Win=30336 Len=17 [TCP segment of a reassembled PDU]
41	47.522790447	10.10.20.30	10.10.10.52	HTTP	1295	HTTP/1.0 200 OK
42	47.523410693	10.10.10.52	10.10.20.30	TCP	60	49689 → 8000 [ACK] Seq=173 Ack=1260 Win=261376 Len=0
43	47.523607186	10.10.10.52	10.10.20.30	TCP	60	49689 → 8000 [FIN, ACK] Seq=173 Ack=1260 Win=261376 Len=0
44	47.524486049	10.10.20.30	10.10.10.52	TCP	54	8000 → 49689 [ACK] Seq=1260 Ack=174 Win=30336 Len=0
45	47.560720061	10.10.10.52	10.10.20.30	TCP	66	49690 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
46	47.561257892	10.10.20.30	10.10.10.52	TCP	66	8000 → 49690 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
47	47.561823640	10.10.10.52	10.10.20.30	TCP	60	49690 → 8000 [ACK] Seq=1 Ack=1 Win=262656 Len=0
48	47.562010414	10.10.10.52	10.10.20.30	HTTP	269	GET /intermediate1/intermediate2/crl/intermediate2.crl.pem HTTP/1.1
49	47.562319609	10.10.20.30	10.10.10.52	TCP	54	8000 → 49690 [ACK] Seq=1 Ack=216 Win=30336 Len=0
50	47.562747779	10.10.20.30	10.10.10.52	TCP	71	8000 → 49690 [PSH, ACK] Seq=1 Ack=216 Win=30336 Len=17 [TCP segment of a reassembled PDU]
51	47.562912337	10.10.20.30	10.10.10.52	HTTP	1295	HTTP/1.0 200 OK
52	47.563339022	10.10.10.52	10.10.20.30	TCP	60	49690 → 8000 [ACK] Seq=216 Ack=1260 Win=261376 Len=0
53	47.563541784	10.10.10.52	10.10.20.30	TCP	60	49690 → 8000 [FIN, ACK] Seq=216 Ack=1260 Win=261376 Len=0
54	47.563808686	10.10.20.30	10.10.10.52	TCP	54	8000 → 49690 [ACK] Seq=1260 Ack=217 Win=30336 Len=0
55	47.600494166	10.10.10.52	10.10.20.30	TCP	60	49686 → 443 [RST, ACK] Seq=491 Ack=1771 Win=0 Len=0

Every CRL request made by the client (though in the screenshot only the last CRL request for the server certificate is shown) is being retried once more since the retrieved CRL is not being accepted. At the end, the connection is being rejected by the client before sending any application data. This behaviour can also be observed in the tool's output, where it is informing that the certificate chain validation failed.

```
[+] Performing FIA_X509_EXT.1.1 Test4 (CRL)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A required certificate is not within its validity period when verifying against the current system clock or the timestamp in the signed file.

Chain status: The revocation function was unable to check revocation for the certificate.

Chain status: The revocation function was unable to check revocation because the revocation server was offline.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

Equivalent results are obtained for Microsoft Edge:

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

For the OCSP scenario, certificates without OCSP signing purpose are being presented, as shown in the following screenshot from one of the OCSP responses sent during the test:

It can be appreciated that the *Rehtse* tool changed the keyPurposeID from id−kp−OCSPSigning to id−kp−serverAuth, based on the following rule:

© 2019 Microsoft Corporation. All rights reserved.

```
{
        "debuglevel":0,
        "patterns":[
                {
                        "match":{
                                "bpf":"tcp",
                                "regex":".*\\x2b\\x06\\x01\\x05\\x05\\x07\\x03\\x09.*"
                        },
                        "replacement":{
                                "regex":"\\x2b\\x06\\x01\\x05\\x05\\x07\\x03\\x09",
                                "replacement":"\\x2b\\x06\\x01\\x05\\x05\\x07\\x03\\x01"
                        }
                }
        ]
}
```

The network capture shows the same behaviour as with the CRL case, retrying the certificate revocation information retrieval once again for every certificate before giving up and rejecting the connection:

No.	Time	Source	Destination	Protocol	Length	Info
32	23.550560039	10.10.10.52	10.10.20.40	TCP	60	49679 → 80 [ACK] Seq=705 Ack=7802 Win=262656 Len=0
33	23.585586737	10.10.10.52	10.10.20.40	HTTP	316	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBTgc2kYnVqyR7ncls0mZZyR4cdhigQUBBGr9Zu3TJdXx0KtXcm3Z0XKiEECAhAA HTTP/1.1
34	23.607070358	10.10.20.40	10.10.10.52	TCP	1514	80 → 49679 [ACK] Seq=7802 Ack=967 Win=64768 Len=1460 [TCP segment of a reassembled PDU]
35	23.621206247	10.10.20.40	10.10.10.52	OCSP	1201	Response
36	23.621574973	10.10.10.52	10.10.20.40	TCP	60	49679 → 80 [ACK] Seq=967 Ack=10409 Win=262656 Len=0
37	35.288019593	10.10.10.52	10.10.20.40	HTTP	273	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBS0YNFlpGsTx4EbOEIpySbeFyb8XgQUcsnWd4bc9Tqi0A7ia9EpavtY6KQCAhAA HTTP/1.1
38	35.313280188	10.10.20.40	10.10.10.52	TCP	1514	80 → 49679 [ACK] Seq=10409 Ack=1186 Win=64512 Len=1460 [TCP segment of a reassembled PDU]
39	35.330350364	10.10.20.40	10.10.10.52	OCSP	1201	Response
40	35.330848815	10.10.10.52	10.10.20.40	TCP	60	49679 → 80 [ACK] Seq=1186 Ack=13016 Win=262656 Len=0
41	35.376153233	10.10.10.52	10.10.20.40	HTTP	316	GET /ocsp/MEMwQTA%2FMD0wOzAJBgUrDgMCGgUABBS0YNFlpGsTx4EbOEIpySbeFyb8XgQUcsnWd4bc9Tqi0A7ia9EpavtY6KQCAhAA HTTP/1.1
42	35.398465288	10.10.20.40	10.10.10.52	TCP	1514	80 → 49679 [ACK] Seq=13016 Ack=1448 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
43	35.409236189	10.10.20.40	10.10.10.52	OCSP	1201	Response
44	35.409739265	10.10.10.52	10.10.20.40	TCP	60	49679 → 80 [ACK] Seq=1448 Ack=15623 Win=262656 Len=0
45	47.106870806	10.10.10.52	10.10.20.30	TCP	60	49678 → 443 [RST, ACK] Seq=491 Ack=1771 Win=0 Len=0
46	47.129769354	10.10.10.52	10.10.20.40	TCP	60	49679 → 80 [RST, ACK] Seq=1448 Ack=15623 Win=0 Len=0

The tool's output is also checked, showing the same result shown before for the CRL case, as it can be observed in the next images for *TlsClientTest* and Microsoft Edge:

```
[+] Performing FIA_X509_EXT.1.1 Test4 (OCSP)

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es -certificateRevocationCheck

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: The revocation function was unable to check revocation for the certificate.

Chain status: The revocation function was unable to check revocation because the revocation server was offline.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

30.2.4.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 4** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 4** activity.

30.2.5 Test 5

30.2.5.1 Setup

The setup is identical to the one defined for Test 1.

30.2.5.2 Procedure

The procedure is identical to the one defined for Test 1.

30.2.5.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *ee-tls-tool* intercepts the outgoing TLS certificate from the server and changes the first byte of the server certificate, increasing its value by one. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(bssl)))
{
    record = (TLS_record*)db->data;

   if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
   {
       handshake = (TLS_handshake_generic*)(record + 1);

       if (handshake->handshake_type == TLS_HANDSHAKE_CERTIFICATE) //certificate
       {
           //the first certificate is the one that's suppose to validate the site
           total_length = get_ssl_3_byte_number(handshake->length);
           if (total_length > 6)
           {
               pos = (unsigned char*)(handshake + 1);
               pos += 3; //skip the total cert length
               write_out(PRINT_INFO, "Changing first byte of the first
                   certificate from
                0x%02x to 0x%02x.", pos[3], pos[3] + 1);
               ++pos[3]; //first byte of data
           }
       }
   }

   data_buffer_merge(&ret, db);
   data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

```
 > Executing FIA_X509_EXT.1.1 Test 5
 > INSTRUCTIONS:
- Ensure certificates are valid (no revocation) and connect
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.25 on port 49691
   I> Starting TLS handshake...
      I> Changing first byte of the first certificate from 0x30 to 0x31.
      E> Socket recv error: Connection reset by peer
      E> TLS handshake error while recv'ing data!
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FIA_X509_EXT.1.1 Test 5
```

The effect on the certificate can be appreciated in the next picture:

© 2019 Microsoft Corporation. All rights reserved.

When the client receives the modified certificate from the server, it rejects the connection and stops the handshake as shown in the following network capture:

The fact that the connection was canceled can also be observed on the output of the tool used to perform the connection:

As the screenshot shows, the connection was closed due the inability to parse the malformed certificate.

Equivalent results are obtained when using a web browser like Edge:

© 2019 Microsoft Corporation. All rights reserved.

And in Internet Explorer for Windows Server:

30.2.5.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 5** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 5** activity.

30.2.6 Test 6

30.2.6.1 Setup

The setup is identical to the one defined for Test 1.

30.2.6.2 Procedure

The procedure is identical to the one defined for Test 1.

30.2.6.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *ee-tls-tool* intercepts the outgoing TLS certificate from the server and changes the last byte of the server certificate, increasing its value by one. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_generic*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_CERTIFICATE) //certificate
        {
            //the first certificate is the one that's suppose to validate the
                site
            total_length = get_ssl_3_byte_number(handshake->length);
            if (total_length > 6)
            {
                pos = (unsigned char*)(handshake + 1);
                pos += 3; //skip the total cert length
                cert_length = get_ssl_3_byte_number(pos);

                pos += cert_length + 3 - 1; //go to the last byte
                write_out(PRINT_INFO, "Changing last byte of the first
                    certificate from
                    0x%02x to 0x%02x.", *pos, (*pos) + 1);
                ++(\*pos); //last byte of data
            }
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

```
 > Executing FIA_X509_EXT.1.1 Test 6
 > INSTRUCTIONS:
- Ensure certificates are valid (no revocation) and connect
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.25 on port 49692
   I> Starting TLS handshake...
      I> Changing last byte of the first certificate from 0x57 to 0x58.
      E> TLS accept error: 1
      E> error:1408F119:SSL routines:ssl3_get_record:decryption failed or bad record mac
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FIA_X509_EXT.1.1 Test 6
```

© 2019 Microsoft Corporation. All rights reserved.

The effect on the certificate can be appreciated in the next picture:

When the client receives the modified certificate from the server, it rejects the connection and stops the handshake as shown in the following network capture:

The fact that the connection was canceled can also be observed on the output of the tool used to perform the connection:

```
[+] Performing FIA_X509_EXT.1.1 Test6

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: Authentication failed because the remote party has closed the transport stream.

Authentication failed - closing the connection.
```

As the screenshot shows, the connection was closed due the inability to parse the malformed certificate.

© 2019 Microsoft Corporation. All rights reserved.

Equivalent results are obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

30.2.6.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 6** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 6** activity.

30.2.7 Test 7

30.2.7.1 Setup

The setup is identical to the one defined for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

30.2.7.2 Procedure

The procedure is identical to the one defined for Test 1.

30.2.7.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* intercepts the outgoing TLS certificate from the server and changes the first byte in the modulus of its public key, increasing its value by one. The corresponding code snippet where the modification is performed is shown as follows:

```
while ((db = buffered_ssl_get_record(&hstate->bssl)))
{
    record = (TLS_record*)db->data;

    if (record->record_type == TLS_RECORD_HANDSHAKE) //TLS handshake
    {
        handshake = (TLS_handshake_generic*)(record + 1);

        if (handshake->handshake_type == TLS_HANDSHAKE_CERTIFICATE) //certificate
        {
            //the first certificate is the one that's suppose to validate the
                site
            total_length = get_ssl_3_byte_number(handshake->length);
            if (total_length > 6)
            {
                pos = (unsigned char*)(handshake + 1);
                mod_pos = (unsigned char*)memmem(pos, total_length,
                    hstate->modulus, hstate->size);

                if (mod_pos)
                {
                    write_out(PRINT_INFO, "Changing first byte of certificate
                        modulus from 0x%02x
                        to 0x%02x.", *mod_pos, (*mod_pos) + 1);
                    ++(\*mod_pos); //last byte of data
                }
            }
        }
    }

    data_buffer_merge(&ret, db);
    data_buffer_dtor(db);
}
```

The server´s output indicates the concrete modification that was performed:

© 2019 Microsoft Corporation. All rights reserved.

```
 > Executing FIA_X509_EXT.1.1 Test 7
 > INSTRUCTIONS:
- Ensure certificates are valid (no revocation) and connect
I> Searching for interface to listen on...
I> Listening on interface 0.0.0.0 on port 443.
   I> Sleeping for a second to catch client retries...
   I> Waiting for app to connect...
   I> Accepted connection from remote client 10.10.20.25 on port 49693
   I> Starting TLS handshake...
      I> Changing first byte of certificate modulus from 0xe9 to 0xea.
      E> TLS accept error: 1
      E> error:1408F119:SSL routines:ssl3_get_record:decryption failed or bad record mac
   I> SSL handshake failed!
   I> SSL init not finished, skipping graceful shutdown!
 > PASS: FIA_X509_EXT.1.1 Test 7
```

The effect on the certificate can be appreciated in the next picture:

```
No.  Time      Source       Destination  Protocol Length Info
   5 0.000017  10.10.20.25  10.10.20.30  TCP        66 58846 → 445 [ACK] Seq=58 Ack=58 Win=1444 Len=0 TSval=2352740435 TSecr=7240137
  14 0.000036  10.10.20.25  10.10.20.30  TCP        66 58846 → 445 [ACK] Seq=382 Ack=1144 Win=1444 Len=0 TSval=2352740477 TSecr=7240138
  17 0.000044  10.10.20.25  10.10.20.30  TCP        66 49693 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  18 0.000046  10.10.20.30  10.10.20.25  TCP        66 443 → 49693 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
  19 0.000048  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  20 0.000049  10.10.20.25  10.10.20.30  TLSv1.2   202 Client Hello
  21 0.000051  10.10.20.30  10.10.20.25  TCP        60 443 → 49693 [ACK] Seq=1 Ack=149 Win=30336 Len=0
  22 0.000052  10.10.20.25  10.10.20.25  TLSv1.2  2836 Server Hello, Certificate, Server Hello Done
  23 0.000097  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=149 Ack=2783 Win=262656 Len=0
  24 0.000100  10.10.20.25  10.10.20.30  TLSv1.2   396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  25 0.000103  10.10.20.30  10.10.20.25  TCP        60 443 → 49693 [FIN, ACK] Seq=2783 Ack=491 Win=31360 Len=0
  26 0.000104  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=491 Ack=2784 Win=262656 Len=0

    Certificate: 308203343082021ca003020102020 46efc4a77300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.epc
      signedCertificate
        version: v3 (2)
        serialNumber: 1862027895
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
      subjectPublicKeyInfo
        algorithm (rsaEncryption)
        subjectPublicKey: 3082010a0282010100eaeec451df1abee723e2fafadaa5b4...
          modulus: 0x00eaeec451df1abee723e2fafadaa5b4fc8631f647571243...
          publicExponent: 65537
      extensions: 2 items
    algorithmIdentifier (sha256WithRSAEncryption)

0160  00 30 82 01 0a 02 82 01  01 00 ea ee c4 51 df 1a   ·0····· ······Q··
0170  be e7 23 e2 fa fa da a5  b4 fc 86 31 f6 47 57 12   ··#····· ···1·GW·
0180  43 f5 cd 11 67 44 93 c0  66 3c 64 4f 14 a4 c7 90   C···gD·· f<dO····
0190  c8 a1 13 74 8b 9a b2 c3  80 6a 66 5a 2f 6e e2 fd   ···t···· ·jfZ/n··
01a0  86 e8 8c ed fc fa f5 18  61 33 89 ea 4e 6c d4 11   ····· a3··Nl··
01b0  4c e1 8b d3 20 9c 60 50  ba ac 07 81 80 2c d8 0b   L··· ·`P ·····,··
```

When the client receives the modified certificate from the server, it rejects the connection and stops the handshake as shown in the following network capture:

```
No.  Time      Source       Destination  Protocol Length Info
  17 0.000044  10.10.20.25  10.10.20.30  TCP        66 49693 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  18 0.000046  10.10.20.30  10.10.20.25  TCP        66 443 → 49693 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
  19 0.000048  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  20 0.000049  10.10.20.25  10.10.20.30  TLSv1.2   202 Client Hello
  21 0.000051  10.10.20.30  10.10.20.25  TCP        60 443 → 49693 [ACK] Seq=1 Ack=149 Win=30336 Len=0
  22 0.000052  10.10.20.25  10.10.20.25  TLSv1.2  2836 Server Hello, Certificate, Server Hello Done
  23 0.000097  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=149 Ack=2783 Win=262656 Len=0
  24 0.000100  10.10.20.25  10.10.20.30  TLSv1.2   396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  25 0.000103  10.10.20.30  10.10.20.25  TCP        60 443 → 49693 [FIN, ACK] Seq=2783 Ack=491 Win=31360 Len=0
  26 0.000104  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [ACK] Seq=491 Ack=2784 Win=262656 Len=0
  27 0.000105  10.10.20.25  10.10.20.30  TCP        60 49693 → 443 [FIN, ACK] Seq=491 Ack=2784 Win=262656 Len=0
  28 0.000106  10.10.20.30  10.10.20.25  TCP        60 443 → 49693 [ACK] Seq=2784 Ack=492 Win=31360 Len=0
```

The fact that the connection was canceled can also be observed on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.1 Test7

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Exception: Authentication failed because the remote party has closed the transport stream.

Authentication failed - closing the connection.
```

As the screenshot shows, the connection was closed due the inability to parse the malformed certificate.

Equivalent results are obtained when using a web browser like Edge:

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

30.2.7.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 7** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 7** activity.

30.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FIA_X509_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

31 FIA_X509_EXT.1.2

The assurance activity for the **FIA_X509_EXT.1.2** requirement is stated as follows:

> The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. The evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA.
>
> **Test 1**: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.
>
> **Test 2**: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate has the CA flag in the basicConstraints extension not set. The validation of the certificate path fails.
>
> **Test 3**: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate has the CA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.

31.1 Documentation Review activity

31.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

31.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

31.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
    a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE

© 2019 Microsoft Corporation. All rights reserved.

b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:

   i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).

   ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

31.2.1 Test 1

31.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2 (no *basicConstraints* extension)
- CN = test.epoche.es

The certificates listed form a valid certification path. The root certificate is available in pem format, the other ones will be sent by the server.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***.

© 2019 Microsoft Corporation. All rights reserved.

31.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 8 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.
- Verify the connection both from the client's output and the *Wireshark* capture.

31.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

The test case in *ee-tls-tool* sends a CA certificate without the basicConstraints extension. The effect on the certificate can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	49694 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000000	10.10.20.30	10.10.20.25	TCP	66	443 → 49694 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000001	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000001	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000002	10.10.20.30	10.10.20.25	TCP	60	443 → 49694 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000002	10.10.20.30	10.10.20.25	TLSv1.2	2800	Server Hello, Certificate, Server Hello Done
7	0.000005	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=149 Ack=2747 Win=262656 Len=0
8	0.000005	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000006	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000006	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [FIN, ACK] Seq=491 Ack=2822 Win=262656 Len=0
11	0.000006	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
12	0.000007	10.10.20.30	10.10.20.25	TCP	60	443 → 49694 [FIN, ACK] Seq=2875 Ack=492 Win=31360 Len=0

```
Length: 2670
Certificates Length: 2667
Certificates (2667 bytes)
  Certificate Length: 824
  Certificate: 308203343082021ca00302010202042469553130 0d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.e
  Certificate Length: 776
  Certificate: 30820304308201eca00302010202041e6fe0e3300d06092a... (id-at-organizationalUnitName=EE 1,id-at-organizationName=EE 1,id-at-commonName=EE
    signedCertificate
      version: v3 (2)
      serialNumber: 510648547
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
        rdnSequence: 3 items (id-at-organizationalUnitName=EE 1,id-at-organizationName=EE 1,id-at-commonName=EE TLS Tools Intermediate 1)
          RDNSequence item: 1 item (id-at-commonName=EE TLS Tools Intermediate 1)
          RDNSequence item: 1 item (id-at-organizationName=EE 1)
          RDNSequence item: 1 item (id-at-organizationalUnitName=EE 1)
      subjectPublicKeyInfo
    algorithmIdentifier (sha256WithRSAEncryption)
    Padding: 0
    encrypted: 4925cdb6182cc795ef6e03e6dea6b3f1f8da16bc6d9bb1c0...
  Certificate Length: 1058
  Certificate: 3082041e30820206a00302010202045cd0cd70300d06092a... (id-at-organizationalUnitName=EE 0,id-at-organizationName=EE 0,id-at-commonName=EE
```

As it can be observed, there are no extensions on the presented intermediate certificate. The behaviour on the client is shown in the next picture, where the connection is rejected, resulting in no application data being sent by the client:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	66	49694 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000000	10.10.20.30	10.10.20.25	TCP	66	443 → 49694 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000001	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
4	0.000001	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
5	0.000002	10.10.20.30	10.10.20.25	TCP	60	443 → 49694 [ACK] Seq=1 Ack=149 Win=30336 Len=0
6	0.000002	10.10.20.30	10.10.20.25	TLSv1.2	2800	Server Hello, Certificate, Server Hello Done
7	0.000005	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [ACK] Seq=149 Ack=2747 Win=262656 Len=0
8	0.000005	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
9	0.000006	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000006	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [FIN, ACK] Seq=491 Ack=2822 Win=262656 Len=0
11	0.000006	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
12	0.000007	10.10.20.30	10.10.20.25	TCP	60	443 → 49694 [FIN, ACK] Seq=2875 Ack=492 Win=31360 Len=0
13	0.000007	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [RST, ACK] Seq=492 Ack=2875 Win=0 Len=0
14	0.000007	10.10.20.25	10.10.20.30	TCP	60	49694 → 443 [RST] Seq=492 Win=0 Len=0

The fact that the connection was cancelled due to a validation failure can also be observed on the output of the tool used to perform the connection:

```
[+] Performing FIA_X509_EXT.1.2 Test1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate's basic constraint extension has not been observed.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

As the screenshot shows, the connection was closed due the lack of the required extension.

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

31.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

31.2.2 Test 2

31.2.2.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2 (CA flag not set in basicConstraints extension)
- CN = test.epoche.es

The certificates listed form a valid certification path. The root certificate is available in pem format, the other ones will be sent by the server.

31.2.2.2 Procedure

The procedure is identical to the one defined for Test 1.

31.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *ee-tls-tool* sends a CA certificate with the *basicConstraints* and the CA flag not set. The effect on the certificate can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
8	0.000034	10.10.20.30	10.10.20.25	TCP	66	443 → 49695 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
9	0.000037	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10	0.000040	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
11	0.000043	10.10.20.30	10.10.20.25	TCP	60	443 → 49695 [ACK] Seq=1 Ack=149 Win=30336 Len=0
12	0.000045	10.10.20.30	10.10.20.25	TLSv1.2	2830	Server Hello, Certificate, Server Hello Done
13	0.000109	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [ACK] Seq=149 Ack=2777 Win=262656 Len=0
14	0.000113	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
15	0.000116	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
16	0.000118	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [FIN, ACK] Seq=491 Ack=2852 Win=262656 Len=0

```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 2704
▾ Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 2700
  Certificates Length: 2697
  ▾ Certificates (2697 bytes)
    Certificate Length: 824
    ▸ Certificate: 308203343082021ca003020102020415cf2e60300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.e
    Certificate Length: 791
    ▾ Certificate: 30820313308201fba0030201020204654b3ed4300d06092a... (id-at-organizationalUnitName=EE 1,id-at-organizationName=EE 1,id-at-commonName=EE
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 1699430100
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-basicConstraints)
              Extension Id: 2.5.29.19 (id-ce-basicConstraints)
              BasicConstraintsSyntax [0 length]
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
        encrypted: b6f159b976ffdacaf5419a0575a5df3a4582f3801b1d4cdc...
```

The behaviour on the client is shown in the next picture, where the connection is rejected, resulting in no application data being sent by the client:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.000031	10.10.20.25	10.10.20.30	TCP	66	49695 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
8	0.000034	10.10.20.30	10.10.20.25	TCP	66	443 → 49695 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
9	0.000037	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10	0.000040	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
11	0.000043	10.10.20.30	10.10.20.25	TCP	60	443 → 49695 [ACK] Seq=1 Ack=149 Win=30336 Len=0
12	0.000045	10.10.20.30	10.10.20.25	TLSv1.2	2830	Server Hello, Certificate, Server Hello Done
13	0.000109	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [ACK] Seq=149 Ack=2777 Win=262656 Len=0
14	0.000113	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
15	0.000116	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
16	0.000118	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [FIN, ACK] Seq=491 Ack=2852 Win=262656 Len=0
17	0.000120	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
18	0.000123	10.10.20.30	10.10.20.25	TCP	60	443 → 49695 [FIN, ACK] Seq=2905 Ack=492 Win=31360 Len=0
19	0.000125	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [RST, ACK] Seq=492 Ack=2905 Win=0 Len=0
20	0.000126	10.10.20.25	10.10.20.30	TCP	60	49695 → 443 [RST] Seq=492 Win=0 Len=0

The fact that the connection was cancelled due to a validation failure can also be observed on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.2 Test2

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Certificate error: RemoteCertificateChainErrors

Chain status: A certificate's basic constraint extension has not been observed.

Exception: The remote certificate is invalid according to the validation procedure.

Authentication failed - closing the connection.
```

As the screenshot shows, the connection was closed due to the lack of the required extension (it is there in this case, but missing the needed flag for validation).

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

31.2.2.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 2** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 2** activity.

31.2.3 Test 3

31.2.3.1 Setup

The setup is identical to the one defined for Test 1 (IPs configuration and installed tools). The certificate used for the connection is different than the one used for Test 1:

- CN = EE Test Root CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2 (basicConstraints with CA flag set to true)
- CN = test.epoche.es

The certificates listed form a valid certification path. The root certificate is available in pem format, the other ones will be sent by the server.

31.2.3.2 Procedure

The procedure is identical to the one defined for Test 1.

31.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

The test case in *ee-tls-tool* sends a CA certificate with the basicConstraints extension, containing the CA flag set to true. The effect on the certificate can be appreciated in the next picture:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000010	10.10.20.25	10.10.20.30	TCP	66	49696 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	0.000017	10.10.20.30	10.10.20.25	TCP	66	443 → 49696 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7	0.000020	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8	0.000021	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
9	0.000024	10.10.20.30	10.10.20.25	TCP	60	443 → 49696 [ACK] Seq=1 Ack=149 Win=30336 Len=0
10	0.000026	10.10.20.30	10.10.20.25	TLSv1.2	2836	Server Hello, Certificate, Server Hello Done
11	0.000032	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=149 Ack=2783 Win=262656 Len=0
12	0.000034	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
13	0.000079	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
14	0.000082	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
15	0.000083	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data
16	0.000086	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
17	0.000088	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=560 Ack=3173 Win=262400 Len=0
18	0.000089	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [FIN, ACK] Seq=560 Ack=3173 Win=262400 Len=0
19	0.000091	10.10.20.30	10.10.20.25	TCP	60	443 → 49696 [ACK] Seq=3173 Ack=561 Win=31360 Len=0

```
    Handshake Type: Certificate (11)
    Length: 2706
    Certificates Length: 2703
  ▾ Certificates (2703 bytes)
      Certificate Length: 824
    ▸ Certificate: 308203343082021ca003020102020403c68b5c300d06092a... (id-at-organizationalUnitName=EE,id-at-organizationName=EE,id-at-commonName=test.
      Certificate Length: 794
    ▾ Certificate: 30820316308201fea003020102020477d19393300d06092a... (id-at-organizationalUnitName=EE 1,id-at-organizationName=EE 1,id-at-commonName=E
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 2010223507
        ▸ signature (sha256WithRSAEncryption)
        ▸ issuer: rdnSequence (0)
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▾ extensions: 1 item
          ▾ Extension (id-ce-basicConstraints)
              Extension Id: 2.5.29.19 (id-ce-basicConstraints)
            ▾ BasicConstraintsSyntax
                cA: True
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
```

The behaviour on the client is shown in the next picture, where the validation is successful and both client and server send application data to each other.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000010	10.10.20.25	10.10.20.30	TCP	66	49696 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	0.000017	10.10.20.30	10.10.20.25	TCP	66	443 → 49696 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7	0.000020	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8	0.000021	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
9	0.000024	10.10.20.30	10.10.20.25	TCP	60	443 → 49696 [ACK] Seq=1 Ack=149 Win=30336 Len=0
10	0.000026	10.10.20.30	10.10.20.25	TLSv1.2	2836	Server Hello, Certificate, Server Hello Done
11	0.000032	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=149 Ack=2783 Win=262656 Len=0
12	0.000034	10.10.20.25	10.10.20.30	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
13	0.000079	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
14	0.000082	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
15	0.000083	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data
16	0.000086	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
17	0.000088	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [ACK] Seq=560 Ack=3173 Win=262400 Len=0
18	0.000089	10.10.20.25	10.10.20.30	TCP	60	49696 → 443 [FIN, ACK] Seq=560 Ack=3173 Win=262400 Len=0
19	0.000091	10.10.20.30	10.10.20.25	TCP	60	443 → 49696 [ACK] Seq=3173 Ack=561 Win=31360 Len=0

The fact that the validation was successful can also be observed on the output of the tool used to perform the connection:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.1.2 Test3

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

As the screenshot shows, the connection was successful.

Equivalent results are obtained when using a web browser like Edge.

And in Internet Explorer for Windows Server:

© 2019 Microsoft Corporation. All rights reserved.

31.2.3.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 3** activity.

31.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FIA_X509_EXT.1.2.

© 2019 Microsoft Corporation. All rights reserved.

32 FIA_X509_EXT.2.1

The assurance activity for the **FIA_X509_EXT.2.1** requirement is stated as follows:

> The evaluator will acquire or develop an application that uses the OS TLS mechanism with an X.509v3 certificate. The evaluator will then run the application and ensure that the provided certificate is used to authenticate the connection.
>
> The evaluator will repeat the activity for any other selections listed.

32.1 Documentation Review activity

32.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

32.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

32.2 Test Activity

To perform the TLS tests required by the Assurance Activities and check their correct outcome, different pieces of software were used:

1. Software acting as a TLS server:
   a. *ee-tls-tool*, a command line tool developed by the evaluation laboratory in order to create a TLS server which is automatically configured prior each test case execution.
2. Software acting as a client from the TOE
   b. *W10ClientAutomator.ps1*, a Powershell script which automatically configures the TOE as a client prior each test case execution. This script allows the evaluator execute test cases using:
      i. different web browsers that come bundled with the TOE (i.e. Internet Explorer and Microsoft Edge).
      ii. *TlsClientTest*, a command line tool developed by the evaluation laboratory. This tool provides more information about the concrete point where SSL validation failed in case of connection error.

Information about their suitability for this evaluation is given in the introduction to FCS_TLSC_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

32.2.1 Test 1

32.2.1.1 Setup

The following certificates, created automatically by *ee-tls-tool* and installed by *W10ClientAutomator.ps1*, shall be used to perform the assurance activities listed on the Protection Profile:

- CN = EE TestRoot CA
- CN = EE Test Intermediate CA 1
- CN = EE Test Intermediate CA 2
- CN = test.epoche.es
- CN = EE Client (client certificate with friendly name "EE Client Certificate")

The listed certificates form a valid certification path. The root certificate is available in pem format, the client certificate in pfx format, and the other ones will be sent by the server.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine (Debian 9 Stretch)
- Client Machine (Platforms listed in the ST)

Both machines are in the same network with the following configuration:

- Server Machine, IP = 10.10.20.30
- Client Machine, IP = 10.10.10.*

Server machine contains the *ee-tls-tool* software, a test suite for TLS and X509 Common Criteria evaluations. It acts as a TLS server configured according to each test. In addition, the *Wireshark* network analyzer is installed on the server machine.

The client machine shall contain the PowerShell script *W10ClientAutomator.ps1* and the *TlsClientTest* tool, which uses the System.Net.Security classes from .Net framework to perform SSL connections, effectively using the authentication mechanism from the OS.

Moreover, the client machine shall have the secure configuration enabled, according to the section **5.2 Operational rerequisites** of the ***Operational Guidance***.

32.2.1.2 Procedure

The following steps shall be performed to complete this test assurance activity:

- On the server machine, run this test case on *ee-tls-tool*. The network traffic capture is started automatically by the tool.
- On the client machine, run the script *W10ClientAutomator.ps1* with the following parameters: <client_method> (e.g. ie for Internet Explorer, edge for Microsoft Edge, or cli for *TlsClientTest* tool) and <ip_server_address> and select the option number 8 when the menu is prompt.
- Once the server is waiting for connection, press Enter to start each test belonging to the suite.

© 2019 Microsoft Corporation. All rights reserved.

- Verify the connection both from the client's output and the *Wireshark* capture.

32.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

Both for connections from *TlsClientTest* and connections from a browser (Microsoft Edge or Internet Explorer), the client provides its certificate during the handshake. The tool gets it by accessing the certificate store of the system, whereas browsers prompt the user for selection. In both cases, the certificate is sent along with the Client Key Exchange message, as shown in the following picture:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [ACK] Seq=1 Ack=1 Win=1026 Len=0
2	0.000001	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
3	0.000002	10.10.20.30	10.10.20.25	TCP	60	443 → 49697 [ACK] Seq=1 Ack=149 Win=237 Len=0
4	0.000003	10.10.20.30	10.10.20.25	TLSv1.2	2981	Server Hello, Certificate, Certificate Request, Server Hello Done
5	0.000013	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [ACK] Seq=149 Ack=2928 Win=1026 Len=0
6	0.000014	10.10.20.25	10.10.20.30	TCP	1514	49697 → 443 [ACK] Seq=149 Ack=2928 Win=1026 Len=1460 [TCP segment of a reassembled PDU]
7	0.000017	10.10.20.25	10.10.20.30	TLSv1.2	294	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
8	0.000018	10.10.20.30	10.10.20.25	TCP	60	443 → 49697 [ACK] Seq=2928 Ack=1849 Win=283 Len=0
9	0.000018	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000019	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
11	0.000020	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data

```
▸ Internet Protocol Version 4, Src: 10.10.20.30, Dst: 10.10.20.25
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49697, Seq: 1, Ack: 149, Len: 2927
▾ Secure Sockets Layer
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  ▸ TLSv1.2 Record Layer: Handshake Protocol: Certificate
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Certificate Request
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 140
```

The client is therefore authenticated and the connection succeeds, as shown in the following network capture for the direct connection using TLS with *TlsClientTest*:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [ACK] Seq=1 Ack=1 Win=1026 Len=0
2	0.000001	10.10.20.25	10.10.20.30	TLSv1.2	202	Client Hello
3	0.000002	10.10.20.30	10.10.20.25	TCP	60	443 → 49697 [ACK] Seq=1 Ack=149 Win=237 Len=0
4	0.000003	10.10.20.30	10.10.20.25	TLSv1.2	2981	Server Hello, Certificate, Certificate Request, Server Hello Done
5	0.000013	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [ACK] Seq=149 Ack=2928 Win=1026 Len=0
6	0.000014	10.10.20.25	10.10.20.30	TCP	1514	49697 → 443 [ACK] Seq=149 Ack=2928 Win=1026 Len=1460 [TCP segment of a reassembled PDU]
7	0.000017	10.10.20.25	10.10.20.30	TLSv1.2	294	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
8	0.000018	10.10.20.30	10.10.20.25	TCP	60	443 → 49697 [ACK] Seq=2928 Ack=1849 Win=283 Len=0
9	0.000018	10.10.20.30	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
10	0.000019	10.10.20.25	10.10.20.30	TLSv1.2	123	Application Data
11	0.000020	10.10.20.30	10.10.20.25	TLSv1.2	315	Application Data
12	0.000021	10.10.20.30	10.10.20.25	TLSv1.2	107	Encrypted Alert
13	0.000022	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [ACK] Seq=1918 Ack=3318 Win=1025 Len=0
14	0.000023	10.10.20.25	10.10.20.30	TCP	60	49697 → 443 [FIN, ACK] Seq=1918 Ack=3318 Win=1025 Len=0
15	0.000024	10.10.20.30	10.10.20.25	TCP	60	443 → 49697 [ACK] Seq=3318 Ack=1919 Win=283 Len=0

The fact that the connection was successful can also be observed on the output of *TlsClientTest* tool:

© 2019 Microsoft Corporation. All rights reserved.

```
[+] Performing FIA_X509_EXT.2.1 Test1

When EE TLS Tool is ready, press Enter to run the test...
:


.\TlsClientTest\TlsClientTest.exe 10.10.20.30 test.epoche.es

Client connected.

Server says:
HTTP/1.1 200 OK
Date: Tue, 20 Feb 2018 08:00:00 GMT
Server: EE TLS Tools
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>

Client closed.
```

When performing the connection using a web browser, a prompt similar to the one shown next appears, which shows the client certificate which will be used for authentication.

© 2019 Microsoft Corporation. All rights reserved.

The fact that the connection is successful once the certificate is selected is shown in the following browser capture:

And in Internet Explorer for Windows Server:

32.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

© 2019 Microsoft Corporation. All rights reserved.

32.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FIA_X509_EXT.2.1.

© 2019 Microsoft Corporation. All rights reserved.

33 FMT_MOF_EXT.1.1 & FMT_SMF_EXT.1.1

The assurance activity for the **FMT_MOF_EXT.1** and **FMT_SMF_EXT.1** requirements is stated as follows:

> **FMT_MOF_EXT.1:** The evaluator shall verify that the TSS describes those management functions that are restricted to Administrators, including how the user is prevented from performing those functions, or not able to use any interfaces that allow access to that function.
>
> - **Test 1:** For each function that is indicated as restricted to the administrator, the evaluation shall perform the function as an administrator, as specified in the Operational Guidance, and determine that it has the expected effect as outlined by the Operational Guidance and the SFR. The evaluator shall then perform the function (or otherwise attempt to access the function) as a non-administrator and observe that they are unable to invoke that functionality.
>
> **FMT_SMF_EXT.1:** The evaluator will verify that every management function captured in the ST is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function. The evaluator will test the operating system's ability to provide the management functions by configuring the operating system and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.

33.1 Documentation Review activity

33.1.1 Findings

The evaluator has reviewed the ***Security Target*** document to identify all the management functions defined by the vendor. These functions are listed in the section **6.5 Security Management** of the TSS, as it can be observed in the following image extracted from the ***Security Target*** document:

© 2019 Microsoft Corporation. All rights reserved.

#	Management Function	Administrator	User
1.	Enable/disable screen lock	√	√
2.	Configure screen lock inactivity timeout	√	√
3.	Configure local audit storage capacity	√	
4.	Configure minimum password Length	√	
5.	~~Configure minimum number of special characters in password~~		
6.	~~Configure minimum number of numeric characters in password~~		
7.	~~Configure minimum number of uppercase characters in password~~		
8.	~~Configure minimum number of lowercase characters in password~~		
9.	~~Enable/disable unauthenticated logon~~		
10.	Configure lockout policy for unsuccessful authentication attempts through [***timeouts between attempts, limiting number of attempts during a time period***]	√	
11.	Configure host-based firewall	√	
12.	Configure name/address of directory server to bind with[26]	√	
13.	Configure name/address of remote management server from which to receive management settings	√	
14.	~~Configure name/address of audit/logging server to which to send audit/logging records~~		
15.	Configure audit rules	√	
16.	Configure name/address of network time server	√	
17.	Enable/disable automatic software update	√	
18.	Configure Wi-Fi interface	√	
19.	Enable/disable Bluetooth interface	√	
20.	Configure USB interfaces	√	
21.	Enable/disable [**local area network interface**]	√	
22.	[**manage Windows Diagnostics settings,**	√	√
	Configure remote connection inactivity timeout]	√	

Table 19 Windows Security Management Functions

In addition, the evaluator has also reviewed the ***Operational Guidance*** document, which includes within its section **6. Management Functions** all the management functions explained throughout the document. In addition, for each management function, it is described all the available methods to configure each function, as well as, the applicable roles and Windows 10 or Windows Server 2019 versions.

© 2019 Microsoft Corporation. All rights reserved.

6.2.1.1 Configuring with MDM

Role	IT Administrator
Windows Editions	Pro, Enterprise

Client certificates may be managed by the IT Administrator using MDM. See the MDM solution documentation for detailed management actions. The following article describes the MDM policy for client certificate management, including deleting certificates:

- ClientCertificateInstall CSP - https://docs.microsoft.com/en-us/windows/client-management/mdm/clientcertificateinstall-csp

6.2.1.2 Configuring with the Windows UI

Role	Standard User
Windows Editions	Home, Pro, Enterprise

The following article describes how to manually import a certificate:

- Import a Certificate: http://technet.microsoft.com/en-us/library/cc754489.aspx

The user obtains a client certificate for authentication by following the procedures in the following article:

- Obtain a Certificate: https://technet.microsoft.com/en-us/library/cc754246.aspx

For each of these management functions, the ***Operational Guidance*** document, includes links to the OS vendor support site, where information about how to configure each of these functions is provided.

These links also provides information about managing these functions either with a standard user or as administrator, specifying the different steps needed for each one. Moreover, the managements functions contains instructions for Windows 10 and Windows Server 2019 editions, in the cases that is needed a different managament approach for each of them.

This is an example of the information provided in the vendor website about how to set up a wireless network:

© 2019 Microsoft Corporation. All rights reserved.

Setting up a wireless network

Applies to: Windows 10, Windows 7, Windows 8.1

A wireless network at home lets you get online from more places in your house. This article describes the basic steps for setting up a wireless network and starting to use it.

Get the right equipment

Before you can set up your wireless network, here's what you'll need:

Broadband Internet connection and modem. A broadband Internet connection is a high-speed Internet connection. Digital Subscriber Line (DSL) and cable are two of the most common broadband connections. You can get a broadband connection by contacting an Internet service provider (ISP). Typically, ISPs that provide DSL are telephone companies and ISPs that provide cable are cable TV companies. ISPs frequently offer broadband modems. Some ISPs also offer combination modem/wireless routers. You can also find these at computer or electronics stores, and online.

Wireless router. A router sends info between your network and the Internet. With a wireless router, you can connect PCs to your network using radio signals instead of wires. There are several different kinds of wireless network technologies, which include 802.11a, 802.11b, 802.11g, 802.11n, and 802.11ac.

Wireless network adapter. A wireless network adapter is a device that connects your PC to a wireless network. To connect your portable or desktop PC to your wireless network, the PC must have a wireless network adapter. Most laptops and tablets—and some desktop PCs—come with a wireless network adapter already installed.

To check whether your PC has a wireless network adapter:

1. Select the **Start** button, type **device manager** in the search box, and then select **Device Manager**.
2. Expand **Network adapters**.
3. Look for a network adapter that might have **wireless** in the name.

Setting up the modem and Internet connection

After you have all the equipment, you'll need to set up your modem and Internet connection. If your modem wasn't set up for you by your Internet service provider (ISP), follow the instructions that came with your modem to connect it to your PC and the Internet. If you're using Digital Subscriber Line (DSL), connect your modem to a phone jack. If you're using cable, connect your modem to a cable jack.

33.1.2 Verdict

The evaluator considers that all the security management functions declared in the ***Security Target*** document are properly defined in the ***Operational Guidance*** document, providing enough information to allow the evaluator to perform each management function.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

33.2 Test Activity

33.2.1 Test 1 - Enable/Disable Screen Lock

The screen lock will be enabled and disabled during this test case.

© 2019 Microsoft Corporation. All rights reserved.

33.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A user account with no administrator rights shall exist. This user shall belong to the default *Users* group.
- An administrator account shall exist. This account shall belong to the default *Administrators* group.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

33.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 1 - Enable&DisableScreenLock.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 1 - Enable&DisableScreenLock.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 1
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Other Logon/Logoff "Events  /success:enable
    /failure:enable
auditpol /set /subcategory:""Logon  /success:enable /failure:enable
#clear the audit log
wevtutil cl security

Write-Host "[I]: Enabling/Disabling screen lock..."

# Lock account
logoff
Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4800 1
pause
GetAuditLog Security 4801 1

Write-Host "[I]: Restoring audit rules..."
auditpol /set /subcategory:"Other Logon/Logoff Events"  /success:disable
/failure:disable
```

© 2019 Microsoft Corporation. All rights reserved.

```
auditpol /set /subcategory:"Logon" /success:disable /failure:disable
```

- Enter the correct password to unlock the workstation and observe the results. Two audit entries shall be shown.
- Observe the result and log out as Administrator.
- Log out, and log in again, but this time using the user account without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following images:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : The workstation was locked.

                       Subject:
                        Security ID:            S-1-5-21-473433637-2525778249-1789179070-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x354C2
                        Session ID:     1
Id                   : 4800
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12551
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 1473
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 824
ThreadId             : 936
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/6/2018 10:48:28 AM
ActivityId           : 5028d5f3-8cf7-0007-f8d5-2850f78cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Other Logon/Logoff Events
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

```
Message              : The workstation was unlocked.

                       Subject:
                        Security ID:            S-1-5-21-473433637-2525778249-1789179070-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x354C2
                        Session ID:     1
Id                   : 4801
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12551
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 1479
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 824
ThreadId             : 936
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/6/2018 10:48:30 AM
ActivityId           : 5028d5f3-8cf7-0007-f8d5-2850f78cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Other Logon/Logoff Events
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

Using a **User account**, the same results were obtained. However, the user account does not have permission to see the event log.

```
[I]: Showing audit event...
Get-WinEvent : Could not retrieve information about the Security log. Error: Attempted to perform an unauthorized operation..
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : LogInfoUnavailable,Microsoft.PowerShell.Commands.GetWinEventCommand
```

33.2.1.4 Verdict

As the above results state, the user with administrator rights and the user without administrator rights can enable and disable the screen lock.

© 2019 Microsoft Corporation. All rights reserved.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

33.2.2 Test 2 - Configure Screen Lock Inactivity Timeout

The screen lock inactivity timeout will be configured during this test case.

33.2.2.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 2 - ConfigureScreenLockInactivityTimeout.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 2 - ConfigureScreenLockInactivityTimeout.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 – Test 2
#

Write–Host "[Q]: Is the PC connected to an AC power or to a DC Power?"
$power = Read–Host –Prompt 'Input your actual power source (AC or DC)'

if ($power –eq 'AC') {
    #AC Power
    $timeout = Read–Host –Prompt 'Set the timeout for the inactivity period'
    Write–Host "[I]: Configuring Screen Lock Inactivity Timeout..."
    Start–Process "cmd.exe" "/c powercfg.exe /setacvalueindex
    SCHEME_CURRENT SUB_VIDEO VIDEOIDLE $timeout" –WorkingDirectory
    "C:\Windows\System32"
    Start–Process "cmd.exe" "/c powercfg.exe /setactive SCHEME_CURRENT"
    –WorkingDirectory "C:\Windows\System32"
} elseif ($power –eq 'DC') {
    #DC Power
    $timeout = Read–Host –Prompt 'Set the timeout for the inactivity period'
    Write–Host "[I]: Configuring Screen Lock Inactivity Timeout..."
    Start–Process "cmd.exe" "/c powercfg.exe /setacvalueindex
    SCHEME_CURRENT SUB_VIDEO VIDEOIDLE $timeout" –WorkingDirectory
    "C:\Windows\System32"
    Start–Process "cmd.exe" "/c powercfg.exe /setactive SCHEME_CURRENT"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    -WorkingDirectory "C:\Windows\System32"
}
else {
    Write-Host "[E]: Please input yor porwer source AC or DC."
}
```

- The script will ask for the input power source of the platform where it is executed. Type AC if the platform is connected to a power source or DC if is running from the battery. Then, type the desired screen lock inactivity timeout. For example, during the test execution, the evaluator set the screen lock inactivity timeout to 2 minutes.

```
PS C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT> & '.\Test 2 - ConfigureScreenLockInactivityTimeout.ps1'
[Q]: Is the PC connected to an AC power or to a DC Power?
Input your actual power source (AC or DC): ac
Set the timeout for the inactivity period: 120
[I]: Configuring Screen Lock Inactivity Timeout...
```

- Wait during the configured period time (e.g. 2 minutes).
- Observe the result.
- Log out, and log in again, but this time, using a user account with no administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has obtained the same result for both users, the screen has been blocked after the configured inactivity time is reached.

33.2.2.4 Verdict

As the above results state, both user with administrator rights and user without administrator rights can configure the screen lock inactivity timeout.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

33.2.3 Test 3 - Configure Local Audit Storage Capacity

The local audit storage capacity will be configured during this test case.

33.2.3.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.3.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 3 - ConfigureLocalAuditStorageCapacity.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 3 - ConfigureLocalAuditStorageCapacity.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 3
```

© 2019 Microsoft Corporation. All rights reserved.

```
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /category:"Object Access" /success:enable /failure:enable

#View the current maximum size of log
Write-Host "[I]: View the current maximum size of log (before the change)..."
wevtutil gl security

#clear the event log
wevtutil cl Security

Write-Host "[I]: Configuring local audit storage capacity..."
#Set the maximum size of log
wevtutil sl security /ms:20972017

#see the current maximum size
Write-Host "[I]: View the current maximum size of log (after the change)..."
wevtutil gl security
Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4657 1

Write-Host "[I]: Restoring audit rules..."
auditpol /set /category:"Object Access" /success:disable /failure:disable
```

- Observe the result.
- Log out log in again, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.3.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following image:

```
PS C:\Users\Administrator.SURFACEBOOK2\Desktop\FMT> & '.\Test 3 - ConfigureLocalAuditStorageCapacity.ps1'
[I]: Setting up audit rules...
The command was successfully executed.
[I]: View the current maximum size of log (before the change)...
name: security
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\security.evtx
  retention: false
  autoBackup: false
  maxSize: 20971520
publishing:
  fileMax: 1
[I]: Configuring local audit storage capacity...
[I]: View the current maximum size of log (after the change)...
name: security
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\security.evtx
  retention: false
  autoBackup: false
  maxSize: 20972018
publishing:
  fileMax: 1
```

```
EventID            : 4657
MachineName        : SURFACEBOOK2
Data               : {}
Index              : 1547
Category           : (12801)
CategoryNumber     : 12801
EntryType          : SuccessAudit
Message            : A registry value was modified.

                     Subject:
                        Security ID:            S-1-5-19
                        Account Name:           LOCAL SERVICE
                        Account Domain:         NT AUTHORITY
                        Logon ID:               0x3e5

                     Object:
                        Object Name:            \REGISTRY\MACHINE\SYSTEM\ControlSet001\Control\WMI\Autologger\EventLog-Security
                        Object Value Name:      OwningChannel
                        Handle ID:              0x6cc
                        Operation Type:         %%1905

                     Process Information:
                        Process ID:             0x6dc
                        Process Name:           C:\Windows\System32\svchost.exe

                     Change Information:
                        Old Value Type:         %%1873
                        Old Value:              security
                        New Value Type:         %%1873
                        New Value:              Security
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-19, LOCAL SERVICE, NT AUTHORITY, 0x3e5...}
InstanceId         : 4657
TimeGenerated      : 12/6/2018 10:52:39 AM
TimeWritten        : 12/6/2018 10:52:39 AM
UserName           :
Site               :
Container          :
```

Therefore, the changes were applied successfully.

© 2019 Microsoft Corporation. All rights reserved.

Using a **User account,** the evaluator has executed the script with no success and the following error was shown:

```
PS C:\Users\userFMT\Desktop\FMT> & '.\Test 3 - ConfigureLocalAuditStorageCapacity.ps1'
[I]: Setting up audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: View the current maximum size of log (before the change)...
Failed to read configuration for log security.
Access is denied.
Failed to clear log Security.
Access is denied.
[I]: Configuring local audit storage capacity...
Failed to read configuration for log security.
Access is denied.
[I]: View the current maximum size of log (after the change)...
Failed to read configuration for log security.
Access is denied.
[I]: Showing audit event...
Get-EventLog : Requested registry access is not allowed.
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:6 char:14
+ ...    $event = Get-EventLog $eventType -InstanceId $eventID -Newest $new ...
+                ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-EventLog], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.GetEventLogCommand

[E]: There is no event log that matches with these conditions: eventId = 4657, newest = 1
[I]: Restoring audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.
```

33.2.3.4 Verdict

As the above results state, the audit storage capacity can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3**.

33.2.4 Test 4 - Configure Minimum Password Length

A minimum password length will be configured during this test case.

33.2.4.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.4.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 4 - ConfigureMinimumPassword.ps1.*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

© 2019 Microsoft Corporation. All rights reserved.

> .\Test 4 - ConfigureMinimumPassword.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 4
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Authentication Policy Change" /success:enable
/failure:enable

Write-Host "[I]: Configuring minimum password length..."
#Change minimum password length
net accounts /minpwlen:7
Start-Sleep -s 1
Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4739 2

Write-Host "[I]: Restoring audit rules..."
auditpol /set /subcategory:"Authentication Policy Change" /success:disable
/failure:disable
```

- Observe the result. An audit entry shall be shown.
- Log out, and log in again, but this time, using a user account with no administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.4.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following image:

```
Message              : Domain Policy was changed.

                       Change Type:             Logoff Policy modified

                       Subject:
                        Security ID:            S-1-5-21-473433637-2525778249-1789179070-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x354C2

                       Domain:
                        Domain Name:            SURFACEBOOK2
                        Domain ID:              S-1-5-21-473433637-2525778249-1789179070

                       Changed Attributes:
                        Min. Password Age:      -
                        Max. Password Age:      -
                        Force Logoff:           -
                        Lockout Threshold:      -
                        Lockout Observation Window:     -
                        Lockout Duration:       -
                        Password Properties:    -
                        Min. Password Length:   -
                        Password History Length:        -
                        Machine Account Quota:  -
                        Mixed Domain Mode:      -
                        Domain Behavior Version:        -
                        OEM Information:        -

                       Additional Information:
                        Privileges:             -
Id                   : 4739
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 13569
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 1617
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 824
ThreadId             : 936
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/6/2018 10:54:39 AM
ActivityId           : 5028d5f3-8cf7-0007-f8d5-2850f78cd401
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Authentication Policy Change
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

As the previous image states, the new password policy was changed with success to the new value.

In addition, the evaluator has checked the *Local Security Policy* (*WIN+r*, then type: *secpol.msc*), *Security Settings->Account Policies->Password Policy* to see the new password policy:

© 2019 Microsoft Corporation. All rights reserved.

Using a **User account**, the evaluator has executed the script with no success and the following error was shown:

```
PS C:\Users\userFMT\Desktop\FMT> & '.\Test 4 - ConfigureMinimumPassword.ps1'
[I]: Setting up audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Configuring minimum password length...
System error 5 has occurred.

Access is denied.

[I]: Showing audit event...
Get-WinEvent : Could not retrieve information about the Security log. Error: Attempted to perform an unauthorized operation..
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : LogInfoUnavailable,Microsoft.PowerShell.Commands.GetWinEventCommand

Get-WinEvent : There is not an event log on the localhost computer that matches "Security".
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Security:String) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingLogsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

Get-WinEvent : The parameter is incorrect
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WinEvent], EventLogException
    + FullyQualifiedErrorId : System.Diagnostics.Eventing.Reader.EventLogException,Microsoft.PowerShell.Commands.GetWinEventCommand

[E]: There is no event log that matches with these conditions: eventId = 4739, newest = 2
[I]: Restoring audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.
```

33.2.4.4 Verdict

As the above results state, the minimum password length can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4**.

33.2.5 Test 10 - Configure Lockout Policy for Unsuccessful Authentication Attempts

The lockout policy for unsuccessful authentication will be configured during this test case.

© 2019 Microsoft Corporation. All rights reserved.

33.2.5.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.5.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 10 - LockoutPolicy.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 10 - LockoutPolicy.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 — Test 10
#

Write-Host "[I]: Getting actual value of lockout policy (before the change)..."
net accounts
Start-Sleep -s 1

Write-Host "[I]: Configuring minimum password length..."
net accounts /lockoutthreshold:3 /lockoutwindow:30 /lockoutduration:30

Write-Host "[I]: Getting actual value of lockout policy (after the change)..."
Start-Sleep -s 1
net accounts
```

- Observe the result.
- Log out, and log in again, but this time using the user account without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.5.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following image:

```
[I]: Getting actual value of lockout policy (before the change)...
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.

[I]: Configuring minimum password length...
The command completed successfully.

[I]: Getting actual value of lockout policy (after the change)...
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                None
Lockout threshold:                                    3
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.
```

Once the threshold has been modified, the information has been updated showing the new values.

Using a **User account**, the evaluator has executed the script with no success and the following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Getting actual value of lockout policy (before the change)...
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                None
Lockout threshold:                                    3
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.

[I]: Configuring minimum password length...
System error 5 has occurred.

Access is denied.

[I]: Getting actual value of lockout policy (after the change)...
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                None
Lockout threshold:                                    3
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.
```

33.2.5.4 Verdict

As the above results state, the lockout policy for unsuccessful authentication attempts can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 10** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 10**.

33.2.6 Test 11 - Configure Host-based Firewall

The host-based firewall will be configured during this test case.

33.2.6.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.6.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.

© 2019 Microsoft Corporation. All rights reserved.

- Run the script *Test 11 - ConfigureHost_basedFirewall.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 11 - ConfigureHost_basedFirewall.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 - Test 11
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"MPSSVC Rule-Level Policy Change" /success:enable
    /failure:enable
wevtutil cl security

Write-Host "[I]: Configuring host-based firewall event..."
$currentState = Get-NetFirewallProfile -Profile Public

if($currentState.Enabled -eq 'True'){
    Set-NetFirewallProfile -Profile Public -Enabled False
}else{
    Set-NetFirewallProfile -Profile Public -Enabled True
}
Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4950 1

Write-Host "[I]: Restoring audit rules..."
auditpol /set /subcategory:"MPSSVC Rule-Level Policy Change" /success:disable
    /failure:disable
```

- Observe the result.
- Log out, and log in again, but this time using the user account without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.6.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following image:

```
EventID             : 4950
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 1655
Category            : (13571)
CategoryNumber      : 13571
EntryType           : SuccessAudit
Message             : A Windows Firewall setting was changed.

                      Changed Profile:   Public

                      New Setting:
                         Type:   Enable Windows Defender Firewall
                         Value:  No
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {Public, Enable Windows Defender Firewall, No}
InstanceId          : 4950
TimeGenerated       : 12/6/2018 10:59:22 AM
TimeWritten         : 12/6/2018 10:59:22 AM
UserName            :
Site                :
Container           :
```

Firewall & network protection

Turn on Windows Firewall
Windows Firewall is turned off. Tap or click to turn it on.

Therefore, the changes were applied successfully.

Using a **User account**, the evaluator has executed the script with no success and the following error was shown:

```
[I]: Setting up audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

Failed to clear log security.
Access is denied.
[I]: Configuring host-based firewall event...
Set-NetFirewallProfile : Access is denied.
At C:\Users\userFMT\Desktop\FMT\Test 12 - ConfigureHostBasedFirewall.ps1:17 char:5
+     Set-NetFirewallProfile -Profile Public -Enabled True
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (MSFT_NetFirewal...rofile?Public"):root/standardci...FirewallProfile) [Set-NetFirewallProfile], CimException
    + FullyQualifiedErrorId : Windows System Error 5,Set-NetFirewallProfile

[I]: Showing audit event...
Get-EventLog : Requested registry access is not allowed.
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:6 char:14
+ ...    $event = Get-EventLog $eventType -InstanceId $eventID -Newest $new ...
+                ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-EventLog], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.GetEventLogCommand

[E]: There is no event log that matches with these conditions: eventId = 4950, newest = 1
[I]: Restoring audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.
```

33.2.6.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

As the above results state, the firewall configuration can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 11** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 11**.

33.2.7 Test 12 - Configure name/address of directory server to bind with

The evaluator will configure the name/address of directory server to bind with and remote management server during this test case.

33.2.7.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

To configure a domain controller, the applicable setup for this test is described in *Configuring Domain Controller and Certificate Authority* document.

Due to this feature is not included in Windows 10 Home Edition (October 2018 Update); this test is not applicable for the platforms with this OS edition.

Moreover, on the platforms with Windows Server 2019 operating system (listed below):

- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Once these platforms are joined to the domain, the domain controller can be configured as a remote management server. Therefore, they can receive management settings without using a MDM server (which is not supported on Windows Server 2019). The procedure used to join these platforms to the domain will be described in this test activity, whereas, the procedure to receive management settings will be explained in detail in the test activity: *Test 13 - Configure name/address of remote management server from which to receive management settings*.

33.2.7.2 Procedure

The steps required to perform this test depend on the type of the user account.

User with administrator rights:

- Log in as Administrator in the computer which is going to be joined to the domain.
- Run the script *Test 12 - ConfigureName-adress of directory server to bind with - Setup.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

© 2019 Microsoft Corporation. All rights reserved.

> .\Test 12 - ConfigureName-adress of directory server to bind with - Setup.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 12
#

Write—Host "[I]: Setting up audit rules..."
auditpol /set / category:'Account Management' /success:enable /failure:enable
wevtutil cl security
```

- Open the Settings application.

- Go to *Accounts*. After that, click-over *Access work or school* menu option.

- Now click the *Connect* button. A new window will be opened. Select in the Alternate actions submenu, the option: *Join this device to a local Active Directory Domain*.

© 2019 Microsoft Corporation. All rights reserved.

- On the new window, type the domain name you will join to:

© 2019 Microsoft Corporation. All rights reserved.

- After that, it a new prompt will ask for the credentials of the server domain computer account.

- On the domain-joined computer, a new user account will be created. Type its user name and account type in the following window:

- Finally, the computer will ask restart the PC to apply the new changes.

© 2019 Microsoft Corporation. All rights reserved.

- Once the computer has been joined to the domain, run the script *Test 12 - Configure name-adress of directory server to bind with - Audit.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 12 - Configure name-adress of directory server to bind with - Audit.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 12
#

Write-Host "[I]: Showing audit event..."
GetAuditLog System 3260 1
Auditpol /set /category:'Account Management' /success:disable /failure:disable
```

- Observe the result.
- Additionally, in the domain controller machine, run the script *Test 12 - Configure name-adress of directory server to bind with - Server.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 12 - Configure name-adress of directory server to bind with - Server.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 12
#

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4741 1
```

- Observe the result.

© 2019 Microsoft Corporation. All rights reserved.

On the platforms with Windows Server 2019 operating system, the following procedure shall be followed:

- Log in as Administrator in the computer which is going to be joined to the domain.
- Run the script *Test 12 - ConfigureName-adress of directory server to bind with - Setup.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 12 - ConfigureName-adress of directory server to bind with - Setup.ps1

- Open the *System Properties* console, (WIN +r, then type: *sysdm.cpl*). Click on the *Change...* button to join to a domain.

© 2019 Microsoft Corporation. All rights reserved.

- In the '*Member of*' section, enter the domain name in the textbox:

- After that, it a new prompt will ask for the credentials of the server domain computer account.

© 2019 Microsoft Corporation. All rights reserved.

- A new window will be opened with a welcome message to the new domain:

- Finally, the computer will ask restart the PC to apply the new changes.

© 2019 Microsoft Corporation. All rights reserved.

- Once the computer has been joined to the domain, run the script *Test 12 - Configure name-adress of directory server to bind with - Audit.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  ```
  .\Test 12 - Configure name-adress of directory server to bind with - Audit.ps1
  ```

- Observe the result.
- Additionally, in the domain controller machine, run the script *Test 12 - Configure name-adress of directory server to bind with - Server.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  ```
  .\Test 12 - Configure name-adress of directory server to bind with - Server.ps1
  ```

User without administrator rights:

- In the domain-joined computer, log out and log in again, but this time using a user without administrator rights.
- Repeat the previous steps, defined for users with administrator rights.
- Non-administrator accounts cannot join to a domain a red warning message will be shown pointing out this lack of provileges.

© 2019 Microsoft Corporation. All rights reserved.

- If the user tries clicks on the '*Connect*' button the *Join this device to a local Active Directory Domain* option is not available, the only configuration allowed for non-administrator users is joining into a MDM.

- On Windows Server, when the *System Properties* console is executed without administrator privileges, a prompt asking for administrator privileges is shown:

© 2019 Microsoft Corporation. All rights reserved.

33.2.7.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has followed the procedure described above for user with administrator rights and has obtained the following results:

```
Message               : This computer has been successfully joined to domain 'ee'.
Id                    : 3260
Version               :
Qualifiers            : 0
Level                 : 4
Task                  : 0
Opcode                :
Keywords              : 36028797018963968
RecordId              : 588
ProviderName          : Workstation
ProviderId            :
LogName               : System
ProcessId             :
ThreadId              :
MachineName           : SURFACEBOOK2
UserId                :
TimeCreated           : 12/6/2018 11:02:54 AM
ActivityId            :
RelatedActivityId     :
ContainerLog          : System
MatchedQueryIds       : {}
Bookmark              : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName      : Information
OpcodeDisplayName     : Info
TaskDisplayName       :
KeywordsDisplayNames  : {Classic}
Properties            : {System.Diagnostics.Eventing.Reader.EventProperty}
```

The screenshot shows the audit log generated in the domain-joined computer. In the domain controller computer, an audit log is generated when the computer has joined to the domain:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID            : 4741
MachineName        : WINDCAC.EE.LAB
Data               : {}
Index              : 32739
Category           : (13825)
CategoryNumber     : 13825
EntryType          : SuccessAudit
Message            : A computer account was created.

                     Subject:
                        Security ID:            S-1-5-21-1294932273-1509031496-868592660-500
                        Account Name:           Administrator
                        Account Domain:         EE
                        Logon ID:               0x9d9797

                     New Computer Account:
                        Security ID:            S-1-5-21-1294932273-1509031496-868592660-1109
                        Account Name:           SURFACELAPTOP2$
                        Account Domain:         EE

                     Attributes:
                        SAM Account Name:       SURFACELAPTOP2$
                        Display Name:           -
                        User Principal Name:    -
                        Home Directory:         -
                        Home Drive:             -
                        Script Path:            -
                        Profile Path:           -
                        User Workstations:      -
                        Password Last Set:      12/5/2018 2:16:49 PM
                        Account Expires:                %%1794
                        Primary Group ID:       515
                        AllowedToDelegateTo:    -
                        Old UAC Value:          0x0
                        New UAC Value:          0x80
                        User Account Control:
                                %%2087
                        User Parameters:        -
                        SID History:            -
                        Logon Hours:            %%1793
                        DNS Host Name:          SURFACELAPTOP2.EE.LAB
                        Service Principal Names:
                                HOST/SURFACELAPTOP2.EE.LAB
                                RestrictedKrbHost/SURFACELAPTOP2.EE.LAB
                                HOST/SURFACELAPTOP2
                                RestrictedKrbHost/SURFACELAPTOP2

                     Additional Information:
                        Privileges              -
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {SURFACELAPTOP2$, EE, S-1-5-21-1294932273-1509031496-868592660-1109, S-1-5-21-1294932273-1509031496-868592660-500...}
InstanceId         : 4741
TimeGenerated      : 12/5/2018 2:16:49 PM
TimeWritten        : 12/5/2018 2:16:49 PM
UserName           :
Site               :
Container          :
```

The evaluator has found that non-administrator accounts cannot join to a domain, therefore the joining to a domain option is not shown, the only configuration allowed for non-administrator users is joining into a MDM, as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

On Windows Server, when the *System Properties* console is executed without administrator privileges, a prompt asking for administrator privileges is shown:

© 2019 Microsoft Corporation. All rights reserved.

33.2.7.4 Verdict

As the above results state, a computer can only be joined to a domain using a user with administrator rights. When a user without administrator rights is used, a prompt asking for the administrator password is shown.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 12** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 12**.

33.2.8 Test 14 - Configure name/address of remote management server from which to receive management settings

The evaluator will configure the name/address of remote management server during this test case.

33.2.8.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

The applicable setup for this test is the same as the defined one for Test 1.

Additionally, a MDM Server running in another machine is necessary. For this test case, a MDM Server provided by the vendor has been used.

To enroll a device with a MDM, the evaluator has followed the applicable setup and steps described in the section **1.2.1.2 Enrolling with a MDM** of the ***Operational Guidance*** document.

Moreover, on the platforms with Windows Server 2019 operating system (listed below):

- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Once these platforms are joined to the domain, the domain controller can be configured as a remote management server. Therefore, they can receive management settings without using a MDM server (which is not supported on Windows Server 2019). The procedure used to join these platforms to the domain it is described in the test activity: *Test 12 - Configure name/address of directory server to bind with* , whereas, the procedure to receive management settings will be explained in detail in this test activity.

33.2.8.2 Procedure

The steps required to perform this test depend on the type of the user account.

User with administrator rights

- Log in as Administrator in the computer which is going to be enrolled to the MDM Server.
- Open the Settings application.

- Go to *Accounts*. After that, click-over *Access work or school* menu option and then click the *Connect* button.

© 2019 Microsoft Corporation. All rights reserved.

- A new window will be opened asking for the MDM credentials:

- The MDM Server will ask for its credentials:

© 2019 Microsoft Corporation. All rights reserved.

- Finally, after a few minutes, the device will be enrolled with the MDM Server.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

On the platforms with Windows Server 2019, once the platforms are domain-joined, the following steps shall be performed to configure on the remote management server to configure a remote setting through a GPO (*Group Policy Object*).

- On the domain computer, open the *Group Policy Management* tool (*WIN+R* and then type "*gpmc.msc*").
- Navigate through the forest and select the name of your domain controller. Right click over it and select *Create a GPO in this domain, and Link it here…* option.

© 2019 Microsoft Corporation. All rights reserved.

- Then, a new window will appear asking for a GPO name. Type any name, e.g. *Test FTP_SMF_EXT.1*, and click the *OK* button.

© 2019 Microsoft Corporation. All rights reserved.

- Afterwards, on the GPO created in the previous steps, right click on it and select the *Edit* option.

- A new window will be opened, navigate through the menus and select any computer or user management setting.

© 2019 Microsoft Corporation. All rights reserved.

Once the GPOs are configured, they are deployed automatically over all the Authenticated computers joined to the domain.

- In some cases, the policy is not successfully deployed. To force the deployment on the remote computer, execute the following command:

  ```
  gpupdate /force
  ```

- To verify on the domain-joined computer the correct deployment of a GPO, the following command can be executed:

  ```
  gpresult /r
  ```

© 2019 Microsoft Corporation. All rights reserved.

```
C:\Users\administrator>gpresult /r

Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0
© 2018 Microsoft Corporation. All rights reserved.

Created on ⌈12/⌉6/⌉2018 at 4:50:02 AM


RSOP data for EE\administrator on WIN10PRO : Logging Mode
---------------------------------------------------------

OS Configuration:            Member Workstation
OS Version:                  10.0.17763
Site Name:                   Default-First-Site-Name
Roaming Profile:             N/A
Local Profile:               C:\Users\administrator
Connected over a slow link?: No


COMPUTER SETTINGS
-----------------
    CN=WIN10PRO,CN=Computers,DC=EE,DC=LAB
    Last time Group Policy was applied: 12/6/2018 at 4:49:43 AM
    Group Policy was applied from:      WINDCAC.EE.LAB
    Group Policy slow link threshold:   500 kbps
    Domain Name:                        EE
    Domain Type:                        Windows 2008 or later

    Applied Group Policy Objects
    ----------------------------
        FPT_SRP_EXT.1.1_Test1and2

    The following GPOs were not applied because they were filtered out
    ------------------------------------------------------------------
        Local Group Policy
            Filtering:  Not Applied (Empty)

    The computer is a part of the following security groups
    -------------------------------------------------------
        BUILTIN\Administrators
        Everyone
```

User without administrator rights:

- In the MDM-enrolled computer, log out and log in again, but this time using a user without administrator rights.
- Repeat the previous steps defined for users with administrator rights.

© 2019 Microsoft Corporation. All rights reserved.

- A prompt asking for the password of the administrator account shall be shown. Since this user does not have permission, the operation could not be performed as shown in the following image:

- On Windows Server, when the *System Properties* console is executed without administrator privileges, a prompt asking for administrator privileges is shown:

© 2019 Microsoft Corporation. All rights reserved.

33.2.8.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has followed the procedure described above for the user with administrator rights and has successfully enrolled the device to the MDM Server. The following screenshots shows this fact:

© 2019 Microsoft Corporation. All rights reserved.

Settings

Managed by Test Corp

Connecting to work or school allows your organization to control some things on this device, such as settings and applications.

Support info

Email:
v-jirir@microsoft.com
Phone:
1-800-MDM-HELP
Website:
http://ccmdm.dynamic-dns.net

Device sync status

Syncing keeps security policies, network profiles, and managed applications up to date.
Last Attempted Sync:
The sync was successful
12/6/2018 11:44:26 AM

Sync

The evaluator has tried to enroll the device to the MDM server with the user without administrator rights. An error message was obtained when the evaluator has attempted to join the tested platform to the MDM Server, as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

On Windows Server, as described in the test activity *Test 12 - Configure name/address of directory server to bind with*, a user without administrator rights cannot join to a domain, therefore, the remote management server cannot be configured and no management settings can be applied.

33.2.8.4 Verdict

As the above results state, a computer can only be enrolled into a remote management server using a user with administrator rights. When a user without administrator rights is used, an insufficient privileges error message is shown.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 13** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 13**.

© 2019 Microsoft Corporation. All rights reserved.

33.2.9 Test 15 - Configure Audit Rules

The audit rules will be configured during this test case.

33.2.9.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.9.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 15 - ConfigureAuditRules.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 15 - ConfigureAuditRules.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 — Test 15
#

Write-Host "[I]: Clearing audit rules..."
auditpol /clear

Write-Host "[I]: Getting current audit rules..."
auditpol /get /category:*

Write-Host "[I]: Adding an audit rule..."
auditpol /set /subcategory:"System Integrity" /success:enable /failure:enable

Write-Host "[I]: Getting current audit rules..."
auditpol /get /category:*

Write-Host "[I]: Removing an audit rule..."
auditpol /set /subcategory:"System Integrity" /success:disable /failure:disable

Write-Host "[I]: Getting current audit rules..."
auditpol /get /category:*
```

- Observe the result.
- Log out, and log in again, but this time, using a user account with no administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

© 2019 Microsoft Corporation. All rights reserved.

33.2.9.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the steps listed above successfully using an **Administrator account.**

First of all, the audit for System Integrity subcategory is disabled:

```
[I]: Getting current audit rules...
System audit policy

Category/Subcategory                    Setting
System
  Security System Extension             No Auditing
  System Integrity                      No Auditing
  IPsec Driver                          No Auditing
  Other System Events                   No Auditing
  Security State Change                 No Auditing
Logon/Logoff
  Logon                                 No Auditing
  Logoff                                No Auditing
  Account Lockout                       No Auditing
  IPsec Main Mode                       No Auditing
  IPsec Quick Mode                      No Auditing
  IPsec Extended Mode                   No Auditing
  Special Logon                         No Auditing
  Other Logon/Logoff Events             No Auditing
  Network Policy Server                 No Auditing
  User / Device Claims                  No Auditing
  Group Membership                      No Auditing
Object Access
  File System                           No Auditing
  Registry                              No Auditing
  Kernel Object                         No Auditing
```

© 2019 Microsoft Corporation. All rights reserved.

```
  SAM                                        No Auditing
  Certification Services                     No Auditing
  Application Generated                      No Auditing
  Handle Manipulation                        No Auditing
  File Share                                 No Auditing
  Filtering Platform Packet Drop             No Auditing
  Filtering Platform Connection              No Auditing
  Other Object Access Events                 No Auditing
  Detailed File Share                        No Auditing
  Removable Storage                          No Auditing
  Central Policy Staging                     No Auditing
Privilege Use
  Non Sensitive Privilege Use                No Auditing
  Other Privilege Use Events                 No Auditing
  Sensitive Privilege Use                    No Auditing
Detailed Tracking
  Process Creation                           No Auditing
  Process Termination                        No Auditing
  DPAPI Activity                             No Auditing
  RPC Events                                 No Auditing
  Plug and Play Events                       No Auditing
  Token Right Adjusted Events                No Auditing
Policy Change
  Audit Policy Change                        No Auditing
  Authentication Policy Change               No Auditing
  Authorization Policy Change                No Auditing
  MPSSVC Rule-Level Policy Change            No Auditing
  Filtering Platform Policy Change           No Auditing
  Other Policy Change Events                 No Auditing
Account Management
  Computer Account Management                No Auditing
  Security Group Management                  No Auditing
  Distribution Group Management              No Auditing
  Application Group Management               No Auditing
  Other Account Management Events            No Auditing
  User Account Management                    No Auditing
DS Access
  Directory Service Access                   No Auditing
  Directory Service Changes                  No Auditing
  Directory Service Replication              No Auditing
  Detailed Directory Service Replication     No Auditing
Account Logon
  Kerberos Service Ticket Operations         No Auditing
  Other Account Logon Events                 No Auditing
  Kerberos Authentication Service            No Auditing
  Credential Validation                      No Auditing
```

After that, the audit for this category is enabled:

```
[I]: Adding an audit rule...
The command was successfully executed.

[I]: Getting current audit rules...
```

© 2019 Microsoft Corporation. All rights reserved.

```
System audit policy

Category/Subcategory                      Setting
System
  Security System Extension               No Auditing
  System Integrity                        Success and Failure
  IPsec Driver                            No Auditing
  Other System Events                     No Auditing
  Security State Change                   No Auditing
Logon/Logoff
  Logon                                   No Auditing
  Logoff                                  No Auditing
  Account Lockout                         No Auditing
  IPsec Main Mode                         No Auditing
  IPsec Quick Mode                        No Auditing
  IPsec Extended Mode                     No Auditing
  Special Logon                           No Auditing
  Other Logon/Logoff Events               No Auditing
  Network Policy Server                   No Auditing
  User / Device Claims                    No Auditing
  Group Membership                        No Auditing
Object Access
  File System                             No Auditing
  Registry                                No Auditing
  Kernel Object                           No Auditing
  SAM                                     No Auditing
  Certification Services                  No Auditing
  Application Generated                   No Auditing
  Handle Manipulation                     No Auditing
  File Share                              No Auditing
  Filtering Platform Packet Drop          No Auditing
  Filtering Platform Connection           No Auditing
  Other Object Access Events              No Auditing
  Detailed File Share                     No Auditing
  Removable Storage                       No Auditing
  Central Policy Staging                  No Auditing
Privilege Use
  Non Sensitive Privilege Use             No Auditing
  Other Privilege Use Events              No Auditing
  Sensitive Privilege Use                 No Auditing
Detailed Tracking
  Process Creation                        No Auditing
  Process Termination                     No Auditing
  DPAPI Activity                          No Auditing
  RPC Events                              No Auditing
  Plug and Play Events                    No Auditing
  Token Right Adjusted Events             No Auditing
Policy Change
  Audit Policy Change                     No Auditing
  Authentication Policy Change            No Auditing
  Authorization Policy Change             No Auditing
  MPSSVC Rule-Level Policy Change         No Auditing
```

© 2019 Microsoft Corporation. All rights reserved.

```
  Filtering Platform Policy Change          No Auditing
  Other Policy Change Events                No Auditing
Account Management
  Computer Account Management               No Auditing
  Security Group Management                 No Auditing
  Distribution Group Management             No Auditing
  Application Group Management              No Auditing
  Other Account Management Events           No Auditing
  User Account Management                   No Auditing
DS Access
  Directory Service Access                  No Auditing
  Directory Service Changes                 No Auditing
  Directory Service Replication             No Auditing
  Detailed Directory Service Replication    No Auditing
Account Logon
  Kerberos Service Ticket Operations        No Auditing
  Other Account Logon Events                No Auditing
  Kerberos Authentication Service           No Auditing
  Credential Validation                     No Auditing
```

Finally, once the audit is disabled the audit log returns to its original state:

```
[I]: Removing an audit rule...
The command was successfully executed.

[I]: Getting current audit rules...
System audit policy

Category/Subcategory                        Setting
System
  Security System Extension                 No Auditing
  System Integrity                          No Auditing
  IPsec Driver                              No Auditing
  Other System Events                       No Auditing
  Security State Change                     No Auditing
Logon/Logoff
  Logon                                     No Auditing
  Logoff                                    No Auditing
  Account Lockout                           No Auditing
  IPsec Main Mode                           No Auditing
  IPsec Quick Mode                          No Auditing
  IPsec Extended Mode                       No Auditing
  Special Logon                             No Auditing
  Other Logon/Logoff Events                 No Auditing
  Network Policy Server                     No Auditing
  User / Device Claims                      No Auditing
  Group Membership                          No Auditing
Object Access
  File System                               No Auditing
  Registry                                  No Auditing
  Kernel Object                             No Auditing
  SAM                                       No Auditing
  Certification Services                    No Auditing
```

© 2019 Microsoft Corporation. All rights reserved.

```
  Application Generated                    No Auditing
  Handle Manipulation                      No Auditing
  File Share                               No Auditing
  Filtering Platform Packet Drop           No Auditing
  Filtering Platform Connection            No Auditing
  Other Object Access Events               No Auditing
  Detailed File Share                      No Auditing
  Removable Storage                        No Auditing
  Central Policy Staging                   No Auditing
Privilege Use
  Non Sensitive Privilege Use              No Auditing
  Other Privilege Use Events               No Auditing
  Sensitive Privilege Use                  No Auditing
Detailed Tracking
  Process Creation                         No Auditing
  Process Termination                      No Auditing
  DPAPI Activity                           No Auditing
  RPC Events                               No Auditing
  Plug and Play Events                     No Auditing
  Token Right Adjusted Events              No Auditing
Policy Change
  Audit Policy Change                      No Auditing
  Authentication Policy Change             No Auditing
  Authorization Policy Change              No Auditing
  MPSSVC Rule-Level Policy Change          No Auditing
  Filtering Platform Policy Change         No Auditing
  Other Policy Change Events               No Auditing
Account Management
  Computer Account Management              No Auditing
  Security Group Management                No Auditing
  Distribution Group Management            No Auditing
  Application Group Management             No Auditing
  Other Account Management Events          No Auditing
  User Account Management                  No Auditing
DS Access
  Directory Service Access                 No Auditing
  Directory Service Changes                No Auditing
  Directory Service Replication            No Auditing
  Detailed Directory Service Replication   No Auditing
Account Logon
  Kerberos Service Ticket Operations       No Auditing
  Other Account Logon Events               No Auditing
  Kerberos Authentication Service          No Auditing
  Credential Validation                    No Auditing
```

Using a **User account**, the evaluator has executed the commands with no success and the following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Clearing audit rules...
Are you sure (Press N to cancel or any other key to continue)?y
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Getting current audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Adding an audit rule...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Getting current audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Removing an audit rule...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Getting current audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.
```

33.2.9.4 Verdict

As the above results state, the audit rules can only be configured using a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 15** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 15**.

33.2.10 Test 16 - Configure Name/Address of Network Time Server

The name and address of a Network Time Server will be configured during this test case

33.2.10.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.10.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.

© 2019 Microsoft Corporation. All rights reserved.

- Run the script *Test 16 - ConfigureNameAddressOfNetworkTimeServer.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 16 - ConfigureNameAddressOfNetworkTimeServer.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1

#
# Test FMT_MOF_EXT.1 — Test 16
#

Write-Host "[I]: Starting the time service..."
net start w32time

Write-Host "[I]: Clearing log..."
wevtutil cl system

Write-Host "[I]: Getting actual configuration"
w32tm /query /configuration

Write-Host "[I]: Configuring the Name Adress of Network Time Server..."
w32tm /config /manualpeerlist:pool.ntp.org /syncfromflags:manual /update
#Default time.windows.com

Write-Host "[I]: Setting time updates to manual..."
w32tm /config /syncfromflags:manual /update

Write-Host "[I]: Getting new configuration"
w32tm /query /configuration

Write-Host "[I]: Performing a manual time update..."
w32tm /resync /force
Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog System 37 1
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.10.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and the address of time server and information about the name of the new network time server are shown. First is the actual address of the Network Time Server, afterwards, the new address.

© 2019 Microsoft Corporation. All rights reserved.

```
NtpServer (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 0 (Local)
InputProvider: 0 (Local)


[I]: Configuring the Name Adress of Network Time Server...
The command completed successfully.
[I]: Setting time updates to manual...
The command completed successfully.
[I]: Getting new configuration
[Configuration]

EventLogFlags: 2 (Local)
AnnounceFlags: 10 (Local)
TimeJumpAuditOffset: 28800 (Local)
MinPollInterval: 10 (Local)
MaxPollInterval: 15 (Local)
MaxNegPhaseCorrection: 54000 (Local)
MaxPosPhaseCorrection: 54000 (Local)
MaxAllowedPhaseOffset: 1 (Local)

FrequencyCorrectRate: 4 (Local)
PollAdjustFactor: 5 (Local)
LargePhaseOffset: 50000000 (Local)
SpikeWatchPeriod: 900 (Local)
LocalClockDispersion: 10 (Local)
HoldPeriod: 5 (Local)
PhaseCorrectRate: 1 (Local)
UpdateInterval: 360000 (Local)


[TimeProviders]

NtpClient (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 1 (Local)
InputProvider: 1 (Local)
AllowNonstandardModeCombinations: 1 (Local)
ResolvePeerBackoffMinutes: 15 (Local)
ResolvePeerBackoffMaxTimes: 7 (Local)
CompatibilityFlags: 2147483648 (Local)
EventLogFlags: 1 (Local)
LargeSampleSkew: 3 (Local)
SpecialPollInterval: 32768 (Local)
Type: NTP (Local)
NtpServer: pool.ntp.org (Local)

NtpServer (Local)
DllName: C:\Windows\system32\w32time.dll (Local)
Enabled: 0 (Local)
InputProvider: 0 (Local)


[I]: Performing a manual time update...
Sending resync command to local computer
The command completed successfully.
```

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the changes were applied successfully. Additionally, the evaluator has reviewed the audit log generated after the *NTP* server was modified:

```
Message              : NTP Client observed a change peer reachability. Ntp Client is now receiving time data from the following NTP Servers: pool.ntp.org
                       (ntp.m|0x0|0.0.0.0:123->216.229.0.50:123);. System Tick Count: 679218.
                       For more information, see https://go.microsoft.com/fwlink/?linkid=845961.
Id                   : 264
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 0
Opcode               : 0
Keywords             : 4611686018427387904
RecordId             : 46
ProviderName         : Microsoft-Windows-Time-Service
ProviderId           : 06edcfeb-0fd0-4e53-acca-a6f8bbf81bcb
LogName              : Microsoft-Windows-Time-Service/Operational
ProcessId            : 2748
ThreadId             : 4808
MachineName          : SURFACEBOOK2
UserId               : S-1-5-19
TimeCreated          : 12/6/2018 11:47:11 AM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-Time-Service/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty}
```

Using a **User account**, the evaluator has executed the script with no success and the following error was shown:

```
[I]: Starting the time service...
System error 5 has occurred.

Access is denied.

[I]: Clearing log...
Failed to clear log system.
Access is denied.
[I]: Getting actual configuration
The following error occurred: Access is denied. (0x80070005)
[I]: Configuring the Name Adress of Network Time Server...
The following error occurred: Access is denied. (0x80070005)
[I]: Setting time updates to manual...
The following error occurred: Access is denied. (0x80070005)
[I]: Getting new configuration
The following error occurred: Access is denied. (0x80070005)
[I]: Performing a manual time update...
Sending resync command to local computer
The following error occurred: Access is denied. (0x80070005)
```

33.2.10.4 Verdict

As the above results state, the name or address of the network time server can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 16** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 16**.

© 2019 Microsoft Corporation. All rights reserved.

33.2.11 Test 17 - Enable/Disable Automatic Software Update

The automatic software update will be enabled and disabled during this test case.

33.2.11.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.11.2 Procedure

To complete this test, the following steps shall be performed:

User with administrator rights:

- Log in using the administrator account.
- Run the script *Test 17 - Enable&DisableAutomaticSoftwareUpdate.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 17 - Enable&DisableAutomaticSoftwareUpdate.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1
. .\common\AuditRegistryKeys.ps1

#
# Test FMT_MOF_EXT.1 — Test 17
#

#    [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\W indows\WindowsUpdate]
#    [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\W indows\WindowsUpdate\AU]
# "NoAutoUpdate"=dword:00000001

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Registry"   /success:enable /failure:enable

Write-Host "[I]: Enabling/Disabling Automatic   Windows Updates..."
$pathFolder =   "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\M
    icrosoft\Windows\WindowsUpdate"
$pathMissing =  "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\M
    icrosoft\Windows\"
$folderName = "WindowsUpdate"
$pathSubFolder =    "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\M
    icrosoft\Windows\WindowsUpdate\AU"
$subpathMissing =   "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\M
    icrosoft\Windows\WindowsUpdate\"
$subfolderName = "AU"
$keyName = "NoAutoUpdate"

# Disable Automatic Windows Updates
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Disabling Automatic Windows Updates"

If (!(Test-Path -Path $pathFolder)){
    New-Item -Path $pathMissing -Name $folderName -Force
    If (!(Test-Path -Path $pathSubFolder)){
        New-Item -Path $subpathMissing -Name $subfolderName -Force
        New-ItemProperty -Path $pathsubFolder -Name $keyName -Value "1"
        -PropertyType DWORD
    } else {
       New-ItemProperty -Path $pathsubFolder -Name $keyName -Value "1"
       -PropertyType DWORD
    }
} else {
    If (!(Test-Path -Path $pathSubFolder)){
        New-Item -Path $subpathMissing -Name $subfolderName -Force
        New-ItemProperty -Path $pathsubFolder -Name $keyName -Value "1"
        -PropertyType DWORD
    } else {
        Set-ItemProperty -Path $pathsubFolder -Name $keyName -Value "1"
    }
}
Start-Process 'gpupdate.exe' -ArgumentList '/force' -WorkingDirectory
    "C:\Windows\System32"
Start-Sleep -s 1

AddAuditToRegKey $pathSubFolder "Users" "SetValue" "ContainerInherit" "none"
    "Success, Failure"

# [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\W indows\WindowsUpdate\AU]
# "NoAutoUpdate"=dword:00000000
# "AUOptions"=dword:00000003
# "ScheduledInstallDay"=dword:00000000
# "ScheduledInstallTime"=dword:00000003

$keyNoAutoUpdate = "NoAutoUpdate"
$keyAUOptions = "AUOptions"
$keyScheduledInstallDay = "ScheduledInstallDay"
$keyScheduledInstallTime = "ScheduledInstallTime"

# Enabling Automatic Windows Updates
Write-Host "[I]: Enabling Automatic Windows Updates"
If (!(Test-Path -Path $pathSubFolder)){
    New-Item -Path $subpathMissing -Name $subfolderName -Force
    Set-ItemProperty -Path $pathsubFolder -Name $keyNoAutoUpdate -Value "0"
    New-ItemProperty -Path $pathsubFolder -Name $keyAUOptions -Value "3"
    -PropertyType DWORD
    New-ItemProperty -Path $pathsubFolder -Name $keyScheduledInstallDay
    -Value "0" -PropertyType DWORD
    New-ItemProperty -Path $pathsubFolder -Name $keyScheduledInstallTime
    -Value "3" -PropertyType DWORD
} else {
    Set-ItemProperty -Path $pathsubFolder -Name $keyNoAutoUpdate -Value "0"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    New-ItemProperty -Path $pathsubFolder -Name     $keyAUOptions -Value "3"
    -PropertyType DWORD
    New-ItemProperty -Path $pathsubFolder -Name $keyScheduledInstallDay
-Value "0" -PropertyType DWORD
New-ItemProperty -Path $pathsubFolder -Name $keyScheduledInstallTime
-Value "3" -PropertyType DWORD
}

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4657 1

Write-Host "[I]: Restoring audit rules..."
RemoveAuditToRegKey $pathSubFolder "Users"  "SetValue" "ContainerInherit"
"none" "Success,Failure"
auditpol /set /subcategory:"Registry"   /success:disable /failure:disable
Remove-Item -Path $pathFolder -Recurse
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.11.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and has enable and disabled the automatic software updates. An audit entry was generated:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID             : 4657
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 2173
Category            : (12801)
CategoryNumber      : 12801
EntryType           : SuccessAudit
Message             : A registry value was modified.

                      Subject:
                         Security ID:            S-1-5-21-473433637-2525778249-1789179070-500
                         Account Name:           Administrator
                         Account Domain:         SURFACEBOOK2
                         Logon ID:               0x426a9

                      Object:
                         Object Name:            \REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU
                         Object Value Name:      ScheduledInstallTime
                         Handle ID:              0xfa0
                         Operation Type:         %%1904

                      Process Information:
                         Process ID:             0x1488
                         Process Name:           C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

                      Change Information:
                         Old Value Type:         -
                         Old Value:              -
                         New Value Type:         %%1876
                         New Value:              3
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {S-1-5-21-473433637-2525778249-1789179070-500, Administrator, SURFACEBOOK2, 0x426a9...}
InstanceId          : 4657
TimeGenerated       : 12/6/2018 11:47:55 AM
TimeWritten         : 12/6/2018 11:47:55 AM
UserName            :
Site                :
Container           :
```

Therefore, the changes were applied successfully.

Using a **User account**, the evaluator has executed the script with no success and the following error was shown:

```
[I]: Setting up audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Enabling/Disabling Automatic Windows Updates...
[I]: Disabling Automatic Windows Updates
Set-ItemProperty : Requested registry access is not allowed.
At C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT\Test 18 - Enable&DisableAutomaticSoftwareUpdate.ps1:40 char:9
+         Set-ItemProperty -Path $pathsubFolder -Name $keyName -Value " ...
+         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...indowsUpdate\AU:String) [Set-ItemProperty], SecurityExcep
   tion
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.SetItemPropertyCommand

Set-Acl : Requested registry access is not allowed.
At C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT\common\AuditRegistryKeys.ps1:19 char:19
+     $RegKey_ACL | Set-Acl $key
+                   ~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...indowsUpdate\AU:String) [Set-Acl], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.SetAclCommand

[I]: Enabling Automatic Windows Updates
Set-ItemProperty : Requested registry access is not allowed.
At C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT\Test 18 - Enable&DisableAutomaticSoftwareUpdate.ps1:68 char:5
+     Set-ItemProperty -Path $pathsubFolder -Name $keyNoAutoUpdate -Val ...
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...indowsUpdate\AU:String) [Set-ItemProperty], SecurityExcep
   tion
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.SetItemPropertyCommand

New-ItemProperty : Requested registry access is not allowed.
At C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT\Test 18 - Enable&DisableAutomaticSoftwareUpdate.ps1:69 char:5
+     New-ItemProperty -Path $pathsubFolder -Name $keyAUOptions -Value  ...
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...indowsUpdate\AU:String) [New-ItemProperty], SecurityExcep
   tion
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.NewItemPropertyCommand

New-ItemProperty : Requested registry access is not allowed.
At C:\Users\SecEvalAdmin\Desktop\TestingWindowsCU\FMT\Test 18 - Enable&DisableAutomaticSoftwareUpdate.ps1:70 char:5
+     New-ItemProperty -Path $pathsubFolder -Name $keyScheduledInstallD ...
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...indowsUpdate\AU:String) [New-ItemProperty], SecurityExcep
   tion
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.NewItemPropertyCommand
```

Moreover, the evaluator has tried to access the *Group Policy Editor* on the account without administrator rights. The following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

33.2.11.4 Verdict

As the above results state, the automatic software update can only be enabled or disabled using a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 17** requirements established in the assurance activity section.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to **Test 17**.

33.2.12 Test 18 - Configure Wi-Fi interface

The Wi-Fi interface will be configured during this test case.

33.2.12.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

To perform this test, the platforms should have a Wi-Fi interface available.

33.2.12.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 18 - Enable&DisableWi-FiInterface.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 18 - Enable&DisableWi-FiInterface.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 — Test 18
#

Write-Host "[I]: Getting list of network adapters..."
Get-NetAdapter -Name Wi-Fi

Write-Host "[I]: Disabling Local Area Network Interface..."
Disable-NetAdapter -Name Wi-Fi -Confirm:$false

Write-Host "[I]: Checking Local Area Network Interface status..."
Get-NetAdapter -Name Wi-Fi

Write-Host "[I]: Enabling Local Area Network Interface..."
Enable-NetAdapter -Name Wi-Fi -Confirm:$false
Start-Sleep 4 # Waiting for the network interface to be enabled

Write-Host "[I]: Checking Local Area Network Interface status..."
Get-NetAdapter -Name Wi-Fi
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.

© 2019 Microsoft Corporation. All rights reserved.

- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.12.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account**. First, the current state of the Wi-Fi interface was shown, then the interface was disabled. Finally, it has been enabled again.

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4       360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
[I]: Disabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disabled     C4-9D-ED-0E-A8-D4         0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
[I]: Enabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4         0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
```

Therefore, the changes were applied successfully.

Using a **User account**, the evaluator has tried to execute the same script, nevertheless, the following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4       360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
[I]: Disabling Local Area Network Interface...
Disable-NetAdapter : Access is denied.
At C:\Users\userFMT\Desktop\FMT\Test 19 - Enable&DisableWi-FiInterface.ps1:9 char:1
+ Disable-NetAdapter -Name Wi-Fi -Confirm:$false
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (MSFT_NetAdapter..."SURFACEBOOK2"):ROOT/StandardCimv2/MSFT_NetAdapter) [Disable-NetAdapter], CimException
    + FullyQualifiedErrorId : Windows System Error 5,Disable-NetAdapter

[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4       360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
[I]: Enabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4       360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5         3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34         1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D       600 Mbps
```

33.2.12.4 Verdict

As the above results state, a Wi-Fi interface can only be enabled or disabled using a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 18** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 18**.

33.2.13 Test 19 - Enable/Disable Bluetooth interface

The Bluetooth interface will be enabled and disabled during this test case.

33.2.13.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

In addition, to perform this test, the platforms should have a Bluetooth interface available.

33.2.13.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 19 - Enable&DisableBluetoothInterface.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 19 - Enable&DisableBluetoothInterface.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 — Test 19
#
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Getting list of network adapters..."
Get-NetAdapter -Name 'Bluetooth Network Connection'

Write-Host "[I]: Disabling Local Area Network Interface..."
Disable-NetAdapter -Name 'Bluetooth Network Connection' -Confirm:$false

Write-Host "[I]: Checking Local Area Network Interface status..."
Get-NetAdapter -Name 'Bluetooth Network Connection'

Write-Host "[I]: Enabling Local Area Network Interface..."
Enable-NetAdapter -Name 'Bluetooth Network Connection' -Confirm:$false
Start-Sleep 4 # Waiting for the network interface to be enabled

Write-Host "[I]: Checking Local Area Network Interface status..."
Get-NetAdapter -Name 'Bluetooth Network Connection'
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.13.3 Results

This test can only be performed in those platforms which have Bluetooth interface.

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account**. First, the current state of the Bluetooth interface was shown, then the interface was disabled. Finally, it has been enabled again.

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Disabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disabled     C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Enabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
```

Therefore, the changes were applied successfully.

Using a **User account**, the evaluator has tried to execute the same script. Nevertheless, the following error was shown:

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Disabling Local Area Network Interface...
Disable-NetAdapter : Access is denied.
At C:\Users\userFMT\Desktop\FMT\Test 20 - Enable&DisableBluetoothInterface.ps1:9 char:1
+ Disable-NetAdapter -Name 'Bluetooth Network Connection' -Confirm:$fal ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (MSFT_NetAdapter..."SURFACEBOOK2"):ROOT/StandardCimv2/MSFT_NetAdapter) [Disable-NetAdapter], CimException
    + FullyQualifiedErrorId : Windows System Error 5,Disable-NetAdapter

[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Enabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
```

33.2.13.4 Verdict

As the above results state, a Bluetooth interface can only be enabled or disabled using a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 19** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 19**.

33.2.14 Test 20 - Configure USB interfaces

The USB interfaces will be configured during this test case.

33.2.14.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.14.2 Procedure

© 2019 Microsoft Corporation. All rights reserved.

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 20 - ConfigureUSBinterface.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 20 - ConfigureUSBinterface.ps1

- The script source code is as follows:

```
Write-Host "Showing content of the USB device..."
ls D:
$usbId = Get-PnpDevice -Class DiskDrive -Status OK | Where {$_.FriendlyName
    -like "*DataTraveler*"} | select InstanceID

#disable usb
Write-Host "Configuring USB interfaces: USB interface disabled..."
Disable-PnpDevice -InstanceId $usbId.InstanceID -Confirm:$false
Write-Host "Showing content of the USB device..."
ls D:

#enable usb
Write-Host "Configuring USB interfaces: USB interface enabled..."
Enable-PnpDevice -InstanceId $usbId.InstanceID -Confirm:$false
start-sleep 5
Write-Host "Showing content of the USB device..."
ls D:
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.14.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account**, showing the contents of the USB device before and after the USB configuration was made:

```
Showing content of the USB device...


    Directory: E:\


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       12/20/2018     15:03                Test
Configuring USB interfaces: USB interface disabled...
Showing content of the USB device...
ls : Cannot find drive. A drive with the name 'E' does not exist.
At C:\Users\evaluador\Desktop\Test 20 - ConfigureUSBinterface.ps1:9 char:1
+ ls E:
+ ~~~~~
    + CategoryInfo          : ObjectNotFound: (E:String) [Get-ChildItem], DriveNotFoundException
    + FullyQualifiedErrorId : DriveNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand

Configuring USB interfaces: USB interface enabled...
Showing content of the USB device...
d-----       12/20/2018     15:03                Test
```

Once the USB interface was configured, the USB drive cannot be found since it was disabled, therefore the changes were applied successfully.

Using a **User account**, the evaluator has tried to execute the same script, nevertheless, the following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

```
Showing content of the USB device...


    Directory: E:\


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       12/20/2018     15:03                Test
Configuring USB interfaces: USB interface disabled...
Disable-PnpDevice : Generic failure
At C:\Users\evaluador\Desktop\Test 20 - ConfigureUSBinterface.ps1:7 char:1
+ Disable-PnpDevice -InstanceId $usbId.InstanceID -Confirm:$false
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (Win32_PnPEntity...OD_DATATRAV...):ROOT\cimv2\Win32_PnPEntity) [Di
   sable-PnpDevice], CimException
    + FullyQualifiedErrorId : HRESULT 0x80041001,Disable-PnpDevice

Showing content of the USB device...
d-----       12/20/2018     15:03                Test
Configuring USB interfaces: USB interface enabled...
Showing content of the USB device...
d-----       12/20/2018     15:03                Test
```

Moreover, the evaluator has tried to access the *Group Policy Editor* on the account without administrator rights. The following error was shown:

© 2019 Microsoft Corporation. All rights reserved.

33.2.14.4 Verdict

As the above result state, the configuration of USB interfaces can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 20** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 20**.

33.2.15 Test 21 - Enable/Disable Local Area Network interface

The Local Area Network interface will be enabled and disabled during this test case.

33.2.15.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.15.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

© 2019 Microsoft Corporation. All rights reserved.

- Log in using the administrator account.
- Run the script *Test 21 - Enable&DisableEthernetInterface.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 21 - Enable&DisableEthernetInterface.ps1

- The script source code is as follows:

```
#
# Test FMT_MOF_EXT.1 — Test 21
#

Write—Host "[I]: Getting list of network adapters..."
Get—NetAdapter

Write—Host "[I]: Disabling Local Area Network Interface..."
Disable—NetAdapter —Name Ethernet —Confirm:$false

Write—Host "[I]: Checking Local Area Network Interface status..."
Get—NetAdapter

Write—Host "[I]: Enabling Local Area Network Interface..."
Enable—NetAdapter —Name Ethernet —Confirm:$false
Start—Sleep 4 # Waiting for the network interface to be enabled

Write—Host "[I]: Checking Local Area Network Interface status..."
Get—NetAdapter
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.15.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account**. First, the current state of the Local Area Network interface was shown, then the interface was disabled. Finally, it has been enabled again.

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Disabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Disabled     50-3E-AA-63-C3-34          0 bps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Enabling Local Area Network Interface...
[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Disconnected C4-9D-ED-0E-A8-D4          0 bps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
```

Therefore, the changes were applied successfully.

Using a **User account**, the evaluator has tried to execute the same script, nevertheless, the following error was shown:

```
[I]: Getting list of network adapters...

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Disabling Local Area Network Interface...
Disable-NetAdapter : No MSFT_NetAdapter objects found with property 'Name' equal to 'NIC3'.  Verify the value of the property and retry.
At C:\Users\userFMT\Desktop\FMT\Test 22 - Enable&DisableEthernetInterface.ps1:9 char:1
+ Disable-NetAdapter -Name 'NIC3' -Confirm:$false
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (NIC3:String) [Disable-NetAdapter], CimJobException
    + FullyQualifiedErrorId : CmdletizationQuery_NotFound_Name,Disable-NetAdapter

[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        360 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
[I]: Enabling Local Area Network Interface...
Enable-NetAdapter : No MSFT_NetAdapter objects found with property 'Name' equal to 'NIC3'.  Verify the value of the property and retry.
At C:\Users\userFMT\Desktop\FMT\Test 22 - Enable&DisableEthernetInterface.ps1:15 char:1
+ Enable-NetAdapter -Name 'NIC3' -Confirm:$false
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (NIC3:String) [Enable-NetAdapter], CimJobException
    + FullyQualifiedErrorId : CmdletizationQuery_NotFound_Name,Enable-NetAdapter

[I]: Checking Local Area Network Interface status...
Wi-Fi                     Marvell AVASTAR Wireless-AC Network ...      11 Up           C4-9D-ED-0E-A8-D4        400 Mbps
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       8 Disconnected C4-9D-ED-0E-A8-D5          3 Mbps
Ethernet 2                Realtek USB GbE Family Controller             5 Up           50-3E-AA-63-C3-34          1 Gbps
Local Area Connection     Xbox Wireless Adapter for Windows             4 Up           62-45-B5-12-A3-0D        600 Mbps
```

33.2.15.4 Verdict

As the above results state, a Local Area Network interface can only be configured by a user with administrator rights.

© 2019 Microsoft Corporation. All rights reserved.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 21** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 21**.

33.2.16 Test 22 - Manage Windows Diagnostics settings

Windows Diagnostics settings will be configured during this test case.

33.2.16.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.16.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 22 - ManageWindowsDiagnosticSettings.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 22 - ManageWindowsDiagnosticSettings.ps1

- The script source code is as follows:

```
. .\common\AuditLog.ps1
. .\common\AuditRegistryKeys.ps1

#
# Test FMT_MOF_EXT.1 - Test 22
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Registry"   /success:enable /failure:enable

Write-Host "[I]: Configuring Diagnostic settings..."
$pathFolder =
    "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion
\Policies\DataCollection"
$keyName = "AllowTelemetry"
$key = Get-ItemProperty -Path $pathFolder -Name $keyName -ErrorAction
    SilentlyContinue

$auditKey =
    "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\"
AddAuditToRegKey $auditKey "Users" "SetValue"   "ContainerInherit" "none"
    "Success,Failure"

If ($key.AllowTelemetry -eq 3){
```

© 2019 Microsoft Corporation. All rights reserved.

```
    Set-ItemProperty -Path $pathFolder -Name    $keyName -Value "1"
    Write-Host  "[I]: Basic mode selected"
}Elseif ($key.AllowTelemetry -eq 1) {
    Set-ItemProperty -Path $pathFolder -Name    $keyName -Value "3"
    Write-Host  "[I]: Full mode selected"
}
    Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4657 1

Write-Host "[I]: Restoring audit rules..."
RemoveAuditToRegKey $auditKey "Users" "SetValue"   "ContainerInherit" "none"
    "Success,Failure"
auditpol /set /subcategory:"Registry"   /success:disable /failure:disable
```

- Observe the result.
- Log out and log in as local user, but this time using a user without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.16.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following images:

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Setting up audit rules...
The command was successfully executed.
[I]: Configuring Diagnostic settings...
[I]: Basic mode selected
[I]: Showing audit event...

-------------------------------------------------------


EventID            : 4657
MachineName        : ENTERPRISE-1809
Data               : {}
Index              : 19035
Category           : (12801)
CategoryNumber     : 12801
EntryType          : SuccessAudit
Message            : A registry value was modified.

                     Subject:
                        Security ID:            S-1-5-21-4250873927-3395330480-3391837428-1001
                        Account Name:           evaluador
                        Account Domain:         ENTERPRISE-1809
                        Logon ID:               0x1d66c

                     Object:
                        Object Name:            \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection
                        Object Value Name:      AllowTelemetry
                        Handle ID:              0xae4
                        Operation Type:         %%1905

                     Process Information:
                        Process ID:             0x1a44
                        Process Name:           C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

                     Change Information:
                        Old Value Type:         %%1876
                        Old Value:              3
                        New Value Type:         %%1876
                        New Value:              1
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-21-4250873927-3395330480-3391837428-1001, evaluador, ENTERPRISE-1809, 0x1d66c...}
InstanceId         : 4657
TimeGenerated      : 12/19/2018 5:36:54 AM
TimeWritten        : 12/19/2018 5:36:54 AM
UserName           :
Site               :
Container          :
```

© 2019 Microsoft Corporation. All rights reserved.

```
[I]: Setting up audit rules...
The command was successfully executed.
[I]: Configuring Diagnostic settings...
[I]: Full mode selected
[I]: Showing audit event...

-------------------------------------------------------


EventID            : 4657
MachineName        : ENTERPRISE-1809
Data               : {}
Index              : 19115
Category           : (12801)
CategoryNumber     : 12801
EntryType          : SuccessAudit
Message            : A registry value was modified.

                     Subject:
                        Security ID:            S-1-5-21-4250873927-3395330480-3391837428-1001
                        Account Name:           evaluador
                        Account Domain:         ENTERPRISE-1809
                        Logon ID:               0x1d66c

                     Object:
                        Object Name:            \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection
                        Object Value Name:      AllowTelemetry
                        Handle ID:              0xcf8
                        Operation Type:         %%1905

                     Process Information:
                        Process ID:             0x1a44
                        Process Name:           C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

                     Change Information:
                        Old Value Type:         %%1876
                        Old Value:              1
                        New Value Type:         %%1876
                        New Value:              3
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-21-4250873927-3395330480-3391837428-1001, evaluador, ENTERPRISE-1809, 0x1d66c...}
InstanceId         : 4657
TimeGenerated      : 12/19/2018 5:37:32 AM
TimeWritten        : 12/19/2018 5:37:32 AM
UserName           :
Site               :
Container          :
```

Using a **User account**, the same results were obtained. However, the user account does not have permission to see the event log.

```
[I]: Showing audit event...
Get-EventLog : Requested registry access is not allowed.
At C:\FMT\common\AuditLog.ps1:6 char:14
+ ...    $event = Get-EventLog $eventType -InstanceId $eventID -Newest $new ...
+                ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-EventLog], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.GetEventLogCommand
```

33.2.16.4 Verdict

As the above results state, the user with administrator rights and the user without administrator rights can choose between *Basic* and *Full* diagnostic data mode.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 22** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 22**.

© 2019 Microsoft Corporation. All rights reserved.

33.2.17 Test 23 - Configure Remote Connection Inactivity Timeout

Remote connection inactivity timeout will be configured during this test case.

33.2.17.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

33.2.17.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log in using the administrator account.
- Run the script *Test 23 - RemoteInactivityTimeout.ps1*. To execute the script, open a *PowerShell* terminal as administrator, go to the path location where the script is stored and type the following command:

  > .\Test 23 - RemoteInactivityTimeout

- The script source code is as follows:

```
. .\common\AuditLog.ps1
. .\common\AuditRegistryKeys.ps1
#
# Test FMT_MOF_EXT.1 - Test 23
#

Write-Host "[I]: Setting up audit rules..."
auditpol /set /subcategory:"Registry"   /success:enable /failure:enable

Write-Host "[I]: Setting up the remote connection inactivity timeout..."
$pathFolder =
    "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows
    NT\Terminal Services"
$keyName = "MaxConnectionTime"
$fullPath = $pathFolder+"\"+$keyName

AddAuditToRegKey $pathFolder "Users" "SetValue"     "ContainerInherit" "none"
    "Success,Failure"

If (!(Test-Path -Path $fullPath)){
    New-ItemProperty -Path $pathFolder -Name    $keyName -Value "60001"
        -PropertyType DWORD    -Force
}else{
    Set-ItemProperty -Path $pathFolder -Name    $keyName -Value "60001"
        -PropertyType DWORD
}
Start-Sleep -s 1

Write-Host "[I]: Showing audit event..."
GetAuditLog Security 4657 1
```

© 2019 Microsoft Corporation. All rights reserved.

```
Write-Host "[I]: Restoring audit rules..."
RemoveAuditToRegKey $pathFolder "Users"     "SetValue" "ContainerInherit" "none"
    "Success,  Failure"
auditpol /set /subcategory:"Registry"   /success:disable /failure:disable
```

- Open the *Group Policy Editor*, (WIN +r, then type: *gpedit.msc*)
- Observe the results.
- Log out and log in again, but this time using the local user account without administrator rights.
- Repeat the previous steps running the script from a non-administrator *PowerShell* terminal.

33.2.17.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has executed the script successfully using an **Administrator account** and an audit entry was generated, as it is shown in the following image:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A registry value was modified.

                       Subject:
                        Security ID:            S-1-5-21-473433637-2525778249-1789179070-500
                        Account Name:           Administrator
                        Account Domain:         SURFACEBOOK2
                        Logon ID:               0x354C2

                       Object:
                        Object Name:            \REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services
                        Object Value Name:      MaxConnectionTime
                        Handle ID:              0xcbc
                        Operation Type:         New registry value created

                       Process Information:
                        Process ID:             0x1f68
                        Process Name:           C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

                       Change Information:
                        Old Value Type:         -
                        Old Value:              -
                        New Value Type:         REG_DWORD
                        New Value:              60001
Id                   : 4657
Version              : 0
Qualifiers           :
Level                : 0
Task                 : 12801
Opcode               : 0
Keywords             : -9214364837600034816
RecordId             : 1634
ProviderName         : Microsoft-Windows-Security-Auditing
ProviderId           : 54849625-5478-4994-a5ba-3e3b0328c30d
LogName              : Security
ProcessId            : 4
ThreadId             : 8692
MachineName          : SURFACEBOOK2
UserId               :
TimeCreated          : 12/6/2018 10:56:20 AM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Security
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      : Registry
KeywordsDisplayNames : {Audit Success}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty...}
```

The remote inactivity timeout time has been successfully modified.

Using a **User account**, the evaluator has executed the script with no success and the following errors were shown:

```
[I]: Setting up audit rules...
Error 0x00000522 occurred:
A required privilege is not held by the client.

[I]: Setting up the remote connection inactivity timeout...
Set-Acl : Requested registry access is not allowed.
At C:\Users\userFMT\Desktop\FMT\common\AuditRegistryKeys.ps1:19 char:19
+     $RegKey_ACL | Set-Acl $key
+                   ~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...rminal Services:String) [Set-Acl], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.SetAclCommand

New-ItemProperty : Requested registry access is not allowed.
At C:\Users\userFMT\Desktop\FMT\Test 9 - RemoteInactivityTimeout.ps1:18 char:5
+     New-ItemProperty -Path $pathFolder -Name $keyName -Value "60001"  ...
+     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACH...rminal Services:String) [New-ItemProperty], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.NewItemPropertyCommand

[I]: Showing audit event...
Get-WinEvent : Could not retrieve information about the Security log. Error: Attempted to perform an unauthorized operation..
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : LogInfoUnavailable,Microsoft.PowerShell.Commands.GetWinEventCommand

Get-WinEvent : There is not an event log on the localhost computer that matches "Security".
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Security:String) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingLogsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

Get-WinEvent : The parameter is incorrect
At C:\Users\userFMT\Desktop\FMT\common\AuditLog.ps1:30 char:14
+ ...     $event = Get-WinEvent -FilterHashtable @{logname = $eventType; id  ...
+                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WinEvent], EventLogException
    + FullyQualifiedErrorId : System.Diagnostics.Eventing.Reader.EventLogException,Microsoft.PowerShell.Commands.GetWinEventCommand
```

In addition, access to the *Local Group Policy Editor* is not allowed to standard users:

© 2019 Microsoft Corporation. All rights reserved.

33.2.17.4 Verdict

As the above results state, the remote connection inactivity timeout can only be modified by a user with administrator rights.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 23** requirements established in the assurance activity section.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to **Test 23**.

33.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FMT_MOF_EXT.1.1-FMT_SMF_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

34 FPT_ACF_EXT.1.1

The assurance activity for the **FPT_ACF_EXT.1.1** requirement is stated as follows:

> The evaluator will confirm that the TSS specifies the locations of kernel drivers/modules, security audit logs, shared libraries, system executables, and system configuration files. Every file does not need to be individually identified, but the system's conventions for storing and protecting such files must be specified. The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action):
>
> - **Test 1:** The evaluator will attempt to modify all kernel drivers and modules.
> - **Test 2:** The evaluator will attempt to modify all security audit logs generated by the logging subsystem.
> - **Test 3:** The evaluator will attempt to modify all shared libraries that are used throughout the system.
> - **Test 4:** The evaluator will attempt to modify all system executables.
> - **Test 5:** The evaluator will attempt to modify all system configuration files.
> - **Test 6:** The evaluator will attempt to modify any additional components selected.

34.1 Documentation Review activity

34.1.1 Findings

The evaluator has reviewed the section **6.6.2.Protection of OS binaries, Audit and Configuration Data** of the ***Security Target*** document. This section states that kernel, device drivers (*.sys files*), system executables (*.exe files*), and dynamically loadable libraries (*.dll files*) are stored in \%systemRoot%\system32 directory and subdirectories.

Additionally, the TSS also states that audit logs are stored in %systemRoot%\system32\winevt and configuration files are located at %systemRoot%\system32\config.

In addition, an explanation on how the permissions are applied over these kinds of files is given. Standard users have permissions to read and execute kernel, device drivers, system executables and libraries, and they are not authorized to access audit logs and configuration files. However, administrator users have permissions to write and modify kernel, device drivers, system executables and libraries, and have full control over audit logs and configuration files.

34.1.2 Verdict

The evaluator considers that the TSS provides enough information related to where kernel, device drivers, system executables, libraries, configuration files, and audit logs are stored in the system and how the permissions are applied over these files.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

34.2 Test Activity

34.2.1 Test 1

The evaluator will attempt to modify all kernel drivers and modules during this test case.

34.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A user account without administrator rights shall be available.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

34.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.
- Run the script *Test 1 - FPT_ACF_EXT.1.1.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script is stored and type the following command:

  > .\Test 1 - FPT_ACF_EXT.1.1.ps1

  This script attempts to modify all driver (*.sys*) files stored in system32 directory and its subdirectories. The script source code is as follows:

```
#
# FPT_ACF_EXT.1.1 - Test 1
#

Write-Host "[I]: FPT_ACF_EXT.1.1 - Test 1: Attempt to modify all kernel drivers
    and modules..."

$kernelDriversPath = "$env:SystemRoot\system32"

$drivers = Get-ChildItem -Path $kernelDriversPath -Recurse -File -Filter
"*.sys" -ErrorAction SilentlyContinue -ErrorVariable errors
foreach ($err in $errors) {
    Write-Warning $err[0].Exception.Message
}

foreach ($driver in $drivers) {
    try {
```

© 2019 Microsoft Corporation. All rights reserved.

```
        Add-Content -Path $driver.FullName -Encoding Byte
        -Value ([Byte] 0xAA) -ErrorAction Stop
    } catch [Exception] {
        Write-Warning $Error[0].Exception.Message
    }
}
```

- Observe the results.

34.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to modify all kernel drivers and modules. The same warning message for each file was obtained. This message states that the access to those files is denied as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

```
Windows PowerShell
PS C:\Users\userFPT\Desktop\FPT_ACF_EXT.1.1> & '.\Test 1 - FPT_ACF_EXT.1.1.ps1'
[I]: FPT_ACF_EXT.1.1 - Test 1: Attempt to modify all kernel drivers and modules...
WARNING: Access to the path 'C:\Windows\system32\com\dmp' is denied.
WARNING: Access to the path 'C:\Windows\system32\config' is denied.
WARNING: Access to the path 'C:\Windows\system32\Configuration' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\DriverData' is denied.
WARNING: Access to the path 'C:\Windows\system32\DriverState' is denied.
WARNING: Access to the path 'C:\Windows\system32\ias' is denied.
WARNING: Access to the path 'C:\Windows\system32\LogFiles\Fax\Incoming' is denied.
WARNING: Access to the path 'C:\Windows\system32\LogFiles\Fax\Outgoing' is denied.
WARNING: Access to the path 'C:\Windows\system32\LogFiles\Firewall' is denied.
WARNING: Access to the path 'C:\Windows\system32\LogFiles\WMI' is denied.
WARNING: Access to the path 'C:\Windows\system32\MsDtc' is denied.
WARNING: Access to the path 'C:\Windows\system32\networklist' is denied.
WARNING: Access to the path 'C:\Windows\system32\SleepStudy' is denied.
WARNING: Access to the path 'C:\Windows\system32\spool\PRINTERS' is denied.
WARNING: Access to the path 'C:\Windows\system32\spool\SERVERS' is denied.
WARNING: Access to the path 'C:\Windows\system32\sru' is denied.
WARNING: Access to the path 'C:\Windows\system32\Tasks' is denied.
WARNING: Access to the path 'C:\Windows\system32\wbem\MOF' is denied.
WARNING: Access to the path 'C:\Windows\system32\WDI' is denied.
WARNING: Access to the path 'C:\Windows\system32\win32k.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\win32kbase.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\win32kfull.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\win32kns.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\1394ohci.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\3ware.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\acpi.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\AcpiDev.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\acpiex.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\acpipagr.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\acpipmi.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\acpitime.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\adp80xx.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\afd.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\afunix.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\agilevpn.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\ahcache.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\amdk8.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\amdppm.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\amdsata.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\amdsbs.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\amdxata.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\appid.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\applockerfltr.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\AppVStrm.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\AppvVemgr.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\AppvVfs.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\arcsas.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\asyncmac.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\atapi.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\ataport.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bam.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\battc.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bcmfn2.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\beep.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bindflt.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bowser.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bridge.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\BtaMPM.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bthenum.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\bthhfenum.sys' is denied.
WARNING: Access to the path 'C:\Windows\system32\drivers\BthMini.sys' is denied.
```

34.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

34.2.2 Test 2

The evaluator will attempt to modify all security audit logs during this test case.

34.2.2.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

34.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.
- Run the script *Test 2 - FPT_ACF_EXT.1.1.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

© 2019 Microsoft Corporation. All rights reserved.

> .\Test 2 - FPT_ACF_EXT.1.1.ps1

This script attempts to modify all audit log (*.evtx*) files stored in system32\winevt\logs directory. The script source code is as follows:

```
#
# FPT_ACF_EXT.1.1 — Test 2
#
Write—Host "[I]: FPT_ACF_EXT.1.1 — Test 2: Attempt to modify all security audit
    logs generated by the logging subsystem..."

$auditLogPath = "$env:SystemRoot\system32\winevt\logs"

$logs = Get—ChildItem —Path $auditLogPath —File —Filter "*.evtx" —ErrorAction
    SilentlyContinue —ErrorVariable errors

foreach ($err in $errors) {
    Write—Warning $err[0].Exception.Message
}

foreach ($log in $logs) {
    try {
        Add—Content —Path $log.FullName —Encoding Byte —Value ([Byte] 0xAA)
            —ErrorAction Stop
    } catch [Exception] {
        Write—Warning $Error[0].Exception.Message
    }
}
```

- Observe the results.

34.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to modify all security audit logs. The same warning message for each file was obtained. This message states that the access to those files is denied as it can be observed in the following image:

34.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

34.2.3 Test 3

The evaluator will attempt to modify all shared libraries during this test case.

34.2.3.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

© 2019 Microsoft Corporation. All rights reserved.

34.2.3.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.
- Run the script *Test 3 - FPT_ACF_EXT.1.1.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

  > .\Test 3 - FPT_ACF_EXT.1.1.ps1

  This script attempts to modify all library (*.dll*) files stored in system32 directory and its subdirectories. The script source code is as follows:

```
#
# FPT_ACF_EXT.1.1 – Test 3
#

Write-Host "[I]: FPT_ACF_EXT.1.1 – Test 3: Attempt to modify all shared
    libraries that are used throughout the system..."

$librariesPath = "$env:SystemRoot\system32"

$libraries = Get-ChildItem -Path $librariesPath -Recurse -Filter "*.dll"
    -ErrorAction SilentlyContinue -ErrorVariable errors

foreach ($err in $errors){
    Write-Warning $err[0].Exception.Message
}

foreach ($library in $libraries) {
    try{
        Add-Content -Path $library.FullName -Encoding Byte -Value ([Byte] 0xAA)
            -ErrorAction Stop
    } catch [Exception]{
        Write-Warning $Error[0].Exception.Message
    }
}
```

- Observe the results.

34.2.3.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to modify all shared libraries. The same warning message for each file was obtained. This message states that the access to those files is denied as it can be observed in the following image:

34.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3**.

34.2.4 Test 4

The evaluator will attempt to modify all system executable files during this test case.

© 2019 Microsoft Corporation. All rights reserved.

34.2.4.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

34.2.4.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.
- Run the script *Test 4 - FPT_ACF_EXT.1.1.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

  > .\Test 4 - FPT_ACF_EXT.1.1.ps1

  This script attempts to modify all system executable (*.exe* and *.com*) files stored in system32 directory and its subdirectories. The script source code is as follows:

```
#
# FPT_ACF_EXT.1.1 – Test 4
#

Write-Host "[I]: FPT_ACF_EXT.1.1 – Test 4: Attempt to modify all system
    executables..."

$executablesPath = "$env:SystemRoot\system32"

$executables = Get-ChildItem -Path $executablesPath -Recurse -Include
    *.com,*.exe -ErrorAction SilentlyContinue -ErrorVariable errors

foreach ($err in $errors){
    Write-Warning $err[0].Exception.Message
}

foreach ($executable in $executables) {
    try {
        Add-Content -Path $executable.FullName -Encoding Byte -Value ([Byte]
            0xAA) -ErrorAction Stop
    } catch [Exception] {
        Write-Warning $Error[0].Exception.Message
    }
}
```

- Observe the results.

34.2.4.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to modify all system executable files. The same warning message for each file was obtained. This message states that the access to those files is denied as it can be observed in the following image:

34.2.4.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4**.

© 2019 Microsoft Corporation. All rights reserved.

34.2.5 Test 5

The evaluator will attempt to modify all system configuration files during this test case.

34.2.5.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

34.2.5.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.
- Run the script *Test 5 - FPT_ACF_EXT.1.1.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

  > .\Test 5 - FPT_ACF_EXT.1.1.ps1

  This script attempts to modify all system configuration files stored in system32\config directory and its subdirectories. The script source code is as follows:

  ```
  #
  # FPT_ACF_EXT.1.1 - Test 5
  #

  Write-Host "[I]: FPT_ACF_EXT.1.1 - Test 5: Attempt to modify all system
      configuration files..."

  $configFilePath = "$env:SystemRoot\system32\config"

  $configFiles= Get-ChildItem -Path $configFilePath -File -ErrorAction
      SilentlyContinue -ErrorVariable errors

  foreach ($err in $errors){
      Write-Warning $err[0].Exception.Message
  }

  foreach ($configFile in $configFiles) {
      try{
          Write-Host $configFile.FullName
      } catch [Exception] {
          Write-Warning $Error[0].Exception.Message
      }
  }
  ```

- Observe the results.

34.2.5.3 Results

The evaluator has performed this test on the following platforms:

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to modify all system configuration files. The following warning message was obtained. This message states that the access to those files is denied as it can be observed in the following image:

Windows PowerShell

```
PS C:\Users\userFPT\Desktop\FPT_ACF_EXT.1.1> & '.\Test 5 - FPT_ACF_EXT.1.1.ps1'
[I]: FPT_ACF_EXT.1.1 - Test 5: Attempt to modify all system configuration files...
WARNING: Access to the path 'C:\Windows\system32\config' is denied.
PS C:\Users\userFPT\Desktop\FPT_ACF_EXT.1.1>
```

34.2.5.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 5** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 5**.

34.2.6 Test 6

The evaluator will attempt to modify additional selected components during this test case.

34.2.6.1 Setup

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

34.2.6.2 Procedure

© 2019 Microsoft Corporation. All rights reserved.

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

34.2.6.3 Results

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

34.2.6.4 Verdict

This test is not applicable due to the assignment made (**None**) by the vendor in the ***Security Target*** document.

34.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_ACF_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

35 FPT_ACF_EXT.1.2

The assurance activity for the **FPT_ACF_EXT.1.2** requirement is stated as follows:

> The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action):
>
> - **Test 1:** The evaluator will attempt to read security audit logs generated by the auditing subsystem
> - **Test 2:** The evaluator will attempt to read system-wide credential repositories
> - **Test 3:** The evaluator will attempt to read any other object specified in the assignment

35.1 Documentation Review activity

35.1.1 Findings

The related assurance activity does not define any action for this requirement.

35.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

35.2 Test Activity

35.2.1 Test 1

The evaluator will attempt to read all the security audit logs during this test case.

35.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A user account without administrator rights shall be available.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

35.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Log into the user account without administrator rights.

© 2019 Microsoft Corporation. All rights reserved.

- Run the script *Test 1 - FPT_ACF_EXT.1.2.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

  > .\Test 1 - FPT_ACF_EXT.1.2.ps1

  This script attempts to read all audit log (*.evtx*) files stored in system32\winevt\logs directory. The script source code is as follows:

```
#FPT_ACF_EXT.1.2 — Test 1
Write-Host "[I]: FPT_ACF_EXT.1.2 — Test 1: Attempt to read security audit logs
    generated by the auditing subsystem...."

$auditLogPath = "$env:SystemRoot\system32\winevt\logs"
$logs = Get-ChildItem -Path $auditLogPath -File -Filter "*.evtx" -ErrorAction
    SilentlyContinue -ErrorVariable errors

foreach ($err in $errors){
    Write-Warning $err[0].Exception.Message
}

foreach ($log in $logs){
    try{
        Get-Content -Path $log.FullName -ErrorAction Stop
    } catch [Exception]{
        Write-Warning $Error[0].Exception.Message
    }
}
```

- Observe the results.

35.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to read all the security audit logs. The same warning message for each file was obtained. This message states that the access to those files is denied as it can be observed in the following image:

35.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

35.2.2 Test 2

The evaluator will attempt to read system-wide credential repositories during this test case.

35.2.2.1 Setup

The applicable setup for this test is the same as the defined one for Test 1.

35.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

© 2019 Microsoft Corporation. All rights reserved.

- Log into the user account without administrator rights.
- Run the script *Test 2 - FPT_ACF_EXT.1.2.ps1*. To execute the script, open a *PowerShell* terminal, go to the path location where the script are stored and type the following command:

  > .\Test 2 - FPT_ACF_EXT.1.2

  This script attempts to read system-wide credential repositories, storing in the registry path HKLM\SAM\SAM. The script source code is as follows:

  ```
  #FPT_ACF_EXT.1.2 — Test 2
  Write—Host "[I]: FPT_ACF_EXT.1.2 — Test 2: Attempt to read system—wide
      credential repositories..."
  try{
      Get—ItemProperty —Path Registry::HKLM\SAM\SAM —ErrorAction Stop
  } catch [Exception]{
      Write—Warning $Error[0].Exception.Message
  }
  ```

- Observe the results.

35.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has attempted to read system-wide credential repositories. The following warning message was obtained. This message states that the access to those files is denied as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

Windows PowerShell

```
PS C:\Users\userFPT\Desktop\FPT_ACF_EXT.1.2> & '.\FPT_ACF_EXT.1.2 - Test 2.ps1'
[I]: FPT_ACF_EXT.1.2 - Test 2: Attempt to read system-wide credential repositories...
WARNING: Requested registry access is not allowed.
PS C:\Users\userFPT\Desktop\FPT_ACF_EXT.1.2>
```

35.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

35.2.3 Test 3

The evaluator will attempt to read additional selected objects during this test case.

35.2.3.1 Setup

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

35.2.3.2 Procedure

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

35.2.3.3 Results

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

35.2.3.4 Verdict

This test is not applicable due to the assignment (**None**) made by the vendor in the ***Security Target*** document.

35.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_ACF_EXT.1.2.

© 2019 Microsoft Corporation. All rights reserved.

36 FPT_ASLR_EXT.1.1

The assurance activity for the **FPT_ASLR_EXT.1.1** requirement is stated as follows:

> The evaluator will select 3 executables included with the TSF. If the TSF includes a web browser it must be selected. If the TSF includes a mail client it must be selected. For each of these apps, the evaluator will launch the same executables on two separate instances of the OS on identical hardware and compare all memory mapping locations. The evaluator will ensure that no memory mappings are placed in the same location. If the rare chance occurs that two mappings are the same for a single executable and not the same for the other two, the evaluator will repeat the test with that executable to verify that in the second test the mappings are different. This test can also be completed on the same hardware and rebooting between application launches

36.1 Documentation Review activity

36.1.1 Findings

The related assurance activity does not define any action for this requirement.

36.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

36.2 Test Activity

Three different executables included with the TSF will be launched in order to verify that no memory mappings are placed in the same location for two different instances of the same executable.

36.2.1 Test 1

36.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- *VMMap* tool (a sysinternal utility that provides the ability to analyze the physical memory) shall be available.

The three executables, which shall be used during the test execution and included with the TSF, are the following:

- Command Prompt (*cmd.exe*).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Edge (*MicrosoftEdge.exe*). For Windows Server, the Internet Explorer application (*iexplore.exe*) will be used instead.
- Mail App (a Trusted Windows Store App, whose process name is *HxOutlook.exe*). For Windows Server the Registry Editor application (*regedit.exe*) will be used instead.

36.2.1.2 Procedure

The following procedure shall be followed to perform the testing activity:

- Run one of the executables defined above (e.g. *Command Prompt*).
- Run the *VMMap* tool and select the process name of the previous executable to examine the memory mapping. Take a screenshot and save the analysis generated by *VMMap*.
- Restart the TOE and repeat the previous steps to execute the second round of memory analysis.
- Compare the obtained results. The memory mappings shall be located in different locations of the memory.
- Repeat the above steps using the other executables (e.g *Microsoft Edge/Internet Explorer* and *Mail App/Registry Editor*).

36.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Go with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Testing platform: Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

© 2019 Microsoft Corporation. All rights reserved.

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF76BC20000
  - Round 2 (Address): 0x00007FF7E9690000

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X00007FF77D4C0000
  - Round 2 (Address): 0X00007FF771BE0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0X00007FF694880000
  - Round 2 (Address): 0X00007FF774620000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF723500000
  - Round 2 (Address): 0X00007FF752000000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0x00007FF758770000
  - Round 2 (Address): 0x00007FF77B840000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF752C60000
  - Round 2 (Address): 0x00007FF6B4180000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0X00007FF76D3B0000
  - Round 2 (Address): 0X00007FF6B4E10000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X00007FF62B730000
  - Round 2 (Address): 0X00007FF721680000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF61C0C0000
  - Round 2 (Address): 0x00007FF73C350000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: - Microsoft Surface Go with Windows 10 Enterprise x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0X00007FF6E75D0000
  - Round 2 (Address): 0X00007FF65EF80000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X00007FF660740000
  - Round 2 (Address): 0X00007FF74B7A0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF60B820000
  - Round 2 (Address): 0x00007FF78FB10000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0X00007FF6E3AA0000
  - Round 2 (Address): 0X00007FF65CE90000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X00007FF7B8820000
  - Round 2 (Address): 0X00007FF68D4E0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF725290000
  - Round 2 (Address): 0x00007FF7F66A0000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0X00000000013E0000
  - Round 2 (Address): 0X0000000000EC0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X0000000000210000
  - Round 2 (Address): 0X0000000000DB0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x0000000000910000
  - Round 2 (Address): 0x0000000000070000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0X00007FF665080000
  - Round 2 (Address): 0X00007FF7E9390000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0X00007FF7EBFD0000
  - Round 2 (Address): 0X00007FF6F22D0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Mail App (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF6341B0000
  - Round 2 (Address): 0x00007FF7D0130000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF759630000
  - Round 2 (Address): 0x00007FF713930000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0x00007FF7E95A0000
  - Round 2 (Address): 0x00007FF6EE2F0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Registry Editor (*regedit.exe*)
  - Round 1 (Address): 0x00007FF6FEE80000
  - Round 2 (Address): 0X00007FF66CF60000

© 2019 Microsoft Corporation. All rights reserved.

Testing Platform: HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Promt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF6C60C0000
  - Round 2 (Address): 0X00007FF6048C0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0x00007FF69B610000
  - Round 2 (Address): 0x00007FF645D20000

© 2019 Microsoft Corporation. All rights reserved.

- Process: MailApp (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF6E2520000
  - Round 2 (Address): 0x00007FF6099D0000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Promt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF74ABC0000
  - Round 2 (Address): 0X00007FF670D30000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0x00007FF62E7F0000
  - Round 2 (Address): 0x00007FF7848F0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: MailApp (*HxOutlook.exe*)
  - Round 1 (Address): 0x00007FF677840000
  - Round 2 (Address): 0x00007FF6AAA80000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Promt (*cmd.exe*)
  - Round 1 (Address): 0x0000000001130000
  - Round 2 (Address): 0X0000000000AF0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Microsoft Edge (*MicrosoftEdge.exe*)
  - Round 1 (Address): 0x0000000001380000
  - Round 2 (Address): 0x0000000000150000

© 2019 Microsoft Corporation. All rights reserved.

- Process: MailApp (*HxOutlook.exe*)
  - Round 1 (Address): 0x00000000010D0000
  - Round 2 (Address): 0x0000000000DF0000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

Since the Mail application and Microsoft Edge browser are not available on this Windows edition, the *Internet Explorer* and *Registry Editor* applications have been used instead.

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF7E19C0000
  - Round 2 (Address): 0x00007FF781D80000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Internet Explorer (*iexplore.exe*)
  - Round 1 (Address): 0x00007FF772500000
  - Round 2 (Address): 0x00007FF60B450000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Registry Editor (*regedit.exe*)
  - Round 1 (Address): 0x00007FF655F80000
  - Round 2 (Address): 0X00007FF644830000

© 2019 Microsoft Corporation. All rights reserved.

Testing platform: Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Since the Mail application and Microsoft Edge browser are not available on this Windows edition, the *Internet Explorer* and *Registry Editor* applications have been used instead.

The evaluator has executed each executable file in two different rounds over the same platform, and has observed that the memory address is different from each round. The following images demonstrate this fact:

- Process: Command Prompt (*cmd.exe*)
  - Round 1 (Address): 0x00007FF6163E0000
  - Round 2 (Address): 0x00007FF72D460000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Internet Explorer (*iexplore.exe*)
  - Round 1 (Address): 0x00007FF724B80000
  - Round 2 (Address): 0x00007FF63E9E0000

© 2019 Microsoft Corporation. All rights reserved.

- Process: Registry Editor (*regedit.exe*)
  - Round 1 (Address): 0x00007FF773630000
  - Round 2 (Address): 0X00007FF6090A0000

© 2019 Microsoft Corporation. All rights reserved.

36.2.1.4 Verdict

As it can be observed in the above images in all cases the memory assigned to one process is different. Therefore, from the results of the previous test, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

© 2019 Microsoft Corporation. All rights reserved.

36.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_ASLR_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

37 FPT_SBOP_EXT.1.1

The assurance activity for the **FPT_SBOP_EXT.1.1** requirement is stated as follows:

> For stack-based OSes, the evaluator will determine that the TSS contains a description of stack-based buffer overflow protections used by the OS. These are referred to by a variety of terms, such as stack cookie, stackguard, and stack canaries. The TSS must include a rationale for any binaries that are not protected in this manner. The evaluator will also perform the following test:
>
> - **Test 1:** The evaluator will inventory the kernel, libraries, and application binaries to determine those that do not implement stack-based buffer overflow protections. This list should match up with the list provided in the TSS.
>
> For OSes that store parameters/variables separately from control flow values, the evaluator will verify that the TSS describes what data structures control values, parameters, and variables are stored. The evaluator will also ensure that the TSS includes a description of the safeguards that ensure parameters and variables do not intermix with control flow values.

37.1 Documentation Review activity

37.1.1 Findings

The evaluator has reviewed the information provided in TSS, section **6.6.3 Protection From Implementation Weaknesses**. This section includes a list about the protections implemented by the TOE. (e.g. *Data Execution Prevention (DEP)* or *Address Space Layout Randomization (ASLR)*).

This TSS section states that all Windows binaries and Windows Store Applications implement stack buffer overrun protection by being compiled with the */GS* option. Moreover, *Windows Store* Applications are checked if they are compiled or not with the buffer overrun protection before ingesting them into the *Windows Store*.

Related to the Windows binaries files, the TSS lists the files which are not compiled with the buffer overrun protection. Including for all of them, a rationale explaining its exclusion. These files are the following:

- *winload.exe* (for x86 and x64 architectures), *winresume.exe* (for x86 and x64 architectures), *tcblaunch.exe* (for x64 architectures), *tcbloader.dll* (for x64 architectures) and *hvloader.exe* (for x64 architectures)
  - These files are loaded before the stack buffer overrun protection mechanism is operational. Therefore, these files are not compiled with this option.
- *ntoskrnl.exe* (for x64 architectures)
  - The 64-bit version of *ntoskrnl.exe* Kernel Patch Protection Mechanism is implemented in a static library, *filter.obj*. The implementation of the Path Protection Mechanism is incompatible with */GS* and therefore *filter.obj* is not compiled */GS*.

© 2019 Microsoft Corporation. All rights reserved.

37.1.2 Verdict

The evaluator considers that the TSS provides enough information related to the stack-base overflow protections implemented by the TOE.

In addition, the TSS has indicated that all windows binaries and Windows Apps have been compiled with the */GS* option except four files (*winload.exe* and *winresume.exe*, for x86 and x64 architectures, and *ntoskrnl.exe*, *tcblaunch.exe*, *tcblaunch.dll* and *hvloader.exe* for x64 architectures). A rationale explaining why these files have not been compiled with */GS* option has been provided.

Hence, the **PASS** verdict is assigned to the documentation review activity.

37.2 Test Activity

37.2.1 Test 1

37.2.1.1 Setup

To do this test, the evaluator shall have access to the *BinSkim* tool and the private symbols to analyze the system binaries to find out whether they have been compiled with stack-based overflow protections or not.

Due to this fact, this test shall be performed at Microsoft facilities.

The following tools are available to the evaluator:

- *Copy-HostSystemFiles.ps1 and Copy-SystemFiles.ps1*: These PowerShell scripts allow the evaluator to copy all the system executables, drivers and libraries stored in the system folder %windir%\System32, maintaining its tree directory structure. The content of both scripts is the following:

```
[CmdletBinding()]
Param(
    [Parameter(
        Mandatory = $true,
        HelpMessage = 'Path to folder where system files will be copied')]
    [string]$DestinationPath,
    [Parameter(
        Mandatory = $false)]
    [switch]$DeleteExisting
)

Process {
    $WinVerRegKey = Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
        NT\CurrentVersion" -ErrorAction Stop
    $DestinationPath = Join-Path -Path $DestinationPath -ChildPath
        $WinVerRegKey.ProductName
    $DestinationPath = Join-Path -Path $DestinationPath -ChildPath
        $Env:PROCESSOR_ARCHITECTURE
    $DestinationPath = Join-Path -Path $DestinationPath -ChildPath
        $WinVerRegKey.BuildLab
```

© 2019 Microsoft Corporation. All rights reserved.

```
    if (Test-Path -Path $DestinationPath)
    {
        if ($DeleteExisting.IsPresent) {
            Remove-Item -Path $DestinationPath -Force -Recurse -ErrorAction Stop}
        else {
            Write-Error "Folder $DestinationPath already exists. Use
                -DeleteExisting or delete folder manually."
            Exit
        }
    }

    $NewFolder = New-Item -Path $DestinationPath -ItemType Directory
        -ErrorAction Stop

    Invoke-Expression -Command ('."{0}\Copy-SystemFiles.ps1" -SystemRootPath
        "{1}" -DestinationPath "{2}"' -f $PSScriptRoot, $env:SystemRoot,
        $DestinationPath)
}
```

```
[CmdletBinding()]
Param(
    [Parameter(
        Mandatory = $true,
        HelpMessage = 'Path to system root folder')]
    [string]$SystemRootPath,
    [Parameter(
        Mandatory = $true,
        HelpMessage = 'Path to folder where system files will be copied')]
    [string]$DestinationPath,
        [Parameter(
        Mandatory = $false)]
    [switch]$DeleteExisting
)

Process{

    if (-not (Test-Path -Path $SystemRootPath -ErrorAction Stop))
    {
        Write-Error -Message "Source system root path $SystemRootPath does not
            exists"
        Exit
    }
    if (-not (Test-Path -Path $DestinationPath -ErrorAction Stop))
    {
        Write-Error -Message "Destination path $DestinationPath does not exists"
        Exit
    }

    foreach ($SubPath in "system32", "system32\drivers")
    {
        $SourcePath = Join-Path -Path $SystemRootPath -ChildPath $SubPath
        if (-not (Test-Path $SourcePath))
```

© 2019 Microsoft Corporation. All rights reserved.

```
        {
            Write-Error "Expected system files path $SourcePath not found"
            Exit
        }

        $CurrentDestinationPath = Join-Path -Path $DestinationPath -ChildPath
            $SubPath

        Write-Output ("Copy system files from: {0} to {1}" -f $SourcePath,
            $CurrentDestinationPath)

        if (Test-Path $CurrentDestinationPath) {
            if ($DeleteExisting.IsPresent) {
                Remove-Item -Path $CurrentDestinationPath -Force -Recurse }
            else {
                Write-Error "Folder $CurrentDestinationPath already exists. Use
                    -DeleteExisting or delete folder manually."
                Exit
                }
        }

        $NewFolder = New-Item -Path $CurrentDestinationPath -ItemType Directory
            -ErrorAction Stop

        Get-ChildItem -Path $SourcePath\*.* -Include @("*.exe","*.dll","*.sys")
            -Force -File | Foreach {
            Write-Output $_.FullName
            Copy-Item -Path $_ -Destination $CurrentDestinationPath -Force
        }
    }
}
```

- *Start-BinSkimRuns.ps1, Run-BinSkimSystemFileList.ps1, Run-BinSkim.ps1*. These files allow the evaluator to run the BinSkim against all the copied files. The following code blocks shows the source code of the scripts quoted above:

```
[CmdletBinding()]
Param(
    [Parameter(Mandatory = $true)]
    [String]$Path,
    [Parameter(Mandatory = $false)]
    [String]$x86SymbolsPath = "SRV*http://symweb",
    [Parameter(Mandatory = $false)]
    [String]$amd64SymbolsPath = "SRV*http://symweb"
)

Get-ChildItem -Path $Path -Recurse -Filter "SystemFileList.txt" | Foreach {

    Write-Output $_.FullName

    $OutputPath = Join-Path -Path (Split-Path -Path $_.FullName -Parent)
        -ChildPath "Results"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    $Command = ('."{0}\Run-binskimSystemFileList.ps1" -SystemFileListPath "{1}"
        -OutputPath "{2}" -AddTimestamp' -f
        $PSScriptRoot,
        $_.FullName,
        $OutputPath)

    if ($OutputPath -match "x86")
    {
        $Command += (' -SymbolsPath "{0}"' -f $x86SymbolsPath)
    }
    elseif ($OutputPath -match "amd64")
    {
        $Command += (' -SymbolsPath "{0}"' -f $amd64SymbolsPath)
    }

    if ($VerbosePreference -eq "Continue")
    {
        $Timestamp = Get-Date -Format "yyyyMMdd-HHmmss"
        Start-Transcript -Path (Join-Path -Path $OutputPath -ChildPath
            ("Run-binskimSystemFiles-{0}.log" -f $Timestamp))
    }
    Write-Output $Command
    Invoke-Expression -Command $Command

    if ($VerbosePreference -eq "Continue")
    {
        Stop-Transcript
    }
}
```

```
[CmdletBinding()]
Param(
    [Parameter(
        Mandatory = $true,
        ValueFromPipeline = $true)]
    [String]$SystemFileListPath,
    [Parameter(
        Mandatory = $true)]
    [String]$OutputPath,
    [Parameter(
        Mandatory = $false,
        HelpMessage="Path to binskim.exe")]
    [String]$binskimPath = $Env:ProgramFiles + "\Microsoft binskim
        2014\binskim.exe",
    [Parameter(
        Mandatory = $false,
        HelpMessage="Private symbols path")]
    [String]$SymbolsPath = "SRV*http://symweb",
    [Parameter(
        Mandatory = $false,
        HelpMessage="binskim check")]
    [String]$Check = "GSCheck",
    [Parameter(
```

© 2019 Microsoft Corporation. All rights reserved.

```
        Mandatory = $false,
        HelpMessage="Indicates whether to delete existing output files")]
    [switch]$DeleteExisting,
    [Parameter(
        Mandatory = $false,
        HelpMessage="Indicates whether to put output files in timestamped
            subfolder")]
    [switch]$AddTimestamp,
    [Parameter(
        Mandatory = $false,
        HelpMessage="Number of attempts to run binskim")]
    [int]$MaxAttempts = 3
)

Begin {
    if (-not (Test-Path -Path $SystemFileListPath))
    {
        Write-Error "$SystemFileListPath not found" -Category InvalidArgument
        exit
    }

    $System32FolderPath = Join-Path -Path (Split-Path -Path $SystemFileListPath
        -Parent) -ChildPath "system32"

    [Array]$SymbolsPathArray = $SymbolsPath.Split(';',
        [System.StringSplitOptions]::RemoveEmptyEntries)
}

Process {
    if ($AddTimestamp)
    {
        $Timestamp = Get-Date -Format "yyyyMMdd-HHmmss"
        $OutputPath = Join-Path -Path $OutputPath -ChildPath $Timestamp
    }

    Write-Verbose "Output path: $OutputPath"

    if (Test-Path -LiteralPath $OutputPath)
    {
        if ($DeleteExisting)
        {
            Remove-Item -LiteralPath $OutputPath -Recurse
        }
        else
        {
            Write-Error -Message "Output folder $OutputPath already exists. Use
                -OutputPath to specify another file or use -DeleteExisting"
            exit
        }
    }
    $NewItem = New-Item -Path $OutputPath -ItemType Directory
```

© 2019 Microsoft Corporation. All rights reserved.

```
$CsvFile = Join-Path -Path $OutputPath -ChildPath "SystemFiles-binskim.csv"
$ErrorFile = Join-Path -Path $OutputPath -ChildPath
    "SystemFiles-binskim-Error.log"
$CsvFileFail = Join-Path -Path $OutputPath -ChildPath
    "SystemFiles-binskim-Fail.csv"

#Test we can write to error file
Add-Content -Path $ErrorFile "test" -ErrorAction Stop
Remove-Item -Path $ErrorFile -ErrorAction Stop

#Run binskim for all files
$binskimOutputPath = Join-Path -Path $OutputPath -ChildPath "binskim"
[array]$BinaryFiles = Get-Content -Path $SystemFileListPath
For ($BinaryFilesIdx = 0; $BinaryFilesIdx -lt $BinaryFiles.Count;
    $BinaryFilesIdx++)
{
    Write-Progress -Activity "Running binskim" -Status $SystemFileListPath
        -CurrentOperation ("{0} ({1}/{2})" -f $BinaryFiles[$BinaryFilesIdx],
        ($BinaryFilesIdx + 1), $BinaryFiles.Length) -PercentComplete
        ($BinaryFilesIdx/$BinaryFiles.Count*100)
    $BinaryPath = Join-Path -Path $System32FolderPath -ChildPath
        $BinaryFiles[$BinaryFilesIdx]

    if (-not (Test-Path -Path $BinaryPath))
    {
        Add-Content -Path $ErrorFile ("Error: Binary file {0} not found" -f
            $BinaryPath)
        continue
    }

    $CurrentOutputPath = Join-Path -Path $binskimOutputPath -ChildPath
        $BinaryFiles[$BinaryFilesIdx]
    $XmlFilePath = $CurrentOutputPath + ".xml"
    $LogFilePath = $CurrentOutputPath + ".log"
    $CurrentOutputPath = Split-Path -Path $CurrentOutputPath -Parent
    if (-not (Test-Path $CurrentOutputPath))
    {
        $NewItem = New-Item -Path $CurrentOutputPath -ItemType Directory
    }

    $Passed = $false
    For($iSym = 0; (-not $Passed) -and ($iSym -lt $SymbolsPathArray.Count);
        $iSym++)
    {
        $Attempts = 0
        do
        {
            $Attempts++

            $Command = ('."{0}\Run-binskim.ps1" -BinaryPath "{1}"
                -XmlFilePath "{2}" -LogFilePath "{3}" -binskimPath "{4}"
                -SymbolsPath "{5}" -Check "{6}" -DeleteExisting' -f
```

© 2019 Microsoft Corporation. All rights reserved.

```
                $PSScriptRoot,
                $BinaryPath,
                $XmlFilePath,
                $LogFilePath,
                $binskimPath,
                $SymbolsPathArray[$iSym],
                $Check)

            Write-Verbose "Executing($Attempts/$MaxAttempts): $Command"

            $binskimResult = Invoke-Expression -Command $Command

            if (($binskimResult.Error -match "E_PDB_NOT_FOUND") -and
                ($SymbolsPathArray[$iSym] -match "srv*"))
            {
                Write-Verbose "Error: E_PDB_NOT_FOUND"
                Start-Sleep -Seconds 20
            }
            elseif ($binskimResult.Error -match "a language could not be
                identified")
            {
                Write-Verbose 'Error: "a language could not be identified"'
                Start-Sleep -Seconds 10
            }
            else
            {
                $Attempts = $MaxAttempts
            }

            $Passed = $binskimResult.Result -match "PASS"
            if ($Passed)
            {
                Add-Member -InputObject $binskimResult -MemberType
                    NoteProperty -Name "SymbolsPath" -Value
                    $SymbolsPathArray[$iSym]
            }
            else
            {
                Add-Member -InputObject $binskimResult -MemberType
                    NoteProperty -Name "SymbolsPath" -Value ""
            }
        }
        While ($Attempts -lt $MaxAttempts)
    }

    if (-not($binskimResult))
    {
        Add-Content -Path $ErrorFile ("Error: Processing {0} failed: {1}" -f
            $BinaryFiles[$BinaryFilesIdx], $Error[0])
    }
    elseif (-not(Test-Path -Path $XmlFilePath))
    {
```

© 2019 Microsoft Corporation. All rights reserved.

```
            Add-Content -Path $ErrorFile ("Error: Processing {0} failed: XML
                file {1} not found" -f $BinaryFiles[$BinaryFilesIdx],
                $XmlFilePath)
        }
        else
        {
            $binskimResult.Binary = $BinaryFiles[$BinaryFilesIdx]

            Export-Csv -LiteralPath $CsvFile -InputObject ($binskimResult |
                Select Binary,Version,Key,Result,IsManaged,XMLFileExists,
            SymbolsPath,Error) -Append -NoTypeInformation

            if ($binskimResult.Result -ne "PASS")
            {
                Export-Csv -LiteralPath $CsvFileFail -InputObject
                    ($binskimResult | Select
                    Binary,Version,Key,Result,IsManaged,XMLFileExists,
                SymbolsPath,Error) -Append -NoTypeInformation
            }
        }
    }
    Write-Progress -Activity "Running binskim" -Status $SystemFileListPath
        -Completed
}

End {
}
```

```
[CmdletBinding()]
Param(
    [Parameter(
        Mandatory = $true,
        ValueFromPipeline = $true,
        HelpMessage="Path to folder containing system files")]
    [String]$BinaryPath,
    [Parameter(
        Mandatory = $false)]
    [String]$XmlFilePath,
    [Parameter(
        Mandatory = $false)]
    [String]$LogFilePath,
    [Parameter(
        Mandatory = $false,
        HelpMessage="Path to Binskim.exe")]
    [String]$BinskimPath = $Env:ProgramFiles + "\Microsoft Binskim
        2014\Binskim.exe",
    [Parameter(
        Mandatory = $false,
        HelpMessage="Private symbols path")]
    [String]$SymbolsPath = "SRV*http://symweb",
    [Parameter(
        Mandatory = $false,
        HelpMessage="Binskim check")]
```

© 2019 Microsoft Corporation. All rights reserved.

```
    [String]$Check = "GSCheck",
    [Parameter(
        Mandatory = $false,
        HelpMessage="Indicates whether to delete existing output files")]
    [switch]$DeleteExisting
)

Begin {
    function AddLogFileContent {
        Param(
            [Parameter(Mandatory = $true)][String]$Path,
            [Parameter(Mandatory = $true)][AllowEmptyString()][String]$Value
        )
        Add-Content -Path $LogFilePath $Value
        Write-Verbose $Value
    }

    if (-not (Test-Path -Path $BinskimPath))
    {
        Write-Error "$BinskimPath not found" -Category InvalidArgument
        exit
    }
}

Process {
    if (-not (Test-Path -Path $BinaryPath))
    {
        Write-Error "Binary file $BinaryPath not found" -Category InvalidArgument
        exit
    }

    if (-not ($XmlFilePath))
    {
        $XmlFilePath = $BinaryPath + ".xml";
    }

    if (-not ($LogFilePath))
    {
        $LogFilePath = $BinaryPath + ".log";
    }

    if (Test-Path -LiteralPath $XmlFilePath)
    {
        if ($DeleteExisting)
        {
            Remove-Item -LiteralPath $XmlFilePath -Recurse
        }
        else
        {
            Write-Error -Message "Xml file $XmlFilePath already exists. Use
                -XmlFilePath to specify another file or use -DeleteExisting"
            exit
```

© 2019 Microsoft Corporation. All rights reserved.

```
        }
    }

    if (Test-Path -LiteralPath $LogFilePath)
    {
        if ($DeleteExisting)
        {
            Remove-Item -LiteralPath $LogFilePath -Recurse
        }
        else
        {
            Write-Error -Message "Log file $LogFilePath already exists. Use
                -LogFilePath to specify another file or use -DeleteExisting"
            exit
        }
    }

    Write-Verbose ("Binary file: {0}" -f $BinaryPath)
    Write-Verbose ("XML file    : {0}" -f $XmlFilePath)
    Write-Verbose ("Log file    : {0}" -f $LogFilePath)

    #create temp Binskim output folder
    $TempOutputPath = Join-Path -Path $env:TEMP -ChildPath
        ([System.IO.Path]::GetRandomFileName())
    while (Test-Path -Path $TempOutputPath)
    {
        $TempOutputPath = Join-Path -Path $env:TEMP -ChildPath
            ([System.IO.Path]::GetRandomFileName())
    }
    $Item = New-Item -ItemType Directory -Path $TempOutputPath

    AddLogFileContent -Path $LogFilePath ("Binskim run started: {0}" -f
        (Get-Date -Format G))
    AddLogFileContent -Path $LogFilePath
        "------------------------------------------------------------"

    $StdOutPath = Join-Path -Path $TempOutputPath -ChildPath "Output.txt"
    $StdErrPath = Join-Path -Path $TempOutputPath -ChildPath "Error.txt"

    $BinskimResult = New-Object PSObject
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
        "Binary" -Value $BinaryPath
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
        "Version" -Value ""
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name "Key"
        -Value ""
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
        "Result" -Value ""
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
        "IsManaged" -Value ""
    Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
        "XMLFileExists" -Value "False"
```

© 2019 Microsoft Corporation. All rights reserved.

```
Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
    "XMLFile" -Value ""
Add-Member -InputObject $BinskimResult -MemberType NoteProperty -Name
    "Error" -Value ""

#Binskim arguments
$BinskimArgs = @('/c', $Check, '/Red', '/NoLogo', '/v', '/s', ('"{0}"' -f
    $SymbolsPath), '/d', ('"{0}"' -f $TempOutputPath), '/target', ('"{0}"'
    -f $BinaryPath))
AddLogFileContent -Path $LogFilePath ([string]::Join(" ", $BinskimArgs))


#Run Binskim
$Proc = Start-Process -FilePath $BinskimPath -ArgumentList $BinskimArgs
    -NoNewWindow -RedirectStandardOutput $StdOutPath -RedirectStandardError
    $StdErrPath -Wait -PassThru
Start-Sleep -m 100

$FirstLine = $true
Get-Content -Path $StdErrPath | Foreach {
    if (-not $FirstLine)
    {
        $BinskimResult.Error += "; "
    }
    $BinskimResult.Error += [string]$_
    $FirstLine = $false
}

#Add Binskim output to log file
AddLogFileContent -Path $LogFilePath ""
AddLogFileContent -Path $LogFilePath "Output:"
Get-Content -Path $StdOutPath | ForEach { AddLogFileContent -Path
    $LogFilePath $_ }
Remove-Item -Path $StdOutPath -Recurse

if ($BinskimResult.Error)
{
    AddLogFileContent -Path $LogFilePath ""
    AddLogFileContent -Path $LogFilePath "Error:"
    Get-Content -Path $StdErrPath | ForEach { AddLogFileContent -Path
        $LogFilePath $_ }
    Remove-Item -Path $StdErrPath -Recurse
}

#Log Binskim exit code
AddLogFileContent -Path $LogFilePath ("Binskim finished with exit code: {0}"
    -f $Proc.ExitCode)

#Find XML file
[array]$XmlFiles = Get-ChildItem -Path $TempOutputPath -Filter *.xml -File
if ($XmlFiles.Length -eq 1)
{
```

© 2019 Microsoft Corporation. All rights reserved.

```
        Move-Item -Path $XmlFiles[0].FullName -Destination $XmlFilePath
        $BinskimResult.XMLFileExists = "True"
        $BinskimResult.XMLFile = $XmlFilePath
        AddLogFileContent -Path $LogFilePath ("Binskim created xml file: {0}" -f
            $XmlFilePath)

        #Process XML file and write to CSV file
        [xml]$XmlDoc = Get-Content -Path $XmlFilePath
        $IsManagedNode = $XmlDoc.SelectSingleNode(".//items/item/IsManaged")
        if ($isManagedNode)
        {
            $BinskimResult.IsManaged = $IsManagedNode.InnerText.ToUpper()
        }

        $ResultNode = $XmlDoc.SelectSingleNode(".//items/item/result")
        if ($ResultNode)
        {
            $BinskimResult.Result = $ResultNode.InnerText.ToUpper()
        }

        $KeyNode = $XmlDoc.SelectSingleNode(".//items/item/key")
        if ($KeyNode)
        {
            $BinskimResult.Key = $KeyNode.InnerText.ToUpper()
        }

        $VersionNode = $XmlDoc.SelectSingleNode(".//items/item/version")
        if ($VersionNode)
        {
            $BinskimResult.Version = $VersionNode.InnerText.ToUpper()
        }
    }

    if ($XmlFiles.Length -eq 0)
    {
        AddLogFileContent -Path $LogFilePath "No Binskim xml file found"
    }
    if ($XmlFiles.Length -gt 1)
    {
        AddLogFileContent -Path $LogFilePath "Multiple Binskim xml files found"
    }

    AddLogFileContent -Path $LogFilePath ""
    AddLogFileContent -Path $LogFilePath ("Binskim run finished: {0}" -f
        (Get-Date -Format G))
    AddLogFileContent -Path $LogFilePath
        "------------------------------------------------------------"

    AddLogFileContent -Path $LogFilePath ""
    AddLogFileContent -Path $LogFilePath "Result:"
    AddLogFileContent -Path $LogFilePath
        "------------------------------------------------------------"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    AddLogFileContent -Path $LogFilePath ("Binary        :{0}" -f
        $BinskimResult.Binary)
    AddLogFileContent -Path $LogFilePath ("Result        :{0}" -f
        $BinskimResult.Result)
    AddLogFileContent -Path $LogFilePath ("IsManaged     :{0}" -f
        $BinskimResult.IsManaged)
    AddLogFileContent -Path $LogFilePath ("XMLFileExists:{0}" -f
        $BinskimResult.XMLFileExists)
    AddLogFileContent -Path $LogFilePath ("XMLFile       :{0}" -f
        $BinskimResult.XMLFile)
    AddLogFileContent -Path $LogFilePath ("Error         :{0}" -f
        $BinskimResult.Error)

    #remove temp output folder
    Remove-Item -Path $TempOutputPath -Recurse

    return $BinskimResult
}

End {
}
```

37.2.1.2 Procedure

The following steps shall be performed in order to complete this testing assurance activity:

- Create a copy of all the system files stored under the system folder, maintaining its tree structure. It can be done by executing the script *Copy-SystemFiles.ps1*.
- Run the script *Start-BinSkimRun.ps1* to start the analysis of the copied files using the *BinSkim* tool.
- Once the process has finished, analyze the output generated during the *BinSkim* execution in order to determine which files have not been compiled with */GS* flag.

37.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has analyzed the *BinSkim* output and has observed the following behaviour. For platforms with x86 architecture, *winload.exe* and *winresume.exe* have not been compiled with the */GS* flag, as it can be observed in the next table:

Binary	Version	Result
winload.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
winresume.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL

For platforms with x64 architecture, there are four more files which have not been compiled with the */GS* flag: ,*tcblaunch.exe*, *tcbloader.dll*, *ntoskrnl.exe* and *hvloader.exe*.

Binary	Version	Result
tcblaunch.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
tcbloader.dll	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
hvloader.dll	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
ntoskrnl.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
winload.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
winresume.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL

In addition, both Windows Server editions (*Standard and Datacenter (x64 architecture)*), produced the same output.

Binary	Version	Result
tcblaunch.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
tcbloader.dll	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
hvloader.dll	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
ntoskrnl.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
winload.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL
winresume.exe	10.0.17763.1 (WINBUILD.160101.0800)	FAIL

37.2.1.4 Verdict

The obtained results have been the same as the ones expected according to the TSS definition. Therefore, the **PASS** verdict is assigned to **Test 1**

© 2019 Microsoft Corporation. All rights reserved.

37.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_SBOP_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

38 FPT_SRP_EXT.1.1

The assurance activity for the **FPT_SRP_EXT.1.1** requirement is stated as follows:

> For each selection specified in the ST, the evaluator will ensure that the corresponding tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action):
>
> - **Test 1:** The evaluator will configure the OS to only allow code execution from the core OS directories. The evaluator will then attempt to execute code from a directory that is in the allowed list. The evaluator will ensure that the code they attempted to execute has been executed.
> - **Test 2:** The evaluator will configure the OS to only allow code execution from the core OS directories. The evaluator will then attempt to execute code from a directory that is not in the allowed list. The evaluator will ensure that the code they attempted to execute has not been executed.
> - **Test 3:** The evaluator will configure the OS to only allow code that has been signed by the OS vendor to execute. The evaluator will then attempt to execute code signed by the OS vendor. The evaluator will ensure that the code they attempted to execute has been executed.
> - **Test 4:** The evaluator will configure the OS to only allow code that has been signed by the OS vendor to execute. The evaluator will then attempt to execute code signed by another digital authority. The evaluator will ensure that the code they attempted to execute has not been executed.
> - **Test 5:** The evaluator will configure the OS to allow execution of a specific application based on version. The evaluator will then attempt to execute the same version of the application. The evaluator will ensure that the code they attempted to execute has been executed.
> - **Test 6:** The evaluator will configure the OS to allow execution of a specific application based on version. The evaluator will then attempt to execute an older version of the application. The evaluator will ensure that the code they attempted to execute has not been executed.
> - **Test 7:** The evaluator will configure the OS to allow execution based on the hash of the application executable. The evaluator will then attempt to execute the application with the matching hash. The evaluator will ensure that the code they attempted to execute has been executed.
> - **Test 8:** The evaluator will configure the OS to allow execution based on the hash of the application executable. The evaluator will modify the application in such a way that the application hash is changed. The evaluator will then attempt to execute the application with the matching hash. The evaluator will ensure that the code they attempted to execute has not been executed.

© 2019 Microsoft Corporation. All rights reserved.

38.1 Documentation Review activity

38.1.1 Findings

The related assurance activity does not define any action for this requirement.

38.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

38.2 Test Activity

38.2.1 Test 1 & Test 2

This test configures the OS to only allow code execution from the core OS directories. An application will be executed, first from a directory in the whitelist, then from a directory in the blacklist.

38.2.1.1 Setup

To execute this test, a Software Restriction Policy must be enabled. To do that, the evaluator should perform these next steps:

- Open the *Local Security Policy* Window (*WIN* + *R*, then type *secpol.msc*)
- Go to *Security Settings* -> *Software Restriction Policies*. Right-click on it and select *New Software Restriction Policies*

© 2019 Microsoft Corporation. All rights reserved.

- Then, go to *Security Levels*, right-click in *Disallowed* and select *Set as default* option.

© 2019 Microsoft Corporation. All rights reserved.

- Restart to apply the changes.

Once this Software Restriction Policy is enabled, there are only two default paths that allow code execution: %SystemRoot% *(System32)* and %ProgramFilesDir% *(Program Files)*.

38.2.1.2 Procedure

To ensure that the rules configured during the setup are properly applied, the evaluator shall perform these steps:

- First, go to System32 directory (*C:\Windows\System32*) and select an executable file (e.g. *cmd.exe*). Create a shortcut to this executable in the Desktop.
- Second, try to execute the executable from the *System32* directory. The program shall be executed properly.
- Finally, try to execute the executable from the shortcut created in the first step. The program shall not be executed properly. An error message will be shown.

38.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has attempted to execute the executable file from the System32 directory. The execution was successful as it can be observed in the following image:

Then, the evaluator has attempted to execute the program from the shortcut created in the desktop. The execution was blocked and an error message was shown, as it can be observed in the following image:

38.2.1.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1 and Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1 and Test 2**.

38.2.2 Test 3 & Test 4

This test configures the OS to only allow code that has been signed by the OS vendor to execute. Two applications will be executed, the first one signed by the OS vendor and the second one signed by another digital authority.

38.2.2.1 Setup

The applicable setup for these tests is the same as for *Test 1 & Test 2*.

In addition, two different executables will be used for the test executions. These executables must be downloaded in the target machine:

- *Firefox Setup 63.0.3exe*
- *MediaCreationTool1809.exe*

38.2.2.2 Procedure

To perform these tests the evaluator shall create a new software restriction rule to force only the code execution of programs signed by the OS vendor.

The evaluator shall obtain the certificate used to sign the file executable. To do that, the evaluator shall perform these steps:

- First, go to the directory where *MediaCreationTool1809.exe* is stored. Right-click over it and open its Properties.
- Second, select the *Digital Signatures* tab, and select one of the digital signatures of the file, e.g. SHA-256, and click on the *Details* button.

© 2019 Microsoft Corporation. All rights reserved.

- Click over *View Certificate* button, to view details about the certificate.

© 2019 Microsoft Corporation. All rights reserved.

- In the new opened window, go to *Details* tab and click over *Copy to file...* button.

© 2019 Microsoft Corporation. All rights reserved.

- Follow the instructions of the *Certificate Export Wizard* and save the certificate using a *DER* encoded format.

© 2019 Microsoft Corporation. All rights reserved.

- Open the *Local Security Policy*. Go to *Secure Settings -> Software Restriction Policies -> Additional Rules*. Right-click and select *New Certificate Rule…*

© 2019 Microsoft Corporation. All rights reserved.

- After that, select the certificate obtained in the previous step and configure the *Security Level* option as *Unrestricted*.

© 2019 Microsoft Corporation. All rights reserved.

- Click on the *Apply* button, a warning prompt will be shown, select the *Yes* option.

- In the next window, select the option: *Enforce certificate rules*. Then click the *OK* button.

© 2019 Microsoft Corporation. All rights reserved.

- Finally restart to apply the changes.

After rebooting, the evaluator shall perform the following steps to ensure that the new rule created is working as expected:

- Go to the location where the executable files are stored and attempt to execute *MediaCreationTool.exe*. The execution shall be successful.
- Now, attempt to execute the *Firefox Setup 63.0.3.exe* file. The execution shall be blocked and an error message will be shown.

38.2.2.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has attempted to execute the executable file which has been signed by the OS vendor. The execution was successful.

Then, the evaluator has attempted to execute the executable file which has been signed by other digital authority and not by the OS vendor. The execution was unsuccessful, showing the next error message:

© 2019 Microsoft Corporation. All rights reserved.

38.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 3 and Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3 and Test 4**.

38.2.3 Test 5 & Test 6

This test configures the OS to only allow to execute a specific version of an application. First the specified version will be executed and after that an older version will be attempted to install.

38.2.3.1 Setup

The ***Security Target*** document includes the following footnote in the SFR definition which states that the software restriction version-based rules are only applicable for Windows Enterprise Edition and Windows Server.

[12] Windows 10 Enterprise and Windows Server can restrict program execution based on a version using AppLocker and Device Guard; Windows 10 Pro and Windows 10 Home editions cannot.

Therefore these tests are only applicable for those editions and the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

Due to these test shall be performed using *Applocker* application, the following setup is necessary to perform the tests:

© 2019 Microsoft Corporation. All rights reserved.

- *Applocker* shall be configured to enforce the execution rules. To do that, go to the *Local Security Policy* tool and go to *Application Control Policies-> Applocker. Right-click* on it, select the *Properties* option and configure it as follows:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

- *Application Identity Service* must be running. The evaluator has developed a script to automate the start of the service and set it to start on Windows boot. The content of the script is the following:

```
#
# Test 5 and 6 — Applocker
#

#Check service status
Write-Host "[I]: Checking Application Identity Service status..."
$currentstatus = Get-Service -Name 'AppIDSvc'

if ($currentStatus.Status -eq "Stopped") {
    Write-Host "[E]: Application Identity Service is stopped"
    Write-Host "[I]: Starting the service..."
    Start-Service AppIDSvc
    Set-ItemProperty -Path
    'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
    AppIDSvc'
     -Name Start -Value 2
    Start-Process 'gpupdate.exe' -ArgumentList '/force' -WorkingDirectory
    "C:\Windows\System32"
} else {
    Write-Host "[I]: Application Identity Service is running"
    Write-Host "[I]: Setting the service to automatic start..."
    #Set-Service AppIDSvc -StartupType "Automatic"
    Set-ItemProperty -Path
    'Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
    AppIDSvc'
     -Name Start -Value 2
    Start-Process 'gpupdate.exe' -ArgumentList '/force' -WorkingDirectory
    "C:\Windows\System32"
}
```

On Windows Server, the service can be started with a GPO following the same steps described at the beginning of the test activity. The GPO should be configured as shown in the following image:

© 2019 Microsoft Corporation. All rights reserved.

The following executables provided by the OS vendor shall be available in the evaluated platform. These executables files are signed by a test certificate issued by a test root:

- *TestConsolev10.exe*
- *TestConsolev11.exe*

In order for the version-based rule to work properly, the evaluator shall include a certificate into the Personal Certificate Store and the Trusted Root Certification Authorities Store. These are the necessary steps to do it:

- Open the Microsoft Management Console tool: (*WIN+R*, then type *mmc*)
- Go to *File -> Add/Remove Snap-in...* (*Control + M*) to add the Certificates snap-in to the viewer. To configure it, follow these steps:
  - Add the *Certificates* Snap-in:

© 2019 Microsoft Corporation. All rights reserved.

– Select the checkbox: *Computer account*.

© 2019 Microsoft Corporation. All rights reserved.

– Select the checkbox: *Local Computer*.

© 2019 Microsoft Corporation. All rights reserved.

- Go to *Personal* -> *Certificates*. Then, right-click in *Actions* -> *All tasks* -> *Import...* and follow the wizard to import the test certificate.

© 2019 Microsoft Corporation. All rights reserved.

– Browse the certificate.

© 2019 Microsoft Corporation. All rights reserved.

– Place it in the _Personal Store)

© 2019 Microsoft Corporation. All rights reserved.

– Finally, click *Next* and *Finish*.

© 2019 Microsoft Corporation. All rights reserved.

- Additionally, go to *Trusted Root Certification Authorities* -> *Certificates* and repeat the steps described above to import the test root.

38.2.3.2 Procedure

© 2019 Microsoft Corporation. All rights reserved.

To perform these tests the evaluator shall create a new version-based software restriction rule to force only the code execution of programs with a specific version.

To do that, the evaluator shall perform these steps:

- Open the Local Security Policy tool: (*WIN+R*, then type *secpol.msc*) and go to *Application Control Policies -> Applocker -> Executable Rules*.
- Right click on it and select *Create New Rule…* option. The wizard shall be opened.
- Configure the new version-rule to allow the execution of the executable file with the version 1.1 as the following images show:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

- Before creating the rule, a warning prompt shall be shown asking whether the default rules shall be created or not. Select the *No* option.

© 2019 Microsoft Corporation. All rights reserved.

- The defined rule in *Applocker* shall be like the following image:

- After creating the rule, attempt to execute *TestConsolev11.exe*. The execution shall be successful.
- Finally, attempt to execute *TestConsolev10.exe*. The execution shall be blocked. A warning message shall be shown providing the information about the error.

38.2.3.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has attempted to execute the executable file which has the allowed specified version defined in the rule/policy. The execution was successful.

© 2019 Microsoft Corporation. All rights reserved.

After that, the evaluator has attempted to execute the executable file which has not the allowed specified version defined in the rule/policy. The execution was unsuccessful and the following error message was shown, indicating that the execution was blocked by the system administrator.

38.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 5 and Test 6** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 5 and Test 6**.

38.2.4 Test 7 & Test 8

This test configures the OS to only allow code execution based on the hash of the application executable. First, an application with a hashed which match with the one configured will be executed. After that, the application will be modified to change the hash before being executed.

© 2019 Microsoft Corporation. All rights reserved.

38.2.4.1 Setup

The applicable setup for these tests is the same as for *Test 1 & Test 2*

In addition, a hexadecimal editor (e.g. *HxD*) must be installed in order to modify the original executable file.

To perform these tests, a bash script has been created to use during the tests execution:

- Test 7&8.bat

38.2.4.2 Procedure

First, a new hash-based software restriction rule is needed. To create it, the evaluator shall perform these following steps:

- First of all, create a copy of *Test7&8.bat*. Then, open it with the hexadecimal editor and modify some bytes.

- Open the *Local Security Policy*. (*WIN+r*, then type *secpol.msc*)
- Go to *Secure Settings -> Software Restriction Policies -> Additional Rules*. Right-click on it and select *New Hash Rule...*

© 2019 Microsoft Corporation. All rights reserved.

- Select the original script (*Test 7&8.bat*) and configure the *Security Level* option as *Unrestricted*.

© 2019 Microsoft Corporation. All rights reserved.

- Click over *Apply* button and restart to apply the changes.

After rebooting, the evaluator shall perform these steps to check that the defined rule work as expected.

- First, go to the location of the *.bat* files and attempt to execute the original script. The execution shall be successful.
- Then, attempt to execute the modified script with a different hash. The execution shall be unsuccessful and an error message shall be shown.

38.2.4.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has attempted to execute the original script file with the same hash as the one defined in the rule. The execution was successful as it can be observed in the following image:

After that, the evaluator has attempted to execute the modified script file with a different hash as the one defined in the rule. The execution was unsuccessful and an error message was shown indicating that the execution was blocked:

38.2.4.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 7 and Test 8** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 7 and Test 8**.

38.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_SRP_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

39 FPT_TST_EXT.1.1

The assurance activity for the **FPT_TST_EXT.1.1** requirement is stated as follows:

> The evaluator will verify that the TSS section of the ST includes a comprehensive description of the boot procedures, including a description of the entire bootchain, for the TSF. The evaluator will ensure that the OS cryptographically verifies each piece of software it loads in the bootchain to include bootloaders and the kernel. Software loaded for execution directly by the platform (e.g. first-stage bootloaders) is out of scope. For each additional category of executable code verified before execution, the evaluator will verify that the description in the TSS describes how that software is cryptographically verified.
>
> The evaluator will verify that the TSS contains a description of the protection afforded to the mechanism performing the cryptographic verification.
>
> The evaluator will perform the following tests:
>
> - **Test 1:** The evaluator will perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors and that the OS properly boots.
> - **Test 2:** The evaluator will modify a TSF executable that is part of the bootchain verified by the TSF (i.e. Not the first-stage bootloader) and attempt to boot. The evaluator will ensure that an integrity violation is triggered and the OS does not boot (Care must be taken so that the integrity violation is determined to be the cause of the failure to load the module, and not the fact that in such a way to invalidate the structure of the module.).
> - **Test 3:** If the ST author indicates that the integrity verification is performed using a public key, the evaluator will verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1.

39.1 Documentation Review activity

39.1.1 Findings

The evaluator has reviewed the section **6.6.4 Windows Platform Integrity and Code Integrity** of the ***Security Target*** document, which describes the different stages of the boot process, providing information about which are the main files loaded in each step of the boot chain.

This information has allowed the evaluator to design a testing plan, which will be used during the test activity. The evaluator has identified four different stages during the boot process:

- **First stage:** Secure boot checks the file integrity of early boot components, comparing them with the values stored in the TPM. Due to the fact that TPM is part of the external TOE environment, this stage will not be tested during the test activity.

© 2019 Microsoft Corporation. All rights reserved.

- **Second stage:** After the preliminary components have been loaded, the UEFI firmware loads the OS Boot Manager. Once the integrity of OS Boot Manager has been checked, it attempts to load one of these boot applications:
  - OS Loader: *winload.exe* or *winload.efi*
  - OS Resume: *winresume.exe* or *winresume.efi (The administrative guidance states that the hibernation is disabled, so this boot application will not be used during the evaluation)*
  - A physical memory testing application: *memtest.exe (The* **Security Target** *document states that it is considered a non-operational mode for the evaluation)*.

  In addition, a list with the critical loaded files during the bootchain when the *winload.exe* is selected has been included. These file are the following:

  - [text intentionally left blank]

- **Third stage:** Once the *winload.exe* or *winload.efi* file has been loaded and its integrity has been checked, the next step in the bootchain is loading the *ntoskrnl.exe* file. Additional critical drivers and libraries are loaded together with this file. The following information is also included regarding Code Integrity, which verifies the integrity of the kernel drivers loaded into the memory. For x64-based computers, all kernel-mode drivers must be digitally signed. If during the boot process an unsigned-driver is loaded, the operating system will not load. In addition, for a x86-based computer only the files listed above must be digitally signed. If any of these files are not signed, the operating system will not load. However, if another unsigned-driver is detected during the boot process, the operating system may be loaded normally.
- **Fourth stage:** After the critical device drivers and libraries have been loaded, the Windows kernel continues to boot the rest of the operating system.

The integrity validation mechanism is explained throughout this section, including information about how the TOE validates each piece of software using a hash based signature and an embedded public key.

Finally, this section provides the following information regarding the ability of the operating system to repair itself when a failure occurs during the boot process:

> *Should the Winload boot application be unable to validate the integrity of one of the Windows image files, the Winload boot application does not continue to load other Windows image files. Rather it displays an error message and fails into a non-operational mode. In limited circumstances the pre-boot environment will attempt to repair the boot environment, such as copying files from a repair partition to repair files with integrity errors. When repair is not possible, the boot manager will ask the user to reinstall Windows.*

© 2019 Microsoft Corporation. All rights reserved.

39.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

39.2 Test Activity

39.2.1 Test 1

The evaluator will observe that the integrity mechanism does not flag any executable with integrity errors and that the OS properly boots.

39.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- The *PcpTool* application should be available and executed on another computer (not the TOE).
- The boot measurements records need to be stored, to do that the following option related to the boot configuration should be applied via *Command Line Terminal*:

  > bcdedit –set {globalsettings} integrityservices enable

39.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- First, the evaluator shall execute a shutdown operation on the TOE.
- Then, a normal system startup shall be performed.
- After that, the *Windows Boot Configuration* Log file should be retrieved. This file is stored in *%windir%\logs\MeasuredBoot\xxx.log* (*where xxx corresponds to the name of the latest log*).
- Copy the log file to the computer with the *Pcptool utility*. Before being read, it is necessary to be decoded first. To do that the following command should be executed in a *Command Line Terminal*:

  > Pcptool.exe decodelog bootlog.log > decodedlog.log

- Open the *decoded* log. The bootloaders integrity measurements are maintained in the TPM's Platform Configuration Register (*PCR*) 13. The log should contain the following structure:

© 2019 Microsoft Corporation. All rights reserved.

```
<EV_Event_Tag PCR="13\"
    EventDigest="f9ed2a9d41ab75a4245737a828864bcc04eb953b99084d0bbf1db7c621b081cd"
    Size="916">
...
<LoadedModule_Aggregation ...\>
<FilePath Size="bootloader executable file size"\>
  bootloader executable filepath/name
</FilePath\>
...
\</EV_Event_Tag>
\windows\system32\<language\>winload.exe.MUI
\EFI\Microsoft\Boot\<language\>bootmgfw.efi.MUI
\windows\system32\hal.dll
\windows\system32\hvix64.exe
\windows\system32\hvloader.exe
\windows\system32\mcupdate\_GenuineIntel.dll
\windows\system32\Winload.efi
\windows\system32\Winresume.efi
\windows\boot\resources\bootres.dll
\windows\system32\BOOTVID.dll
\windows\system32\drivers\ACPI.sys
\windows\system32\drivers\acpiex.sys
\windows\system32\drivers\ApiSetSchema.dll
\windows\system32\drivers\CEA.sys
\windows\system32\drivers\CI.dll
\windows\system32\drivers\CLASSPNP.sys
\windows\system32\drivers\CLFS.sys
\windows\system32\drivers\clipsp.sys
\windows\system32\drivers\cmicext.sys
\windows\system32\drivers\cng.sys
\windows\system32\drivers\disk.sys
\windows\system32\drivers\EhStorClass.sys
\windows\system32\drivers\fileinfo.sys
\windows\system32\drivers\FLTMGR.SYS
\windows\system32\drivers\Fs\_Rec.sys
\windows\system32\drivers\fvevol.sys
\windows\system32\drivers\fwpkclng.sys
\windows\system32\drivers\hwpolicy.sys
\windows\system32\drivers\kd.dll
\windows\system32\drivers\ksecdd.sys
\windows\system32\drivers\ksecpkg.sys
\windows\system32\drivers\mcupdate\_GenuineIntell.dll
\windows\system32\drivers\mountmgr.sys
\windows\system32\drivers\msiadsrv.sys
\windows\system32\drivers\msrpc.sys
\windows\system32\drivers\mup.sys
\windows\system32\drivers\ndis.sys
\windows\system32\drivers\NETIO.SYS
\windows\system32\drivers\NTFS.sys
\windows\system32\drivers\ntosext.sys
\windows\system32\drivers\ntoskrnl.exe
\windows\system32\drivers\partmgr.sys
```

© 2019 Microsoft Corporation. All rights reserved.

```
\windows\system32\drivers\pci.sys
\windows\system32\drivers\pcw.sys
\windows\system32\drivers\pdc.sys
\windows\system32\drivers\PSHED.dll
\windows\system32\drivers\rdyboost.sys
\windows\system32\drivers\spaceport.sys
\windows\system32\drivers\starport.sys
\windows\system32\drivers\storahci.syhs
\windows\system32\drivers\tm.sys
\windows\system32\drivers\tpcip.sys
\windows\system32\drivers\tpm.sys
\windows\system32\drivers\vdvroot.sys
\windows\system32\drivers\Volmgr.sys
\windows\system32\drivers\volmgrx.sys
\windows\system32\drivers\volsnap.sys
\windows\system32\drivers\Wd01000.sys
\windows\system32\drivers\WdBoot.sys
\windows\system32\drivers\WdFilter.sys
\windows\system32\drivers\WDFLDR.SYS
\windows\system32\drivers\werkernel.sys
\windows\system32\drivers\Wfplwfs.sys
\windows\system32\drivers\WindowsTrustedRT.sys
\windows\system32\drivers\WindowsTrustedRTProxy.sys
\windows\system32\drivers\WMILIB.SYS
\windows\system32\drivers\Wof.sys
\windows\system32\drivers\WppRecorder.sys
```

It is implicit that the system has verified the integrity measurements indicated in the log, otherwise the system would not have booted as it is described in the documentation review activity.

39.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has obtained the same results in all tested platforms.

The evaluator has configured the OS to log the measurement boot records. After shutting down the machine and starting it again, the measured log has been analyzed in order to check that there is no integrity error.

39.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

39.2.2 Test 2

An executable file which is part of the bootchain will be modified and then an attempt to boot the TOE will be performed.

39.2.2.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A hexadecimal editor (e.g. *HxD*) must be installed in the evaluated platforms to modify the binary files loaded during the boot process.
- A *WinPE* USB for both architectures must be available.

39.2.2.2 Procedure

During the documentation review activity, the evaluator has identified four different stages during the boot process. The first one involves the TPM, which is part of the external TOE environment, it will not be tested.

For the remaining stages, the evaluator considers that the test should be performed using one of the critical files loaded during the bootchain and listed in the TSS section of the ***Security Target*** document. Any of these files can be used to test any evaluated platforms due to the digital signature is mandatory for all of them, regardless of the architecture of the operating system.

Therefore, the evaluator shall modify one of the files of the second stage listed in the TSS section (e.g. *ntosknrl.exe*) and after its modification, an attempt to boot shall be performed.

The following steps are necessary to perform this test activity:

- Boot the TOE and create a copy of the target file (e.g. *ntoskrnl.exe*). This file will be modified in below steps.
- Modify any bytes of the file using a hexadecimal editor. (e.g. *HxD*).

© 2019 Microsoft Corporation. All rights reserved.

ntoskrnl.exe

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  45 26 45 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  E&Eprogram canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  D2 4A 8C 38 96 2B E2 6B 96 2B E2 6B 96 2B E2 6B  ÒJŒ8–+âk–+âk–+âk
00000090  F3 4D E4 6A 9F 2B E2 6B F3 4D E3 6A 98 2B E2 6B  óMäjŸ+âkóMãj˜+âk
000000A0  96 2B E3 6B F7 2B E2 6B 96 2B E2 6B 17 2B E2 6B  –+ãk÷+âk–+âk.+âk
000000B0  9F 53 71 6B 8F 2B E2 6B F3 4D EF 6A 63 2F E2 6B  ŸSqk.+âkóMïjc/âk
000000C0  F3 4D E7 6A 8D 2B E2 6B F3 4D E6 6A EA 2B E2 6B  óMçj.+âkóMæjê+âk
000000D0  F3 4D E1 6A D3 2B E2 6B F3 4D E2 6A 97 2B E2 6B  óMájÓ+âkóMâj—+âk
000000E0  F3 4D 1D 6B 97 2B E2 6B F3 4D E0 6A 97 2B E2 6B  óM.k—+âkóMàj—+âk
```

- Boot the TOE into the Windows Pre Environment using the *WinPE* USB and replace the original file with the modified one. Verify that there are no other copies of the file in the partition where the OS is installed (e.g. C:\Windows and C:\Windows\WinSxS). Otherwise, the OS will use them to repair itself as pointed in the Documentation Activity of this requirement.
- Restart the TOE and observe the behaviour. The system shall not boot properly and an error screen shall be shown.
- Boot the TOE into the Windows Pre Environment using the *WinPE* USB drive, and restore the corrupt file using the backup created on the first step.
- Restart the TOE and observe that the system boot properly.

39.2.2.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has modified an executable that is part of the bootchain (*ntoskrnl.exe*) and attempt to boot. The OS detected that there was a corrupt file during the boot process. After that, the OS showed the *Automatic Repair* screen pointing out that the OS could not be repaired, also a log file is generated with details about the error produced.

The following screenshot shows the error screen for all Windows operating systems editions evaluated:

Additionally, the error log file has been retrieved:

```
Startup Repair diagnosis and repair log
------------------------------------
Last successful boot time: 12/6/2018 8:26:26 PM (GMT)
Number of repair attempts: 1

Session details
------------------------------------
System Disk = \Device\Harddisk0
Windows directory = C:\Windows
```

© 2019 Microsoft Corporation. All rights reserved.

```
AutoChk Run = 0
Number of root causes = 1

Test Performed:
-----------------------------
Name: Check for updates
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: System disk test
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Disk failure diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Disk metadata test
Result: Completed successfully. Error code = 0x0
Time taken = 93 ms

Test Performed:
-----------------------------
Name: Disk metadata test
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Target OS test
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Volume content check
Result: Completed successfully. Error code = 0x0
Time taken = 47 ms

Test Performed:
-----------------------------
Name: Boot manager diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
```

© 2019 Microsoft Corporation. All rights reserved.

```
-----------------------------
Name: System boot log diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Event log diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Internal state check
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Check for installed LCU
Result: Completed successfully. Error code = 0x0
Time taken = 2515 ms

Test Performed:
-----------------------------
Name: Check for pending package install
Result: Completed successfully. Error code = 0x0
Time taken = 1891 ms

Test Performed:
-----------------------------
Name: Boot status test
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Setup state check
Result: Completed successfully. Error code = 0x0
Time taken = 78 ms

Test Performed:
-----------------------------
Name: Registry hives test
Result: Completed successfully. Error code = 0x0
Time taken = 516 ms

Test Performed:
-----------------------------
Name: Windows boot log diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms
```

© 2019 Microsoft Corporation. All rights reserved.

```
Root cause found:
-----------------------------
Boot critical file c:\windows\system32\ntoskrnl.exe is corrupt.

Repair action: File repair
Result: Failed. Error code = 0x57
Time taken = 1188 ms

-----------------------------
-----------------------------
```

This record shows that the last boot could not be performed due to the modification of the *ntoskrnl.exe* file, proving that the *Integrity Violation* of the OS is working properly. The corrupted file was replaced with the original one, unmodified. After that, the boot process was successful.

On Windows Server operating system editions, when the original file is replaced with the corrupted one, the OS detects that there was a corrupt file during the boot process. Proving that the *Integrity Violation* of the OS is working properly. After that, the OS showed the *Recovery* screen without creating any log file.

Additionally, the evaluator has reviewed the log entries to check the event log related to the boot process. The following audit entry was created:

© 2019 Microsoft Corporation. All rights reserved.

```
EventID             : 20
MachineName         : SURFACEBOOK2
Data                : {}
Index               : 1037
Category            : (31)
CategoryNumber      : 31
EntryType           : Information
Message             : The description for Event ID '20' in Source 'Microsoft-Windows-Kernel-Boot' cannot be found.  The
                      local computer may not have the necessary registry information or message DLL files to display
                      the message, or you may not have permission to access them.  The following information is part of
                      the event:'false', 'false', '7', '2'
Source              : Microsoft-Windows-Kernel-Boot
ReplacementStrings  : {false, false, 7, 2}
InstanceId          : 20
TimeGenerated       : 12/6/2018 12:36:29 PM
TimeWritten         : 12/6/2018 12:36:29 PM
UserName            : NT AUTHORITY\SYSTEM
Site                :
Container           :
```

The string *'false'* on the event message means that the last shutdown and boot process event could not be performed. The event obtained through the Event Viewer tool shows the complete event message:

The corrupted file was replaced with the original one, unmodified. After that, the boot process was successful.

39.2.2.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

The evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

39.2.3 Test 3

To do this test, the evaluator shall have access to the required binaries signed by the vendor CA.

Due to this fact, this test shall be performed at Microsoft facilities.

The evaluator shall attempt to boot the TOE using an executable which has been signed with a certificate which does not have the *Code Signing* purpose. After that, the evaluator shall repeat the same procedure using an executable which has been signed with a certificate which has the *Code Signing* purpose.

39.2.3.1 Setup

To perform the test, the evaluator shall need a binary signed with a valid certificate and without *Code Signing* purpose in the *extendedKeyUsage* field. The vendor has provided the following binary files, which shall be used during the test execution.

- *cng.sys.codesigned*: This file has been signed using a certificate with *Code Signing* purpose in the *extendedKeyUsage* field as it can be shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

- *cng.sys.serverauthsigned*: This file has been signed using a certificate without the *extendedKeyUsage* field as it can be shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

- Additionally, a *WinPE* USB for both architectures (x64 & x86) must be available.

39.2.3.2 Procedure

The modified binary file during this procedure is *cng.sys*, which is stored in %windir%/system32/drivers. The evaluator shall carry out the following steps in order to replace the valid binary with the invalid one, and then attempt to boot.

- Run a *cmd* terminal as administrator and type the following command in order to enable the test signing mode:

  ```
  bcdedit /set testsigning on
  ```

  - By enabling this mode the TOE will load binaries which have not been signed by a trusted certification authority during the boot process.

© 2019 Microsoft Corporation. All rights reserved.

  - NOTE: The evaluator has identified that this command does not work properly for those platforms which the Secure Boot Policy is enabled in the *BIOS/UEFI*. To disable this policy, go to *BIOS/UEFI* and switch off the *Secure Boot* option.

- After that, create a copy of the valid binary. This copy will be restored in next steps.
- Boot the TOE into the *WinPE* and replace the valid binary with the one signed by a certificate without the *extendedKeyUsage* field. Verify that there are no other copies of the file in the partition where the OS is installed (e.g. C:\Windows and C:\Windows\WinSxS). Otherwise, the OS will use them to repair itself as pointed in the Documentation Activity of this requirement.
- Restart the TOE and observe that an integrity violation is triggered and the system does not boot properly.
- Boot the TOE again into the *WinPE*, and replace the bad binary with the one signed by a certificate with the *Code Signing* purpose in the *extendedKeyUsage* field. Restart the TOE and observe that the boot process is completed successfully.
- Finally in order to restore the original state of the tested platform, boot the TOE into the *WinPE* and replace the binary with the backup created at the second step. After that, restart the TOE, open a *Command Line Terminal* as administrator and type the following command in order to disable the test signing mode:

  ```
  bcdedit /set testsigning off
  ```

39.2.3.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has replaced the *cng.sys* binary file loaded during the boot process with one signed using a certificate without the *extendedKeyUsage* field. In this case, the TOE did not boot properly because an integrity error was triggered.

The operating system has detected the integrity error and has started an attempt to auto repair itself. The attempt has not finished properly and the following error screen was shown for all Windows 10 operating system editions evaluated:

A log file is generated in the C:\Windows\System32\LogFiles\Srt\SrtTrail.txt.

The content of this error log file is the following:

```
Startup Repair diagnosis and repair log
------------------------------------
Last successful boot time: 12/7/2018 8:30:19 PM (GMT)
Number of repair attempts: 1

Session details
------------------------------------
System Disk = \Device\Harddisk0
Windows directory = C:\Windows
AutoChk Run = 0
Number of root causes = 0
```

© 2019 Microsoft Corporation. All rights reserved.

```
Test Performed:
-----------------------------
Name: Check for updates
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: System disk test
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Disk failure diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Disk metadata test
Result: Completed successfully. Error code = 0x0
Time taken = 156 ms

Test Performed:
-----------------------------
Name: Disk metadata test
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Target OS test
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Volume content check
Result: Completed successfully. Error code = 0x0
Time taken = 78 ms

Test Performed:
-----------------------------
Name: Boot manager diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: System boot log diagnosis
Result: Completed successfully. Error code = 0x0
```

© 2019 Microsoft Corporation. All rights reserved.

```
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Event log diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 16 ms

Test Performed:
-----------------------------
Name: Internal state check
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Check for installed LCU
Result: Completed successfully. Error code = 0x0
Time taken = 2390 ms

Test Performed:
-----------------------------
Name: Check for pending package install
Result: Completed successfully. Error code = 0x0
Time taken = 1531 ms

Test Performed:
-----------------------------
Name: Boot status test
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
Name: Setup state check
Result: Completed successfully. Error code = 0x0
Time taken = 63 ms

Test Performed:
-----------------------------
Name: Registry hives test
Result: Completed successfully. Error code = 0x0
Time taken = 406 ms

Test Performed:
-----------------------------
Name: Windows boot log diagnosis
Result: Completed successfully. Error code = 0x0
Time taken = 0 ms

Test Performed:
-----------------------------
```

© 2019 Microsoft Corporation. All rights reserved.

```
Name: Bugcheck analysis
Result: Completed successfully. Error code =  0x0
Time taken = 110 ms

_____________________________
_____________________________
```

This record shows that the last boot could not be performed and the repair attempt failed due to the modification of the *cng.sys* file, proving that the *Integrity Violation* of the OS is working properly.

Finally, after replacing the bad binary, with the one signed by a certificate with the *Code Signing* value in the *extendedKeyUsage* field, the boot process has finished successfully.

On Windows Server operating system editions, when the original *cng.sys* file is replaced with the one withoud extensions, the OS detects that there was a corrupt file during the boot process. Proving that the *Integrity Violation* of the OS is working properly. After that, the OS showed the *Recovery* screen without creating any log file.

Additionally, the evaluator has reviewed the log entries to check the event log related to the boot process. The following audit entry was created:

```
EventID            : 20
MachineName        : DELL5290
Data               : {}
Index              : 7549
Category           : (31)
CategoryNumber     : 31
EntryType          : Information
Message            : The description for Event ID '20' in Source 'Microsoft-Windows-Kernel-Boot' cannot be found.  The local computer
                     may not have the necessary registry information or message DLL files to display the message, or you may not have
                     permission to access them.  The following information is part of the event:'false', 'false', '38', '2'
Source             : Microsoft-Windows-Kernel-Boot
ReplacementStrings : {false, false, 38, 2}
InstanceId         : 20
TimeGenerated      : 12/7/2018 12:38:30 PM
TimeWritten        : 12/7/2018 12:38:30 PM
UserName           : NT AUTHORITY\SYSTEM
Site               :
Container          :
```

© 2019 Microsoft Corporation. All rights reserved.

The string *'false'* on the event message means that the last shutdown and boot process event could not be performed. The event obtained through the Event Viewer tool shows the complete event message:

After replacing the previous bad binary with the one signed by a certificate with the *Code Signing* value in the *extendedKeyUsage* field, the boot process finished successfully.

39.2.3.4 Verdict

As the above results state, an integrity error is triggered if a binary file which has been signed using a certificate with an *extendedKeyUsage* value different from *Code Signing* is loaded during the boot process. Otherwise, the boot process is completed properly.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3**.

© 2019 Microsoft Corporation. All rights reserved.

39.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TST_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

40 FPT_TUD_EXT.1.1

The assurance activity for the **FPT_TUD_EXT.1.1** requirement is stated as follows:

> The evaluator will check for an update using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require installing and temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update. (The evaluator is also to ensure that this query occurs over a trusted channel as described in FTP_ITC_EXT.1.)

40.1 Documentation Review activity

40.1.1 Findings

The evaluator has reviewed the section **6.13 Managing Updates** of the ***Operational Guidance*** document, which describes the steps which should be followed to check for updates.

6.13.4 Checking for OS updates using the Windows UI

Role	Standard User
Windows Editions	Home, Pro, Enterprise, Server Standard
Related Assurance Activities	FPT_TUD.1:A:1

To check for Windows updates, follow these steps:

Open **Settings**
Navigate to **Update & security**
Click the Check for updates button.

In addition, there is a link to the Windows support website with a FAQ and instructions to keep the PC up to date. The following screenshot extracted from the ***Operational Guidance*** document, shows this information:

6.13 Managing updates

The following article provides an overview of Windows Update and matching FAQ list:

- Windows Update FAQ: https://support.microsoft.com/en-us/help/12373/windows-update-faq

© 2019 Microsoft Corporation. All rights reserved.

40.1.2 Verdict

The ***Operational Guidance*** document includes in its section **6.13 Managing Updates** enough information to allow the evaluator to determine how to check for new updates for both operating systems, Windows 10 and Windows Server.

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

40.2 Test Activity

40.2.1 Test 1

The evaluator shall ensure that the OS has the ability to check for updates for the operating system and the query is performed over a trusted channel.

40.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A network protocol analyzer (e.g. *Wireshark*).

40.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed to ensure that the update checking process is performed over a trusted channel as described in FTP_ITC_EXT.1:

- Open *Wireshark* and configure it to listen through the active Ethernet interface.
- Go to *Settings->Update & Security ->Windows Update*. Then click in *Check for updates*.
- After that, observe the network traffic in *Wireshark*. A secure connection between the TOE and the update server shall be established over a secure and trusted channel. This secure channel must be established as described in *FTP_ITC_EXT.1* requirement. A filter over the capture can be applied, showing only the packets related with the secure channel establishment. To do this, write *ssl* in the Filter text field in *Wireshark* and then click in *Apply* button.

ssl

No.	Time	Source	Destination	Protocol	Length	Info
16	0.751400	10.105.129.119	13.68.93.109	TLSv1.2	141	Application Data
17	0.751449	10.105.129.119	13.68.93.109	TLSv1.2	92	Application Data
18	0.751471	10.105.129.119	13.68.93.109	TLSv1.2	281	Application Data
20	0.828609	13.68.93.109	10.105.129.119	TLSv1.2	92	Application Data
35	0.831457	13.68.93.109	10.105.129.119	TLSv1.2	1440	Application Data [TCP segment of a reassembled PDU]
37	0.832267	13.68.93.109	10.105.129.119	TLSv1.2	213	Application Data
44	0.909308	13.68.93.109	10.105.129.119	TLSv1.2	359	Application Data, Application Data
47	0.991148	13.68.93.109	10.105.129.119	TLSv1.2	96	Application Data
52	1.033206	10.105.129.119	13.68.93.109	TLSv1.2	239	Client Hello
56	1.114201	13.68.93.109	10.105.129.119	TLSv1.2	1222	Server Hello, Certificate, Server Key Exchange, Server Hello Done

© 2019 Microsoft Corporation. All rights reserved.

40.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has initiated the update checking process and has checked that the connection between the TOE and the update server is established over a secure channel.

First of all, the evaluator has identified the cipher suites in the *Client Hello* packet. These cipher suites match with the ones defined in FCS_TLSC_EXT.1.

© 2019 Microsoft Corporation. All rights reserved.

After that, the server selects one of the cipher suites offered by the client machine. It could be observed in the *Server Hello* packet.

```
No.  Time      Source          Destination     Protocol  Length Info
52   1.033206  10.105.129.119  13.68.93.109    TLSv1.2    239   Client Hello
56   1.114201  13.68.93.109    10.105.129.119  TLSv1.2   1222   Server Hello, Certificate, Server Key Exchange, Server Hello Done
57   1.115870  10.105.129.119  13.68.93.109    TLSv1.2    180   Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
58   1.194095  13.68.93.109    10.105.129.119  TLSv1.2    105   Change Cipher Spec, Encrypted Handshake Message
59   1.194096  13.68.93.109    10.105.129.119  TLSv1.2    123   Application Data
61   1.195679  10.105.129.119  13.68.93.109    TLSv1.2    141   Application Data
62   1.195720  10.105.129.119  13.68.93.109    TLSv1.2     92   Application Data
63   1.195740  10.105.129.119  13.68.93.109    TLSv1.2    282   Application Data

▸ Frame 56: 1222 bytes on wire (9776 bits), 1222 bytes captured (9776 bits) on interface 0
▸ Ethernet II, Src: Cisco_46:02:d6 (e8:ed:f3:46:02:d6), Dst: Microsof_0e:a8:d4 (c4:9d:ed:0e:a8:d4)
▸ Internet Protocol Version 4, Src: 13.68.93.109, Dst: 10.105.129.119
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 49699, Seq: 2773, Ack: 186, Len: 1168
▸ [3 Reassembled TCP Segments (3940 bytes): #53(1386), #54(1386), #56(1168)]
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 3935
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 90
        Version: TLS 1.2 (0x0303)
      ▸ Random: 5c0826dcd5627ebe2821cad0dd0bc85872fff8b357783032...
        Session ID Length: 32
        Session ID: a91a0000446ff11c7a878ccc43e7cdec9c9e3fecab7d1308...
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
        Compression Method: null (0)
        Extensions Length: 18
      ▸ Extension: application_layer_protocol_negotiation (len=5)
      ▸ Extension: extended_master_secret (len=0)
      ▸ Extension: renegotiation_info (len=1)
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 3500
        Certificates Length: 3497
      ▸ Certificates (3497 bytes)
    ▾ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 329
      ▸ EC Diffie-Hellman Server Params
    ▾ Handshake Protocol: Server Hello Done
        Handshake Type: Server Hello Done (14)
        Length: 0
```

Afterwards, the server sends the certificate and its certificate chain.

```
No.  Time      Source          Destination     Protocol  Length Info
44   0.909308  13.68.93.109    10.105.129.119  TLSv1.2    359   Application Data, Application Data
47   0.991148  13.68.93.109    10.105.129.119  TLSv1.2     96   Application Data
52   1.033206  10.105.129.119  13.68.93.109    TLSv1.2    239   Client Hello
56   1.114201  13.68.93.109    10.105.129.119  TLSv1.2   1222   Server Hello, Certificate, Server Key Exchange, Server Hello Done
57   1.115870  10.105.129.119  13.68.93.109    TLSv1.2    180   Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
58   1.194095  13.68.93.109    10.105.129.119  TLSv1.2    105   Change Cipher Spec, Encrypted Handshake Message
59   1.194096  13.68.93.109    10.105.129.119  TLSv1.2    123   Application Data
61   1.195679  10.105.129.119  13.68.93.109    TLSv1.2    141   Application Data

    ▸ Extension: renegotiation_info (len=1)
  ▾ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 3500
      Certificates Length: 3497
    ▾ Certificates (3497 bytes)
      Certificate Length: 1695
    ▾ Certificate: 3082069b30820483a003020102021333000000f2396b4e10... (id-at-commonName=sls.update.microsoft.com,id-at-organizationalUnitName=DSP,id-at-organizationName=Microsoft,
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 0x33000000f2396b4e10f53958300000000000f2
        ▸ signature (sha256WithRSAEncryption)
        ▾ issuer: rdnSequence (0)
          ▸ rdnSequence: 5 items (id-at-commonName=Microsoft Update Secure Server CA 2.1,id-at-organizationName=Microsoft Corporation,id-at-localityName=Redmond,id-at-stateOrProvi
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▸ extensions: 8 items
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
        encrypted: 3dc32a4d973e0839e91a5648d6444c333b38a7cf9206db1b...
    Certificate Length: 1796
    ▾ Certificate: 308207003082048a003020102021333000000ab891a2c8... (id-at-commonName=Microsoft Update Secure Server CA 2.1,id-at-organizationName=Microsoft Corporation,id-at-lo
      ▾ signedCertificate
          version: v3 (2)
          serialNumber: 0x330000000ab891a2c80a50a5df00000000000a
        ▸ signature (sha256WithRSAEncryption)
        ▾ issuer: rdnSequence (0)
          ▸ rdnSequence: 5 items (id-at-commonName=Microsoft Root Certificate Authority 2011,id-at-organizationName=Microsoft Corporation,id-at-localityName=Redmond,id-at-stateOrPr
        ▸ validity
        ▸ subject: rdnSequence (0)
        ▸ subjectPublicKeyInfo
        ▸ extensions: 9 items
      ▸ algorithmIdentifier (sha256WithRSAEncryption)
        Padding: 0
        encrypted: a29bee036b47dbe99e495b2755ab466d185ad108158f849a...
  ▸ Handshake Protocol: Server Key Exchange
  ▸ Handshake Protocol: Server Hello Done
```

The root CA of the certificate chain sent by the server matches with one of the root certificates stored in the trusted certificates store of the client machine.

© 2019 Microsoft Corporation. All rights reserved.

40.2.1.4 Verdict

The evaluator has verified that the connection between the client and the update server is performed over a trusted channel, which it has been established as described in FTP_ITC_EXT.1 and FCS_TLSC_EXT.1

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

40.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TUD_EXT.1.1.

© 2019 Microsoft Corporation. All rights reserved.

41 FPT_TUD_EXT.1.2

The assurance activity for the **FPT_TUD_EXT.1.2** requirement is stated as follows:

> For the following tests, the evaluator will initiate the download of an update and capture the update prior to installation. The download could originate from the vendor's website, an enterprise-hosted update repository, or another system (e.g. network peer). All supported origins for the update must be indicated in the TSS and evaluated.
>
> - **Test 1:** The evaluator will ensure that the update has a digital signature belonging to the vendor prior to its installation. The evaluator will modify the downloaded update in such a way that the digital signature is no longer valid. The evaluator will then attempt to install the modified update. The evaluator will ensure that the OS does not install the modified update.
> - **Test 2:** The evaluator will ensure that the update has a digital signature belonging to the vendor. The evaluator will then attempt to install the update (or permit installation to continue). The evaluator will ensure that the OS successfully installs the update.

41.1 Documentation Review activity

41.1.1 Findings

The evaluator has reviewed the section **6.6.5 Windows and Application Updates** of the ***Security Target*** document, which is related to how the updates are provided by the vendor.

This section states that the operating system updates are delivered through the *Windows Update* capability, which is enabled by default. Additionally, this section also states that the user can obtain update files by visiting the following vendor website:

- Windows Update Catalog

Moreover, the ***Security Target*** document, includes the following information in its section **6.6.5.2 Distributing updates**, pointing again that the origin supported by the OS for system updates is the Windows Update.

> ***6.6.5.2 Distributing updates***
>
> There are several distribution channels for updates to Windows and Windows applications:
>
> - Windows Update: Windows Update is the web service for delivering Windows updates to directly to consumers.
> - Windows Server Update Services (WSUS): WSUS is a server role in Windows Server which IT administrators can use to distribute application updates to users within their enterprise.
> - Windows Store: The Windows Store is a web service for delivering updates to Universal Windows Platform apps which were originally installed from the Windows Store.

© 2019 Microsoft Corporation. All rights reserved.

41.1.2 Verdict

The ***Security Target*** document includes enough information in its TSS section to allow the evaluator to know how the updates are provided by the vendor and how to obtain an update file from the vendor website.

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

41.2 Test Activity

To do this test, the evaluator shall have access to the Wincab tool that allows to manipulate *MSU* packages.

Due to this fact, this test shall be performed at Microsoft facilities.

41.2.1 Test 1

An attempt to install an update with a non-valid digital signature will be performed during this test case.

41.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- The *Wincab* tool provided by the OS vendor which allows the evaluator to pack and unpack the *MSU* update file and extract its content.
- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

41.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed (the below procedure applies both for Windows 10 and Windows Server):

Obtain an update file:

The following steps must be performed to download an update file:

- Open Microsoft Edge and go to: Windows Update Catalog
- In the search box type *'Windows 10 1809'* or *'Windows Server 2019'*, a list of available updates will be shown. The applicable updates for Windows 10 October 2018 Update for x64-based system are not applicable for Windows Server 2019 version 1809, they must be downloaded separately.

© 2019 Microsoft Corporation. All rights reserved.

- Choose one update from the list, ensuring that the selected one is valid for the architecture of the platform which is under testing. Once the operating system and the architecture of the update file have been checked, click the *Download* button.
- A new window will be opened, click on the hyperlink to start the download and wait until the download has finished. The downloaded file shall have the *.msu* extension (*Microsoft Update Standalone Package*).

Modifying some parts of the file:

Once the update file has been downloaded and taking into account the information provided by the vendor regarding the internal structure of the MSU files, the evaluator has designed two tests to ensure that the integrity checking is applied over the different files inside the MSU package.

The vendor has provided the *WinCab* tool, which allows the evaluator to pack and unpack a MSU file and extract its content.

To extract the content of a *MSU* file into a folder the following command shall be typed in a *Command Line Terminal*:

```
wincab /folder <folder path> /extractcab <path to MSU>
```

To pack the contents of a folder into a *.cab* file, the following command shall be typed in a *Command Line Terminal*:

```
wincab /folder <folder path> /createcab <path to resulting CAB file>
```

The evaluator has designed the following two tests, which shall be performed using the *WinCab* tool:

- **Test 1-1**: Unpack the MSU file, modify the content of the xml description file, pack the files and attempt to install the update. The update shall be installed properly since the xml file is not signed.
- **Test 1-2:** Unpack the MSU file, modify the content of the cab file (this file contains the payload), pack the files and attempt to install the update. The update shall not be installed since the signature verification fails.

The evaluator has developed a script to automate the modification of the update packages as decribed above.

- Run as administrator, the script *FPT_TUD_EXT.1 - Test 1.ps1*. The script behaviour is explained as follows:
  - A new folder will be created. The evaluator shall copy the original update files (with *.msu* extension) in the folder named '*Input\Original*' and separate them based on their architecture (x86,x64 and Server). After that, the script will perform the modification of the update packages as described in '*Test1-1*' and '*Test1-2*'. Two update packages will be created with the previously described modifications applied. In the '*Modified*' folder is located the update package for '*Test1-1*' and in '*Invalid*' the update package for '*Test1-2*'
  - The script source code is as follows:

© 2019 Microsoft Corporation. All rights reserved.

```
Begin {

# Path names
$folderpath = "Input"
$invalidPath = "Invalid"
$modifiedPath = "Modified"
$originalPath = "Original"
$unpackedPath = "Unpacked"

# Creates the folder and subfolder
function createArchFolder ([string] $path, [string[]] $architecture) {
    $valid = "ARM64","X86","X64","SERVER"
    If(!(Test-Path $path)) {
        New-Item -ItemType Directory -Force -Path $path | Out-Null
        Set-Location $path
    } else {
        Write-Warning "[E]: Directory $path already exists..."
    }
    ForEach ($arch in $architecture){
        if ($valid.Contains($arch)){
            If(!(Test-Path $arch)) {
                New-Item -ItemType Directory -Force -Path $arch | Out-Null
            } else {
                Write-Warning "[E]: Directory $arch already exists..."
            }
        } else {
            Write-Warning "[E]: Architecture '$arch' not valid..."
        }
    }
    Set-Location ..
}

function createDirectories () {
    #Step 1 - Create directories
    $arch = Read-Host 'Type the supported architectures separated by coma
        (ARM64, x86, x64, Server)'
    $architecture = $arch.replace(' ','').split(",").toUpper()
    If($architecture){
        If(!(Test-Path $folderpath)) {
            New-Item -ItemType Directory -Force -Path $folderpath | Out-Null
            Set-Location -Path $folderpath
        } else {
            Write-Warning "[E]: Directory $folderpath already exists..."
        }
    }

    # Invalid Updates
    createArchFolder $invalidPath $architecture
    # Modified Updates
    createArchFolder $modifiedPath $architecture
    # Original Updates
    createArchFolder $originalPath $architecture
```

© 2019 Microsoft Corporation. All rights reserved.

```
    # Unpacked Updates
    createArchFolder $unpackedPath $architecture
}

function extractUpdates () {
#Step 2 - Extract
    Set-Location $originalPath
    If ((Test-Path "X64")) {
        $updateFilex64 =  Get-ChildItem .\x64\
        $updateFilex64 =  $updateFilex64.Name
        ..\..\wincab\Wincab.exe /folder "`..\Unpacked\x64\" /extractcab
            "`.\x64\$updateFilex64"
        Copy-Item ..\Unpacked\x64\* ..\Modified\x64
        Copy-Item ..\Unpacked\x64\* ..\Invalid\x64
    } else {
        Write-Warning "[I]: Update file for 'x64' platform does not exist or
            cannot be found. Skipping it."
    }
    If ((Test-Path "X86")) {
        $updateFilex86 =  Get-ChildItem .\x86\
        $updateFilex86 = $updateFilex86.Name
        ..\..\wincab\Wincab.exe /folder "`..\Unpacked\x86\" /extractcab
            "`.\x86\$updateFilex86"
        Copy-Item ..\Unpacked\x86\* ..\Modified\x86
        Copy-Item ..\Unpacked\x86\* ..\Invalid\x86
    } else {
        Write-Warning "[I]: Update file for 'x86' platform does not exist or
            cannot be found. Skipping it."
    }
    If ((Test-Path "SERVER")) {
        $updateFileServer =  Get-ChildItem .\Server\
        $updateFileServer = $updateFileServer.Name
        ..\..\wincab\Wincab.exe /folder "`..\Unpacked\Server\" /extractcab
            "`.\Server\$updateFileServer"
        Copy-Item ..\Unpacked\Server\* ..\Modified\Server
        Copy-Item ..\Unpacked\Server\* ..\Invalid\Server
    } else {
        Write-Warning "[I]: Update file for 'Server' platform does not exist
            or cannot be found. Skipping it."
    }
    If ((Test-Path "ARM64")) {
        $updateFileARM64 =  Get-ChildItem .\ARM64\
        $updateFileARM64 = $updateFileARM64.Name
        ..\..\wincab\Wincab.exe /folder "`..\Unpacked\arm64\" /extractcab
            "`.\arm64\$updateFileARM64"
        Copy-Item ..\Unpacked\arm64\* ..\Modified\arm64
        Copy-Item ..\Unpacked\arm64\* ..\Invalid\arm64
    } else {
        Write-Warning "[I]: Update file for 'ARM64' platform does not exist or
            cannot be found. Skipping it."
    }
    Set-Location ..
```

© 2019 Microsoft Corporation. All rights reserved.

```
}

function modifyBuildTXT ([string]$arch, [int]$posoffset ) {
    Set-Location $modifiedPath\$arch
    $filetohexedit = Get-ChildItem *.txt

    (Get-Content $filetohexedit.PSPath) -replace('KB Article Number="(.*)"',
        'KB Article Number="7777777"') | Set-Content $filetohexedit.PSPath

    Set-Location ../..
}

function modifyCAB ([string]$arch) {
#Change ofset A00 to E&E (hex) - Invalid
    Set-Location $invalidPath\$arch
    $filetohexedit = Get-ChildItem Win*.cab

    $bytes = [System.IO.File]::ReadAllBytes($filetohexedit.DirectoryName+'\'+
    $filetohexedit.Name)
    $offset = 2560

    $bytes[$offset] = 0x45 #E on hex
    $bytes[$offset + 1] = 0x26 # & on hex
    $bytes[$offset + 2] = 0x45

    [System.IO.File]::WriteAllBytes($filetohexedit, $bytes)
    Set-Location ../..
}

function packUpdates () {
# Step 3 - Pack #1
    Write-Warning "[I]: Creating update packages with the modifications in the
        unverified part..."
    If((Test-Path "`Modified\X64\")) {
        ..\wincab\Wincab.exe /folder "`Modified\x64\" /createcab
            "`Modified\x64\Modifiedx64.cab"
        Rename-Item -Path 'Modified\x64\Modifiedx64.cab' -NewName
            'Modifiedx64.msu'
    } else {
        Write-Warning "[I]: Update file for 'x64' platform does not exist or
            cannot be found. Skipping it."
    }
    If((Test-Path "`Modified\X86\")) {
        ..\wincab\Wincab.exe /folder "`Modified\x86\" /createcab
            "`Modified\x86\Modifiedx86.cab"
        Rename-Item -Path 'Modified\x86\Modifiedx86.cab' -NewName
            'Modifiedx86.msu'
    } else {
        Write-Warning "[I]: Update file for 'x86' platform does not exist or
            cannot be found. Skipping it."
    }
    If((Test-Path "`Modified\SERVER\")) {
```

© 2019 Microsoft Corporation. All rights reserved.

```
        ..\wincab\Wincab.exe /folder "`Modified\Server\" /createcab
            "`Modified\Server\ModifiedServer.cab"
        Rename-Item -Path 'Modified\Server\ModifiedServer.cab' -NewName
            'ModifiedServer.msu'
    } else {
        Write-Warning "[I]: Update file for 'Server' platform does not exist
            or cannot be found. Skipping it."
    }
    If ((Test-Path "`Modified\ARM64\")) {
        ..\wincab\Wincab.exe /folder "`Modified\ARM64\" /createcab
            "`Modified\ARM64\ModifiedARM64.cab"
        Rename-Item -Path 'Modified\ARM64\ModifiedARM64.cab' -NewName
            'ModifiedARM64.msu'
    } else {
        Write-Warning "[I]: Update file for 'ARM64' platform does not exist or
            cannot be found. Skipping it."
    }

    Write-Warning "[I]: Creating update packages with the modifications in the
        verified part..."
    If ((Test-Path "`Modified\X64\")) {
        ..\wincab\Wincab.exe /folder "`Invalid\x64\" /createcab
            "`Invalid\x64\Invalidx64.cab"
        Rename-Item -Path 'Invalid\x64\Invalidx64.cab' -NewName
            'Invalidx64.msu'
    } else {
        Write-Warning "[I]: Update file for 'x64' platform does not exist or
            cannot be found. Skipping it."
    }
    If ((Test-Path "`Modified\x86\")) {
        ..\wincab\Wincab.exe /folder "`Invalid\x86\" /createcab
            "`Invalid\x86\Invalidx86.cab"
        Rename-Item -Path 'Invalid\x86\Invalidx86.cab' -NewName
            'Invalidx86.msu'
    } else {
        Write-Warning "[I]: Update file for 'x86' platform does not exist or
            cannot be found. Skipping it."
    }
    If ((Test-Path "`Modified\SERVER\")) {
        ..\wincab\Wincab.exe /folder "`Invalid\Server\" /createcab
            "`Invalid\Server\InvalidServer.cab"
        Rename-Item -Path 'Invalid\Server\InvalidServer.cab' -NewName
            'InvalidServer.msu'
    } else {
        Write-Warning "[I]: Update file for 'Server' platform does not exist
            or cannot be found. Skipping it."
    }
    If ((Test-Path "`Modified\ARM64\")) {
        ..\wincab\Wincab.exe /folder "`Invalid\ARM64\" /createcab
            "`Invalid\ARM64\InvalidARM64.cab"
        Rename-Item -Path 'Invalid\ARM64\InvalidARM64.cab' -NewName
            'InvalidARM64.msu'
```

© 2019 Microsoft Corporation. All rights reserved.

```
    } else {
        Write-Warning "[I]: Update file for 'ARM64' platform does not exist or
            cannot be found. Skipping it."
    }
    Set-Location ..
    }
}

Process {
Write-Output "[I]: Creating directories..."
createDirectories
Write-Warning "[I]: Place the original system updates that will be used for
    both architectures, then press Enter key to continue"
Pause
Write-Warning "[I]: Extracting updates..."
extractUpdates
Write-Warning "[I]: Modifying the updates..."
If((Test-Path "`Modified\x64\")) {
    modifyBuildTXT "x64" "338"
    modifyCAB "x64"
}else{
    Write-Warning "[I]: Update file for 'x64' platform does not exist or
        cannot be found. Skipping it."
}
If((Test-Path "`Modified\x64\")) {
    modifyBuildTXT "x86" "293"
    modifyCAB "x86"
}
else{
    Write-Warning "[I]: Update file for 'x86' platform does not exist or
        cannot be found. Skipping it."
}
If((Test-Path "`Modified\Server\")) {
    modifyBuildTXT "Server" "293"
    modifyCAB "Server"
}
else{
    Write-Warning "[I]: Update file for 'Server' platform does not exist or
        cannot be found. Skipping it."
}
If((Test-Path "`Modified\ARM64\")) {
    modifyBuildTXT "ARM64" "293"
    modifyCAB "ARM64"
}
else{
    Write-Warning "[I]: Update file for 'ARM64' platform does not exist or
        cannot be found. Skipping it."
}
Write-Warning "[I]: Packing updates..."
packUpdates
}
```

© 2019 Microsoft Corporation. All rights reserved.

```
End {
    #Clean
}
```

41.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has performed the test defined in the Procedure section and has obtained the following results:

For Test 1-1: The evaluator has obtained the same results for all the platforms. After modifying the target file (*xml description file*) the update has been installed successfully as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the modification of not signed files of the MSU package does not affect the update installation, since it is performed over non critical files.

For Test 1-2: The evaluator has also obtained the same results for all the platforms. The evaluator has modified the *.cab* file, which contains the payload of the update, and has attempted to install the update without success.

During the installation of the update, the operating system has performed a digital signature validation. The verification failed and , as result, the following error message was shown:

Therefore, the modification of the *.cab* file affects the update installation, since it is performed over a critical file.

© 2019 Microsoft Corporation. All rights reserved.

41.2.1.4 Verdict

After analyzing the obtained results, the evaluator considers that the update file can not be installed properly, when the modified file is used and verified during the installation. In the cases where the modified file is not used during the installation or its integrity is not validated, the update can be applied successfully.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

41.2.2 Test 2

An installation of one update with a valid digital signature belonging to the vendor will be performed during this test case. The update shall be installed successfully.

41.2.2.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- The original update file downloaded for *Test 1* is available.

41.2.2.2 Procedure

The evaluator shall attempt to install the original update file downloaded in *Test 1*. The evaluator shall observe that the operating system successfully installs the update.

41.2.2.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has attempted to install an update file which has a valid digital signature belonging to the OS vendor. The installation was successful as it can be observed in the following images:

41.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to **Test 2**.

41.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TUD_EXT.1.2.

© 2019 Microsoft Corporation. All rights reserved.

42 FPT_TUD_EXT.2.1

The assurance activity for the **FPT_TUD_EXT.2.1** requirement is stated as follows:

> The evaluator will check for updates to application software using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update. (The evaluator is also to ensure that this query occurs over a trusted channel as described in FTP_ITC_EXT.1.)

42.1 Documentation Review activity

42.1.1 Findings

The evaluator has reviewed the section **6.13 Managing Updates** of the ***Operational Guidance*** document, which provides information regarding how to check for application updates in both operating systems (Windows 10 and Windows Server 2019).

Regarding Windows 10, the vendor provides the following link to the vendor support website, in which the procedure to check for application updates through the *Windows Store* is described.

6.13.6 Checking for Windows Store application updates

Role	Standard User
Windows Editions	Home, Pro, Enterprise
Related Assurance Activities	FPT_TUD.2:A:1

The following article describes how to check for updates to applications installed from the Windows Store:

- Check for updates for apps and games from Windows Store: https://support.microsoft.com/en-us/help/4026259/microsoft-store-check-updates-for-apps-and-games

The process, summarised, is the following:

- Go to *Start*, then open *Microsoft Store*. Click the *More ...* (*three dots*) button near the search box and then select *Downloads & Updates*. On the *Downloads and Updates* page, click the *Get updates* button.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

Regarding Windows Server 2019, the vendor has provided the following link, describing the necessary steps to check for updates. This section also includes a link explaining the required infrastructure (*Windows Server Update Service*)

> 6.13 Managing updates
>
> The following article provides an overview of Windows Update and matching FAQ list:
>
> - Windows Update FAQ: https://support.microsoft.com/en-us/help/12373/windows-update-faq

42.1.2 Verdict

The ***Operational Guidance*** document, includes in its section **6.13 Managing Updates** two pointers to the vendor support website.

The first one, provides enough information to allow the evaluator to determine how to check for application updates through *Windows Store* in Windows 10 October 2018 Update operating system.

The second one, explains the required infrastructure needed to check for updates of applications using *Windows Server Update Service*, since the *Windows Store* is not available for any edition of Windows Server 2019.

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

42.2 Test Activity

42.2.1 Test 1

The evaluator shall ensure that the OS has the ability to check for application updates and the query is performed over a trusted channel.

42.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

- A network protocol analyzer (e.g. *Wireshark*)
- (*Optional but recommended*) Microsoft Windows Store app must be configured with a Microsoft account before checking for updates.

A different approach was taken to test this requirement in Windows Server 2019 since the Windows Store is not available for any edition of Windows Server 2019. The vendor has provided

© 2019 Microsoft Corporation. All rights reserved.

a Hyper-V virtual machine infrastructure to test application updates via the Windows Software Update Services (WSUS). This infrastructure is composed of:

- Domain controller machine (Windows Server 2016 Datacenter)
- Certificate Authority machine (Windows Server 2016 Datacenter)
- WSUS Server machine (Windows Server 2016 Datacenter)
- The Target of Evaluation (Windows Server 2016 Standard)

The vendor has delivered this environment preconfigured so the application updates are automatically downloaded onto the TOE client machine (Windows Server 2019) as soon as it is joined to the domain.

42.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed to ensure that the update checking process is done over a trusted channel as described in FPT_ITC_EXT.1:

- Open *Wireshark* and configure it to listen through the active *Ethernet* interface.
- Open the *Windows Store* app and follow the steps identified in the Documentation Review activity section to check for application updates.
- After that, observe the network traffic in *Wireshark*. A secure connection between the TOE and the update server shall be established over a secure and trusted channel. This secure channel must be established as described in *FTP_ITC_EXT.1*. A filter over the capture can be applied, showing only the packets related with the secure channel establishment. To do this, write *ssl* in the *Filter* text field in *Wireshark* and then click in *Apply* button.

ssl

No.	Time	Source	Destination	Protocol	Length	Info
803	5.104424	10.105.129.82	52.114.132.21	TLSv1.2	1023	Application Data
804	5.104868	10.105.129.82	52.114.132.21	TLSv1.2	729	Application Data
805	5.143368	64.4.54.50	10.105.129.82	TLSv1.2	1239	Application Data
810	5.181559	52.114.132.21	10.105.129.82	TLSv1.2	413	Application Data
819	5.657848	10.105.129.82	65.52.108.90	TLSv1.2	244	Client Hello
823	5.741327	65.52.108.90	10.105.129.82	TLSv1.2	1101	Server Hello, Certificate, Server Key Exchange, Server Hello Done
824	5.778935	10.105.129.82	65.52.108.90	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

To complete this rest assurance activity for Windows Server 2019, the following steps shall be performed.

- Add the TOE Windows Server 2019 instance to the domain WINCC.
- Go to *Settings->Update & Security ->Windows Update*. Then click in *Check for updates*.
- After that, observe the network traffic in *Wireshark*. A secure connection between the TOE and the update server shall be established over a secure and trusted channel. This secure channel must be established as described in *FTP_ITC_EXT.1* requirement. A filter over the capture can be applied, showing only the packets related with the secure channel establishment. To do this, write *ssl* in the Filter text field in *Wireshark* and then click in *Apply* button.

42.2.1.3 Results

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has initiated the update checking process and has checked that the connection between the TOE and the update server is established over a secure channel.

The evaluator has identified the handshake messages between the client and the server to configure and establish the secure channel. First of all, the evaluator has identified the cipher suites selected by the client, which matches with the ones defined in FCS_TLSC_EXT.1.

No.	Time	Source	Destination	Protocol	Length	Info
819	5.657848	10.105.129.82	65.52.108.90	TLSv1.2	244	Client Hello
820	5.740687	65.52.108.90	10.105.129.82	TCP	1440	443 → 49766 [ACK] Seq=1 Ack=191 Win=263168 Len=1386 [TCP segment of a reassembled PDU]
821	5.740776	65.52.108.90	10.105.129.82	TCP	1440	443 → 49766 [ACK] Seq=1387 Ack=191 Win=263168 Len=1386 [TCP segment of a reassembled PDU]
822	5.740805	10.105.129.82	65.52.108.90	TCP	54	49766 → 443 [ACK] Seq=191 Ack=2773 Win=131584 Len=0
823	5.741327	65.52.108.90	10.105.129.82	TLSv1.2	1101	Server Hello, Certificate, Server Key Exchange, Server Hello Done
824	5.778935	10.105.129.82	65.52.108.90	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
825	5.859134	65.52.108.90	10.105.129.82	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message

```
▸ Frame 819: 244 bytes on wire (1952 bits), 244 bytes captured (1952 bits) on interface 0
▸ Ethernet II, Src: SamsungE_c1:09:72 (ac:af:b9:c1:09:72), Dst: IETF-VRRP-VRID_67 (00:00:5e:00:01:67)
▸ Internet Protocol Version 4, Src: 10.105.129.82, Dst: 65.52.108.90
▸ Transmission Control Protocol, Src Port: 49766, Dst Port: 443, Seq: 1, Ack: 1, Len: 190
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 185
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 181
        Version: TLS 1.2 (0x0303)
      ▸ Random: 5c100fb84780b7c04779f1fab61b6930d29c704e1020d226...
        Session ID Length: 0
        Cipher Suites Length: 26
      ▾ Cipher Suites (13 suites)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
          Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
        Compression Methods Length: 1
      ▸ Compression Methods (1 method)
```

Afterwards, the server sends the certificate and its certificate chain.

© 2019 Microsoft Corporation. All rights reserved.

The root CA of the certificate chain sent by the server match with one of the root certificate stored in the Trusted Certificate store on the client machine.

When testing Windows Server 2019, it could be observed how the update information was sent using TLS v.1.2 after the TOE was joined to the domain and the query for updates was performed.

In addition, the cipher suites presented by the client are the ones configured according to the ***Operational Guidance*** document, proving that the operating system mechanism is effectively using the evaluated configuration:

© 2019 Microsoft Corporation. All rights reserved.

```
No.  Time        Source          Destination     Protocol Length Info
38   15.920183   192.168.1.109   192.168.1.106   TLSv1.2  240    Client Hello
41   15.923263   192.168.1.106   192.168.1.109   TLSv1.2   98    Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
47   18.593521   192.168.1.109   192.168.1.106   TLSv1.2  180    Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
48   18.595625   192.168.1.106   192.168.1.109   TLSv1.2  105    Change Cipher Spec, Encrypted Handshake Message
49   18.595845   192.168.1.106   192.168.1.109   TLSv1.2  123    Application Data
51   18.610037   192.168.1.109   192.168.1.106   TLSv1.2  141    Application Data
52   18.610146   192.168.1.109   192.168.1.106   TLSv1.2  321    Application Data

    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 181
  ▾ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 177
      Version: TLS 1.2 (0x0303)
    ▸ Random: 5c12a9da71b81a1967f7bea0b99233b029cea420a798e670...
      Session ID Length: 0
      Cipher Suites Length: 26
    ▾ Cipher Suites (13 suites)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
        Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Compression Methods Length: 1
    ▸ Compression Methods (1 method)
      Extensions Length: 110
    ▸ Extension: server_name (len=24)
    ▸ Extension: status_request (len=5)
    ▸ Extension: supported_groups (len=8)
    ▸ Extension: ec_point_formats (len=2)
```

Moreover, the server sends the certificate and its certificate chain. This certificate belongs to the Certificate Authority machine described in the setup section.

```
No.  Time        Source          Destination     Protocol Length Info
38   15.920183   192.168.1.109   192.168.1.106   TLSv1.2  240    Client Hello
41   15.923263   192.168.1.106   192.168.1.109   TLSv1.2   98    Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
47   18.593521   192.168.1.109   192.168.1.106   TLSv1.2  180    Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
48   18.595625   192.168.1.106   192.168.1.109   TLSv1.2  105    Change Cipher Spec, Encrypted Handshake Message
49   18.595845   192.168.1.106   192.168.1.109   TLSv1.2  123    Application Data
51   18.610037   192.168.1.109   192.168.1.106   TLSv1.2  141    Application Data
52   18.610146   192.168.1.109   192.168.1.106   TLSv1.2  321    Application Data

▸ Frame 41: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: Microsof_94:a3:0f (00:15:5d:94:a3:0f), Dst: Dell_48:f4:ec (14:18:77:48:f4:ec)
▸ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.109
▸ Transmission Control Protocol, Src Port: 8531, Dst Port: 49827, Seq: 2921, Ack: 187, Len: 44
▸ [3 Reassembled TCP Segments (2964 bytes): #39(1460), #40(1460), #41(44)]
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 2959
    ▸ Handshake Protocol: Server Hello
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1001
        Certificates Length: 998
      ▾ Certificates (998 bytes)
          Certificate Length: 995
        ▾ Certificate: 308203df308202c7a0030201020213600000091e0d20986... (id-at-commonName=wincc-wsus.wincc.us)
          ▾ signedCertificate
              version: v3 (2)
              serialNumber: 0x6000000091e0d209867b107e9c000000000091
            ▸ signature (sha256WithRSAEncryption)
            ▾ issuer: rdnSequence (0)
              ▸ rdnSequence: 3 items (id-at-commonName=wincc-WINCC-CA-CA,dc=wincc,dc=us)
            ▸ validity
            ▸ subject: rdnSequence (0)
            ▸ subjectPublicKeyInfo
            ▸ extensions: 7 items
          ▸ algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: 3c01a6ae7f679e541b94d0904414da3fe1f3bccdb2e3323b...
```

Once the updates have been queried and installed by the Settings application, the following information is obtained in the TOE, showing which software applications have been installed:

© 2019 Microsoft Corporation. All rights reserved.

42.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfilment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

42.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FPT_TUD_EXT.2.1.

© 2019 Microsoft Corporation. All rights reserved.

43 FPT_TUD_EXT.2.2

The assurance activity for the **FPT_TUD_EXT.2.2** requirement is stated as follows:

> The evaluator will initiate an update to an application. This may vary depending on the application, but it could be through the application vendor's website, a commercial app store, or another system. All origins supported by the OS must be indicated in the TSS and evaluated. However, this only includes those mechanisms for which the OS is providing a trusted installation and update functionality. It does not include user or administrator-driven download and installation of arbitrary files.
>
> - **Test 1:** The evaluator will ensure that the update has a digital signature which chains to the OS vendor or another trusted root managed through the OS. The evaluator will modify the downloaded update in such a way that the digital signature is no longer valid. The evaluator will then attempt to install the modified update. The evaluator will ensure that the OS does not install the modified update.
> - **Test 2:** The evaluator will ensure that the update has a digital signature belonging to the OS vendor or another trusted root managed through the OS. The evaluator will then attempt to install the update. The evaluator will ensure that the OS successfully installs the update.

43.1 Documentation Review activity

43.1.1 Findings

The evaluator has reviewed the section **6.6.5.1 Windows Store Applications** of the ***Security Target*** document. The information provided explains how the application packages are verified during its installation using a digital signature from the vendor with the *Code Signing* usage.

The applications are contained in *AppX* packages following the *OPC* standard, whose internal structure are explained in detail in the ***Security Target*** document within its section **10. Appendix B: AppX Package Implementation**.

In addition, the ***Security Target*** document, includes the following information in its section **6.6.5.2 Distributing updates** regarding the origins supported by the operating systems for application updates:

© 2019 Microsoft Corporation. All rights reserved.

6.6.5.2 Distributing updates

There are several distribution channels for updates to Windows and Windows applications:

- Windows Update: Windows Update is the web service for delivering Windows updates to directly to consumers.
- Windows Server Update Services (WSUS): WSUS is a server role in Windows Server which IT administrators can use to distribute application updates to users within their enterprise.
- Windows Store: The Windows Store is a web service for delivering updates to Universal Windows Platform apps which were originally installed from the Windows Store.

Summarizing, the application updates can be distributed through the following channel:

- Windows Store: this method supported by Windows 10 operating system, allows to get update of UWP apps previously installed from it.
- Windows Server Update Services: this method supported by Windows Server 2019 is used by IT administrator to distribute application updates.

43.1.2 Verdict

The evaluator has reviewed the information provided in the TSS section of the ***Security Target*** document. This information allows the evaluator to identify the origin supported by both operating systems (Windows 10 and Windows Server 2019) for the application updates.

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

43.2 Test Activity

43.2.1 Test 1

The evaluator will ensure that the update has a digital signature which chains to the OS vendor or other trusted root managed by the OS. An application update shall be modified so its digital signature is no longer valid. Therefore the evaluator will ensure that the OS does no install the modified update.

43.2.1.1 Setup

Before the test execution, the following setup conditions must be fulfilled:

For Windows 10 and Windows Server 2019 operating systems:

- *SignTool*, a command line tool that provides the ability to sign files and verify signatures in files. It is distributed with the *Windows 10 Software Development Kit (SDK)*.
- A hexadecimal editor (e.g. *HxD*)

© 2019 Microsoft Corporation. All rights reserved.

- A file archiver (e.g. *WinRar*).
- *AppInstallApp_1.0.0.0* and *AppInstallApp_1.1.0.0 .appxbundle* (or *.appx*) installer packages provided by the vendor. These packages require the following dependency packages:
  - *Microsoft.NET.Native.Framework.1.3.appx*
  - *Microsoft.NET.Native.Runtime.1.4.appx*
  - *Microsoft.VCLibs.x64.14.00.appx*

43.2.1.2 Procedure

Before completing this test assurance activity, a description about the components implicated in an application update package is needed:

Internal files description:

As stated in the documentation review activity, an *AppX* has a certain package implementation, explained in the *Appendix B* of the ***Security Target*** document.

This is the relevant information about the *AppX* package implementation:

- The *.appx* package is a compressed file package conforming to the Open Packaging Conventions (*OPC*) standard, which is a ZIP compression format. The, *.appxbundle* package is a compressed file conforming *OPC* standard, which is a ZIP compression format that contains one or more *.appx* packages.
- Main files in *.appx* package: An *.appx* file is composed by the following security-relevant files:
  - *AppxBlockMap.xml* - Contains the SHA256 hashes of each 64k block of data in all of the files in the package. Note that if a single file spans more than 64k it is represented by multiple blocks maps.
  - *AppxSignature.p7x* - This file contains the digital signature that encompasses the security relevant information in the package.
- In addition to these files, the *.appx* package also contains others non security-relevant files related to the content application itself. Each component in the package can be identified in a binary editor searching the magic number *50 4B 03 04*.
- Cryptographic validation during the application installation: When processing an *.appxbundle* (or _.appx*) package the installer first verifies the* AppxBlockMap.xml *by computing a hash value and comparing the value with the signed hash value in the associated* AppxSignature.p7x *file. If the hash and signature validate passes, each block-hash is compared with the hash in the* AppxBlockMap.xml *file. As it is stated in the first bullet, the* .appxbundle *file may contain more than one* .appx *file, for example one* .appx *file for each supported platform (e.g.* x86, x64,...*) or for each supported language (e.g.* en-US, fr-FR, es-ES,...*). Components that are NOT installed are NOT validated.

Taking into account this information, the evaluator has performed the following test:

Digital signature modification:

- Create a copy of the application update installer file, which is going to be modified.

© 2019 Microsoft Corporation. All rights reserved.

- Open a *cmd* terminal and go to the folder where *SignTool* is installed (typical path is: **%programfiles(x86)%\WindowsKits\10\bin\10.0.17763.0\x64**).
- Obtain the digital signature typing the following command:

  ```
  signtool.exe verify /all /pa /v updatefile.appxbundle
  ```

  The signature will be shown in *Hash of file (sha256)* field.
- Open the application update installer file (*AppInstallApp_1.1.0.0.appxbundle*) with a compressor file tool and extract the *AppxSignature.p7x* file. This file contains the digital signature of the application update installer file.
- Open the obtained file in the hexadecimal editor and search the value obtained in the third step. Modify some bytes of the digital signature and save the file.
- Replace the original *AppxSignature.p7x* with the modified one using the file compressor tool (e.g. *WinRar*). Note that, the application installer has not been unpacked; therefore its internal structure is still valid.
- The digital signature can be checked using the *Signtool* utility and typing the following command:

  ```
  signtool verify /all /pa /v updatefile.appxbundle
  ```

  Ensure that digital signature is no longer valid after the modification.
- Finally, attempt to install the modified application installer file and observe that the operating system rejects the operation. To do that, open a *PowerShell* terminal as administrator and type the following command:

  ```
  .\Add-AppDevPackage.ps1
  ```

43.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The results obtained during the test execution are described as follows:

To check the signature verification during an application update, the evaluator has performed the following test. Two applications with different version created by the evaluator in *.appxbundle* format have been used. Each *.appxbundle* contains three *.appx* files, one for each architecture: ARM, 32-bits and 64-bits.

Before performing the test and due to these applications are not signed by a trusted certification authority, the evaluator has performed the following steps in order to allow their installation:

- Enable the developer mode. To do that, go to *Settings->Update & Security-> For developers* and select *Developer mode* option.

Digital signature modification:

- First, the evaluator has executed the *signtool* command:

  > signtool verify /all /pa /v AppInstallApp_1.1.0.0.appxbundle

  and has obtained the hash of the file (SHA-256) from the application installer file:

© 2019 Microsoft Corporation. All rights reserved.

```
C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x64>signtool.exe verify /all /pa /v "D:\Windows scripts\TestingWindowsO18U\FPT\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\App
InstallApp_1.0.0.0_1.1.1.0_x86_x64_arm_Debug.appxbundle"

Verifying: D:\Windows scripts\TestingWindowsO18U\FPT\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\AppInstallApp_1.0.0.0_1.1.1.0_x86_x64_arm_Debug.appxbundle

Signature Index: 0 (Primary Signature)
Hash of file (sha256): 415050584158504 33AF3275BF237A173FE8D724D36E8BC300724638CB1C03A317416BCBDCD83AAA64158434451 66C399D870EFC11716931E24DD69641B77EDDAD4FD04CBD86B488F5E70EEAD415843548AE94B77A2A3A7C0123A95DA7311
FFF3C947B0DF40E003FF5EA69492E7B8EEAAA4158424DF0FC49E94EFF5711948713511C672726799A4155A6EBADAD493BF8560E941624

Signing Certificate Chain:
    Issued to: Epoche
    Issued by: Epoche
    Expires:   Wed Nov 14 11:05:55 2040
    SHA1 hash: 036B2BFC283567E596731557B4C1265120F1C0C9

File is not timestamped.


Successfully verified: D:\Windows scripts\TestingWindowsO18U\FPT\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\AppInstallApp_1.0.0.0_1.1.1.0_x86_x64_arm_Debug.appxbundle

Number of signatures successfully Verified: 1
Number of warnings: 0
Number of errors: 0
```

- Second, the evaluator has extracted the *AppxSignature.p7x* from the *AppX* bundle and opened it in the hexadecimal editor. After that, the evaluator has searched the hash obtained above. This hash is at the offset *0x00000090* as it is shown in the following picture:

- Third, the hash is modified:

© 2019 Microsoft Corporation. All rights reserved.

- After the modification, the file is added to the *AppX* bundle, the evaluator has checked again the digital signature. Now, the obtained results states that the digital signature could not be verified:

- Finally, the evaluator has attempted to install the modified application installer file. The modification is detected during the installation so the application could not be installed. An error message is shown:

43.2.1.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has verified for both operating systems that the application updates are not installed if there is an integrity error in some file used during the update process.

Therefore, the evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to **Test 1**.

43.2.2 Test 2

The evaluator will ensure that the update has a digital signature which chains to the OS vendor or other trusted root managed by the OS. The evaluator will ensure that the OS successfully installs the update.

43.2.2.1 Setup

The applicable setup for this test is the same as the one defined for *Test 1*.

43.2.2.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

The evaluator shall attempt to install the original application installer file used in Test 1. To do that the evaluator shall carry out the following steps:

- Open a *PowerShell* terminal as administrator and type the following command:

  ```
  Add-AppxPackage AppInstallApp_1.0.0.0.appxbundle -DependencyPath \architecture
  ```

  Observe that the application shall be installed successfully.

After that, the previous application will be updated to the latest version. To do that the evaluator shall carry out the following steps:

- Open a *PowerShell* terminal as administrator and type the following command:

  ```
  Add-AppxPackage AppInstallApp_1.1.0.0.appxbundle -DependencyPath \architecture
  ```

  Observe that the application shall be updated successfully.

43.2.2.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The results obtained during the test execution are described as follows:

- The evaluator has attempted to install an application file which has a valid digital signature belonging to the evaluator. The installation was successful, as it can be observed in the following pictures:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

```
PS C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test> .\Add-AppDevPackage.ps1
Found bundle: C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\AppInstallApp_1.0.0.0_1.0
.0.0_x86_x64_arm_Debug.appxbundle
Found certificate: C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\AppInstallApp_1.0.0.
0_1.0.0.0_x86_x64_arm_Debug.cer

Before installing this app, you need to do the following:
        - Install the signing certificate
The certificate was successfully installed.

Installing app...
Found dependency package(s):
C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\Dependencies\x86\Microsoft.NET.CoreRunt
ime.2.1.appx
C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\Dependencies\x86\Microsoft.VCLibs.x86.D
ebug.14.00.appx
C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\Dependencies\x64\Microsoft.NET.CoreRunt
ime.2.1.appx
C:\Users\Administrator.WINDOWS-7OMIBE7\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.0.0.0_Debug_Test\Dependencies\x64\Microsoft.VCLibs.x64.D
ebug.14.00.appx

Success: Your app was successfully installed.
Press Enter to continue...:
```

© 2019 Microsoft Corporation. All rights reserved.

- After that, the evaluator has attempted to install an update of the application which has a valid digital signature belonging to the vendor. The update was successful, as it can be observed in the following pictures:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

```
PS C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test> .\Add-AppDevPackage.ps1
Found bundle: C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\AppInstallApp_1.0.
0.0_1.1.1.0_x86_x64_arm_Debug.appxbundle

Installing app...
Found dependency package(s):
C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\Dependencies\x86\Microsoft.NET.C
oreRuntime.2.1.appx
C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\Dependencies\x86\Microsoft.VCLib
s.x86.Debug.14.00.appx
C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\Dependencies\x64\Microsoft.NET.C
oreRuntime.2.1.appx
C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test\Dependencies\x64\Microsoft.VCLib
s.x64.Debug.14.00.appx

Success: Your app was successfully installed.
Press Enter to continue...:
PS C:\Users\secevaladmin\Desktop\FPT_TUD_EXT.2.2\FPT_TUD_EXT.2.2\AppInstallApp_1.0.0.0_1.1.1.0_Debug_Test>
```

© 2019 Microsoft Corporation. All rights reserved.

43.2.2.4 Verdict

The evaluator has verified for both operating systems that the application updates are installed properly if the update file has a digital signature which chains to the OS vendor.

Therefore, the evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to **Test 2**.

43.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TUD_EXT.2.2.

© 2019 Microsoft Corporation. All rights reserved.

44 FTA_TAB.1.1

The assurance activity for the **FTA_TAB.1.1** requirement is stated as follows:

> The evaluator will configure the OS, per instructions in the OS manual, to display the advisory warning message "TEST TEST Warning Message TEST TEST". The evaluator will then log out and confirm that the advisory message is displayed before logging in can occur.

44.1 Documentation Review activity

44.1.1 Findings

The evaluator has reviewed the ***Operational Guidance*** document, which explains the necessary steps to configure the OS to display the advisory warning message. This document includes in Section *6.11 Managing the logon banner* two links to the vendor support webpage.

These links had been checked and the evaluator has followed the described steps to configure the OS advisory message items:

- Message title for users attempting to log on.
- Message text for users attempting to log on.

The following screenshots show the information provided through the links included in the ***Operational Guidance*** document.

© 2019 Microsoft Corporation. All rights reserved.

Interactive logon: Message title for users attempting to log on

04/19/2017 • 3 minutes to read • Contributors

Applies to

- Windows 10

Describes the best practices, location, values, policy management and security considerations for the **Interactive logon: Message title for users attempting to log on** security policy setting.

Reference

This security setting allows you to specify a title that appears in the title bar of the window that contains the **Interactive logon: Message title for users attempting to log on**. This text is often used for legal reasons—for example, to warn users about the ramifications of misusing company information, or to warn them that their actions might be audited.

The **Interactive logon: Message title for users attempting to log on** and Interactive logon: Message text for users attempting to log on policy settings are closely related. **Interactive logon: Message title for users attempting to log on** specifies a message title to be displayed to users when they log on.

Not using this warning-message policy setting leaves your organization legally vulnerable to trespassers who unlawfully penetrate your network. Legal precedents have established that organizations that display warnings to users who connect to their servers over a network have a higher rate of successfully prosecuting trespassers.

When these policy settings are configured, users will see a dialog box before they can log on to the server console.

Possible values

- *User-defined title*
- Not defined

Best practices

1. It is advisable to set **Interactive logon: Message title for users attempting to log on** to a value similar to one the following:
    - RESTRICTED SYSTEM

© 2019 Microsoft Corporation. All rights reserved.

Interactive logon: Message text for users attempting to log on

04/19/2017 • 3 minutes to read • Contributors

Applies to

- Windows 10

Describes the best practices, location, values, management, and security considerations for the **Interactive logon: Message text for users attempting to log on** security policy setting.

Reference

The **Interactive logon: Message text for users attempting to log on** and Interactive logon: Message title for users attempting to log on policy settings are closely related. **Interactive logon: Message text for users attempting to log on** specifies a text message to be displayed to users when they log on. Interactive logon: Message title for users attempting to log on specifies a title to appear in the title bar of the window that contains the text message. This text is often used for legal reasons—for example, to warn users about the ramifications of misusing company information, or to warn them that their actions might be audited.

Not using this warning-message policy setting leaves your organization legally vulnerable to trespassers who unlawfully penetrate your network. Legal precedents have established that organizations that display warnings to users who connect to their servers over a network have a higher rate of successfully prosecuting trespassers.

When these policy settings are configured, users will see a dialog box before they can log on to the server console.

Possible values

The possible values for this setting are:

- User-defined text
- Not defined

Best practices

- It is advisable to set **Interactive logon: Message text for users attempting to log on** to a value similar to one of the following:

  1. IT IS AN OFFENSE TO CONTINUE WITHOUT PROPER AUTHORIZATION.

44.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

44.2 Test Activity

44.2.1 Test 1

During this test, an advisory warning message will be configured and will be shown by the OS before establishing a user session.

© 2019 Microsoft Corporation. All rights reserved.

44.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- The *PowerShell* execution policy shall be configured to allow the execution of *PowerShell* scripts.

44.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- Run the script FTA_TAB.1.ps1, to execute it, open a *PowerShell* terminal as administrator, then type:

  ```
  .\FTA_TAB.1.ps1
  ```

- The computer will be restarted. This script modifies the following *Windows Registry* keys related to the advisory warning message:

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
  \\CurrentVersion\Policies\System\legalnoticecaption

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
  \\CurrentVersion\Policies\System\legalnoticetext
  ```

- The source code of the script is the following:

  ```
  #
  # Test FTA_TAB.1
  #

  Write-Host "[I]: This script enables/disables the warning message on login
      screen. NOTE: Your
    computer will be restarted"
  $pathFolder = "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
  CurrentVersion\Policies\System"
  $keyName1 = "legalnoticecaption"
  $keyName2 = "legalnoticetext"
  $key1 = Get-ItemProperty -Path $pathFolder -Name $keyName1 -ErrorAction
      SilentlyContinue
  $key2 = Get-ItemProperty -Path $pathFolder -Name $keyName2 -ErrorAction
      SilentlyContinue

  If (($key1.legalnoticecaption -eq "") -and ($key2.legalnoticetext -eq "")) {
      Set-ItemProperty -Path
          Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
      CurrentVersion
        \Policies\System -Name legalnoticecaption -Value "WARNING MESSAGE"
      Set-ItemProperty -Path
          Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
      CurrentVersion
        \Policies\System -Name legalnoticetext -Value "TEST TEST Warning Message
            TEST TEST"
  ```

© 2019 Microsoft Corporation. All rights reserved.

```
    Write-Host "The computer will be restarted!"
    Pause
    Restart-Computer -Force
} else {
    Set-ItemProperty -Path
        Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
    CurrentVersion
      \Policies\System -Name legalnoticecaption -Value ""
    Set-ItemProperty -Path
        Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
    CurrentVersion
      \Policies\System -Name legalnoticetext -Value ""
    Write-Host "The computer will be restarted!"
    Pause
    Restart-Computer -Force
}
```

44.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has run the script above mentioned to configure the OS to display an advisory message.

On Windows, after rebooting the machine and before the login screen, the following message is shown:

© 2019 Microsoft Corporation. All rights reserved.

44.2.1.4 Verdict

The evaluator considers that, the results obtained during the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

44.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FTA_TAB.1.1.

© 2019 Microsoft Corporation. All rights reserved.

45 FTP_ITC_EXT.1.1 (DTLS)

The assurance activity for the **FTP_ITC_EXT.1.1 (DTLS)** requirement is stated as follows:

> The evaluator will configure the OS to communicate with another trusted IT product as identified in the second selection. The evaluator will monitor network traffic while the OS performs communication with each of the servers identified in the second selection. The evaluator will ensure that for each session a trusted channel was established in conformance with the protocols identified in the first selection.

45.1 Documentation Review activity

45.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

45.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

45.2 Test Activity

The assurance activity states that the OS shall use DTLS 1.0 and 1.2 to support web traffic and datagram-based application protocols. These capabilities are tested in FCS_DTLS_EXT.1.1 and FCS_DTLS_EXT.1.2, where the tools *WebClient.exe* and *WebServer.exe* were used, effectively verifying that a trusted channel was established for UDP traffic with web contents. The documentation for FCS_DTLS_EXT.1.1 includes information about the suitability of these tools for this evaluation.

45.2.1 Test 1

45.2.1.1 Setup

The applicable setup for this test is covered in the Test Activity section for the FCS_DTLS_EXT.1.1 requirement.

45.2.1.2 Procedure

The procedure for this test is covered in the Test Activity section for the FCS_DTLS_EXT.1.1 requirement.

© 2019 Microsoft Corporation. All rights reserved.

45.2.1.3 Results

From the FCS_DTLS_EXT.1.1 results, it was verified that a trusted channel was effectively established for UDP traffic with web contents by using DTLS 1.0 as well as DTLS 1.2.

45.2.1.4 Verdict

According to the obtained results for FCS_DTLS_EXT.1.1 and FCS_DTLS_EXT.1.2, the evaluator considers that the requirements established in the assurance activity section for FTP_ITC_EXT.1(DTLS) are fulfilled. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

45.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FTP_ITC_EXT.1.1(DTLS).

© 2019 Microsoft Corporation. All rights reserved.

46 FTP_ITC_EXT.1.1 (TLS)

The assurance activity for the **FTP_ITC_EXT.1.1 (TLS)** requirement is stated as follows:

> The evaluator will configure the OS to communicate with another trusted IT product as identified in the second selection. The evaluator will monitor network traffic while the OS performs communication with each of the servers identified in the second selection. The evaluator will ensure that for each session a trusted channel was established in conformance with the protocols identified in the first selection.

46.1 Documentation Review activity

46.1.1 Findings

Assurance activity does not state any documentation review activity for this requirement.

46.1.2 Verdict

Assurance activity does not state any documentation review activity for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

46.2 Test Activity

46.2.1 Test 1

46.2.1.1 Setup

The certificates used for this Assurance Activity are automatically generated by the Active Directory Certificate Services, and its configuration is explained below.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine acting as Active Directory Domain Controller, Root Certificate Authority and Web Server (Windows Server 2016)
- Client Machine (Platforms listed in the ST)

These machines are all in the same network with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.10.*

The client machine shall have the secure configuration enabled, according to the section **5.2 Operational prerequisites** of the ***Operational Guidance***. In addition, The *Wireshark* network analyzer is installed on the client machine.

© 2019 Microsoft Corporation. All rights reserved.

The Server Machine shall have the following roles enabled: Active Directory Domain Services, Active Directory Certificate Services and Web Server (IIS). In order to configure the Server Machine the next steps are followed.

"Active Directory Domain Services" role needs to be installed first. It can be done following the next steps:

- Open the Server Manager form the task bar.
- From the Server Manager Dashboard select "Manage" and then "Add Roles and Features".
- Click next until the "Select server roles" screen, then select "Active Directory Domain Services" and then click "Next", accepting the suggested additional features.
- Click "Next" until the confirmation screen is reached, review the selections made and click "Install".
- When finished, click on the "Promote this server to a domain controller" option, in blue colour together with the results.
- From the Deployment Configuration tab select "Add a new forest". Insert your root domain name into the corresponding field and click next. For this evaluation, "lab.ee" was introduced.

- Select Forest and Domain functional levels. In this case, both are set to "Windows Server 2016". Domain Name System (DNS) server is selected by default. Type a password in the DSRM password field and then click "Next".

© 2019 Microsoft Corporation. All rights reserved.

- The next screen warns about DNS delegation unavailability. Click “Next”.
- Confirm the NetBIOS name (“LAB” for this evaluation), and then click “Next”.

© 2019 Microsoft Corporation. All rights reserved.

- The location of SYSVOL, database folders and log files are selected by default. Click "Next".
- Click "Next" on the "Review Options" screen.
- The system will check to ensure all prerequisites are installed on the system for several minutes. If these tests pass, click "Install".

After the computer restarts, an LDAP user shall be created in the Active Directory Domain Services, following the next steps:

- Open the Server Manager from the task bar.
- Select "AD DS" form the left panel in the Server Manager Dashboard.
- Right click on the server and select "Active Directory Administrative Center".
- Select the section <Domain Name> (local) (Domain Name is set during the Active Directory installation).
- Double-click on the Users folder.
- On the right panel, click "New" in the "Users" section and then click "User".
- On the "Create User" screen, fill the fields: Full name "John Doe", user UPN logon "jdoe". Introduce a valid password and select "Other password options" to avoid the need to change it later and then click "OK".

© 2019 Microsoft Corporation. All rights reserved.

- Verify the new user appears in the list of users of the domain.

Next, Active Directory Certificate Services shall be installed and configured following the next steps:

© 2019 Microsoft Corporation. All rights reserved.

- Open the Server Manager form the task bar.
- From the Server Manager Dashboard select "Manage" and then "Add Roles and Features".
- Click next until the "Select server roles" screen, then select "Active Directory Certificate Services" and click "Next", accepting the suggested additional features.
- Click "Next" until the "Select role services" screen is shown, and then check the option "Certification Authority" and click "Next".
- Click "Install".
- When the installation is finished, click "Configure Active Directory Certificate Services on the destination server", in blue color together with the results.
- If not there already, write the name of the administrator ("LAB/Administrator" for this evaluation) in the "Credentials" field and click "Next".

- On the "Role services" screen check the option "Certification Authority" and click "Next".
- On the "Setup Type" screen, select "Enterprise CA" and click "Next".
- On the "CA Type" screen, select "Root CA" and click "Next".
- On the "Private Key" screen, select "Create a new private key" and click "next".
- On the "Cryptography for CA" screen, select SHA256 and click "Next".
- On the "CA Name" screen, all fields are left to default. Click "Next".

© 2019 Microsoft Corporation. All rights reserved.

- On the “Validity Period” screen, select 25 years and click “Next”.
- On the “CA Database” screen, the database locations are left to default. Click “Next”.
- Review the information on the “Confirmation” screen and click “Configure”.

© 2019 Microsoft Corporation. All rights reserved.

Once finished, reboot the Server Machine. This automatically creates a certificate with *serverAuth* key usage, with reference identifier set to <machine name>.<domain FQDN> (for this evaluation this was "LAB-EE.lab.ee") and issued by the root CA just created.

© 2019 Microsoft Corporation. All rights reserved.

The last role that needs to be installed is "Web Server (IIS)". In order to do so next steps shall be followed:

- Open the Server Manager form the task bar.
- From the Server Manager Dashboard select "Manage" and then "Add Roles and Features".
- Click next until the "Select server roles" screen.
- Select "Web Server (IIS)" and then click "Next", accepting the suggested additional features.
- Click "Next" until the "Select role services" screen is shown, and then leave the options selected by default, additionally checking "Client Certificate Mapping Authentication" under "Web Server" -> "Security". Then click "Next".
- Click "Install" and restart the machine when finished.

The IIS server shall be configured so that its contents can only be accessed authenticating with a valid client certificate belonging to a user present in the Active Directory, concretely the John Doe user just created. The next steps shall be followed to configure such a scenario:

© 2019 Microsoft Corporation. All rights reserved.

- Open the Server Manager form the task bar.
- From the Server Manager Dashboard select "Tools" and then "Internet Information Services (IIS) Manager".
- On the left panel, click on <Computer name> (<NetBIOS name>\<Username>), which corresponds to "LAB-EE (LAB\Administrator)" for this evaluation.
- In the middle "Features View" panel, double-click on "Authentication". Disable "Anonymous Authentication" and enable "Active Directory Client Certificate Authentication".

- On the left panel, click on "Default Web Site".
- On the right "Actions" panel, click "Bindings...".
- Click "Add..." and select "https" as "Type", and the <machine name>.<domain FQDN> certificate ("LAB-EE.lab.ee" in this evaluation) as "SSL certificate", leaving the other options by default. Then click "OK" and "Close".

© 2019 Microsoft Corporation. All rights reserved.

- On the middle "Features View" panel, double-click "SSL Settings".
- Check "Require SSL" and select "Require" client certificates. Then click "Apply" on the right panel.

The Client Machine needs to belong to the domain just created so that it can get a client certificate for authentication. In order to do so, follow the next steps:

- Ensure the DNS of the network adapter are set to the Server Machine's IP.
- Right click on the "Windows" icon and click "System".
- On the displayed "About" screen, click on "Connect to work or school".
- On the "Microsoft Account" popup screen, select "Join this device to a local Active Directory domain" under "Alternate actions".

© 2019 Microsoft Corporation. All rights reserved.

- On the "Join a domain" popup screen, introduce the FQDN of the domain that was created ("lab.ee" for this evaluation) and click next.

© 2019 Microsoft Corporation. All rights reserved.

- Introduce the Administrator's username and password.
- In the "Add an account" popup screen, click "Skip".
- A system restart is required after the successful domain join.

Once restarted, login as the domain user created previously ("LAB\jdoe" for this evaluation). The next steps shall be followed to request the client certificate from the certificate authority:

- Open "mmc.exe" (e.g. from WIN+R).
- Click on "File" -> "Add/Remove Snap-in…".
- Select "Certificates" on the left panel and click "Add" and then "OK".
- On the left panel, under "Certificates" right click on "Personal", select "All Tasks" and then "Request new Certificate…".
- Click "Next".
- On the "Select Certificate Enrollment Policy" screen, select "Active Directory Enrollment Policy" and click "Next".

© 2019 Microsoft Corporation. All rights reserved.

- On the "Request Certificates" screen, check "User" and click "Enroll".

- Once finished, the presence of the client certificate can be verified in "mmc.exe" if the "Certificates" section under "Personal" is refreshed.

© 2019 Microsoft Corporation. All rights reserved.

46.2.1.2 Procedure

The following steps shall be performed in order to complete this test assurance activity:

- On the client machine, use a web browser (Microsoft Edge or Internet Explorer) to navigate to https://<machine name>.<domain FQDN> (for this evaluation, "https://lab-ee.lab.ee".
- A popup prompting the user to select a client certificate for authentication should be shown. Select the user's certificate.
- Verify on the browser that the connection is successfully performed and a trusted communication channel is established using TLS.
- On the Server Machine, open the Server Manager from the task bar.
- Select "AD CS" from the left panel in the Server Manager Dashboard.
- Right click on the server and select "Certification Authority".
- Select "Issued Certificates" under the CA name on the left panel.
- Right click on the client certificate (requester name "LAB\jdoe" in this evaluation) and select "Revoke Certificate" under "All Tasks".

- Right click on "Revoked Certificates" on the left panel and then select "Publish" under "All Tasks".
- On the "Publish CRL" window, select "New CRL" and click "OK".

© 2019 Microsoft Corporation. All rights reserved.

- On the client machine, use a web browser once again (Microsoft Edge or Internet Explorer) to navigate to https://<machine name>.<domain FQDN> (for this evaluation, "https://lab-ee.lab.ee".
- A popup prompting the user to select a client certificate for authentication should be shown. Select the user's certificate.
- Verify on the browser that user is presented a 403 page informing that the access is denied due to the user not being successfully authenticated.

46.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows 10 Home x86 and Windows 10 Home x64, these tests have not been performed because they do not have the functionality to be domain joined.

When the user connects to the server using a browser like Microsoft Edge, it is asked to select the client certificate as shown in the following picture:

© 2019 Microsoft Corporation. All rights reserved.

And using Internet Explorer:

In the first case, when the certificate is valid, the contents server by the web server can be accessed since the user is successfully authenticated, as can be verified in the browser as follows:

© 2019 Microsoft Corporation. All rights reserved.

The network capture shows that the connection is effectively using TLS 1.2:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.005523	10.10.20.25	10.10.20.40	TLSv1.2	236	Client Hello
6	0.006747	10.10.20.40	10.10.20.25	TLSv1.2	1683	Server Hello, Certificate, Server Hello Done
21	4.820046	10.10.20.25	10.10.20.40	TLSv1.2	236	Client Hello
22	4.820475	10.10.20.40	10.10.20.25	TLSv1.2	1683	Server Hello, Certificate, Server Hello Done
25	4.825977	10.10.20.25	10.10.20.40	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
26	4.826328	10.10.20.25	10.10.20.40	TLSv1.2	155	Application Data
28	4.826516	10.10.20.25	10.10.20.40	TLSv1.2	363	Application Data
29	4.827876	10.10.20.40	10.10.20.25	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
30	4.828031	10.10.20.40	10.10.20.25	TLSv1.2	139	Application Data
33	4.828780	10.10.20.40	10.10.20.25	TLSv1.2	107	Application Data
34	4.829406	10.10.20.25	10.10.20.40	TLSv1.2	107	Application Data
36	4.876348	10.10.20.40	10.10.20.25	TLSv1.2	123	Application Data

© 2019 Microsoft Corporation. All rights reserved.

Additionally, the CRL checking capability of the authentication server is checked when trying to authenticate when the client certificate is already revoked by the CA. The contents of the web server are not shown due to the invalid credentials, as it can be observed in the following picture:

With the previous described steps, it has been possible to verify the establishment of a trusted communications channel with an IT entity with all the requirements stated for this assurance activity: it serves a web page (*web traffic* capability) which requires a valid client certificate to be presented (verified via its *authentication server*) which is in turn checked for revocation (via *CRL checking)*.

46.2.1.4 Verdict

According to the results presented in the previous section, the evaluator considers that the test results obtained during the **Test 1** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** activity.

46.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to this FTP_ITC_EXT.1.1(TLS).

© 2019 Microsoft Corporation. All rights reserved.

47 FTP_TRP.1.1

The assurance activity for the **FTP_TRP.1.1** requirement is stated as follows:

> The evaluator will examine the TSS to determine that the methods of remote OS administration are indicated, along with how those communications are protected. The evaluator will also confirm that all protocols listed in the TSS in support of OS administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator will confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator will also perform the following tests:
>
> - **Test 1:** The evaluator will ensure that communications using each remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
> - **Test 2:** For each method of remote administration supported, the evaluator will follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
> - **Test 3:** The evaluator will ensure, for each method of remote administration, the channel data is not sent in plaintext.
> - **Test 4:** The evaluator will ensure, for each method of remote administration, modification of the channel data is detected by the OS.

47.1 Documentation Review activity

47.1.1 Findings

Section **6.8 Trusted Channels** of the ***Security Target*** document describes the protections used for the *TLS* and *HTTPS* communications. Moreover, there is a pointer to the section **6.2.3 Networking (TLS)** where the description of how the communication are protected is located.

This section also describes the offered remote access methods, which are the following:

- Remote Desktop Services.
- Connect to another computer using Remote Desktop Connection.

Finally, the Operational Guidance ***Operational Guidance*** document, includes in section **6.17 Managing Remote Administration** the necessary steps to establish a *Remote Desktop Connection*.

© 2019 Microsoft Corporation. All rights reserved.

47.1.2 Verdict

The evaluator has found in the TSS the methods used for the remote access and the protocols used to protect the connection.

In addition, the evaluator has checked the ***Operational Guidance*** document, includes information for establishing the remote administrative sessions for each supported method listed in the ***Security Target*** document.

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

47.2 Test Activity

47.2.1 Test 1 & Test 3

The evaluator will ensure that communications using Remote Administration methods are successful. In addition, for each method of remote administration, the data is not sent in plain text.

47.2.1.1 Setup

On all the platforms, the remote administration method is the Remote Desktop Connection. The necessary scenario to perform the tests over this method is described in the assurance activity section and it is composed by the following elements:

- Server Machine (Windows Server 2019)
- Client Machine (Platforms listed in the ST)

These machines are in the same network.

47.2.1.1.1 Remote Desktop Administration

To be able to use the Remote Administration, the following steps shall be performed in the Server Machine.

- Go to Server Manager application, click *Manage* then, *Add Roles and Features*.
- Click on *Next*.
- Select *Role-based or feature-based installation* and click on *Next*.
- Mark the checkbox with the option: *Select a server from the server pool* and choose the server with the IP of the Server Machine and click on *Next*.
- Select *Remote Desktop Services*. Press *Add features* and click on *Next*.
- Click on *Next*.
- Press *Install*.

© 2019 Microsoft Corporation. All rights reserved.

To record the establishment of the remote administration connection, the Server Machine shall have a network packet analyzer application installed (e.g. *Wireshark*).

The Client Machine shall have enabled the secure configuration according to the section 5 of the ***Operational Guidance***. In addition, a network packet analyzer application shall be also installed (e.g. *Wireshark*).

47.2.1.2 Procedure

To complete this test assurance activity, the following steps shall be performed:

- In the Server and Client Machines, open a network packet analyzer application (e.g. *Wireshark*) to verify that the remote connection is established correctly.
- Open the "*Remote Desktop Connection*" application. (*WIN+r*, then type *mstsc.exe*)
- In the computer name field, type the server name (e.g. *HPELITEX2*) and in username field type the user name of the server (e.g. *evaluador*)

© 2019 Microsoft Corporation. All rights reserved.

- Click *Connect*.
- Type the password in the new prompt.

© 2019 Microsoft Corporation. All rights reserved.

- Accept the certificate sent by the server by clicking on *Yes*

© 2019 Microsoft Corporation. All rights reserved.

After that, a connection using *Remote Desktop Web Access* shall be performed to connect a remote user to the server machine. The following steps shall be performed:

- In the client machine, open Internet Explorer web browser and access the RD Web Access URL (e.g. *https://10.10.10.10/RDWeb*) once the webpage is loaded, click on *Continue to this website (not recommended)*. When prompted, click '*Allow*'. to enable the Microsoft Remote Desktop Services Web Access add-on.

© 2019 Microsoft Corporation. All rights reserved.

- In the RD Web Access page, type the Domain\Username (e.g. TEST\Administrator) and Password of the Server Machine. Then, click the *Sign in* button.

© 2019 Microsoft Corporation. All rights reserved.

- After that, click on the *Connect to a remote PC* tab.

- In the connection options page, type the IP of the client machine. After that, click on *Connect* button.

- Now, click on *Connect* button.

© 2019 Microsoft Corporation. All rights reserved.

- A prompt will be shown asking the Login credentials of the machine we are connecting to.

- Click on *Yes* to accept the certificate.

© 2019 Microsoft Corporation. All rights reserved.

- Finally, a connection is performed via the native application *Remote Desktop Connection* of the client machine.

© 2019 Microsoft Corporation. All rights reserved.

47.2.1.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has established successfully a remote connection by following the settings of the ***Operational Guidance*** document as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

To ensure that the channel data of this connection is not sent as plain text, the packets captured by *Wireshark* have been analyzed as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
30	9.765494	10.10.10.109	10.10.10.10	TLSv1.2	208	Client Hello
31	9.769775	10.10.10.10	10.10.10.109	TLSv1.2	1263	Server Hello, Certificate, Server Key Exchange, Server Hello Done
32	9.775539	10.10.10.109	10.10.10.10	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
33	9.781842	10.10.10.10	10.10.10.109	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
34	9.782551	10.10.10.109	10.10.10.10	TLSv1.2	140	Application Data
35	9.784488	10.10.10.10	10.10.10.109	TLSv1.2	272	Application Data
36	9.785638	10.10.10.109	10.10.10.10	TLSv1.2	666	Application Data
37	9.790224	10.10.10.10	10.10.10.109	TLSv1.2	142	Application Data
38	9.790580	10.10.10.109	10.10.10.10	TLSv1.2	179	Application Data
39	9.792727	10.10.10.10	10.10.10.109	TLSv1.2	87	Application Data
40	9.792945	10.10.10.109	10.10.10.10	TLSv1.2	545	Application Data

```
> Transmission Control Protocol, Src Port: 49673, Dst Port: 3389, Seq: 48, Ack: 20, Len: 154
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 149
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 145
        Version: TLS 1.2 (0x0303)
      > Random: 5c0ae7afc8c9279f44c1e3c57725b7f1e9499654829b4076...
        Session ID Length: 0
        Cipher Suites Length: 26
      v Cipher Suites (13 suites)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
          Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
        Compression Methods Length: 1
```

Once the connection is established, the user data are sent in the *Application Data* messages, and these are encrypted using TLS 1.2. In addition, the cipher suites selected by the client are the ones configured for the evaluation configuration. Therefore, the remote connection is established using a secure channel.

Finally, when the remote administration is carried out through the *RD Web Access*, the connection with the remote computer is made via the Remote Desktop native application of the client machine. Since this method is evaluated when a local user performs a remote desktop connection, the results are the same for both cases.

47.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrates the fulfilment of the **Test 1** and **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1** and **Test 3**.

47.2.2 Test 2

The evaluator will ensure that following the ***Operational Guidance*** document, there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.

© 2019 Microsoft Corporation. All rights reserved.

47.2.2.1 Setup

The machines used to perform this test shall have enabled the secure configuration according to the section **5 Evaluated configuration** of the ***Operational Guidance*** document.

47.2.2.2 Procedure

To perform this test, the necessary steps listed in the section **6.17 Managing Remote Administration** of the ***Operational Guidance*** document shall be followed to configure a remote connection.

47.2.2.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has configured the machines according to the ***Operational Guidance*** document before establishing the connection. Once the connection is established, it is secured using *TLS1.2* and *X509* certificate as it can be observed in the following image:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
30	9.765494	10.10.10.109	10.10.10.10	TLSv1.2	208	Client Hello
31	9.769775	10.10.10.10	10.10.10.109	TLSv1.2	1263	Server Hello, Certificate, Server Key Exchange, Server Hello Done
32	9.775539	10.10.10.109	10.10.10.10	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
33	9.781842	10.10.10.10	10.10.10.109	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
34	9.782551	10.10.10.109	10.10.10.10	TLSv1.2	140	Application Data
35	9.784488	10.10.10.10	10.10.10.109	TLSv1.2	272	Application Data
36	9.785638	10.10.10.109	10.10.10.10	TLSv1.2	666	Application Data
37	9.790224	10.10.10.10	10.10.10.109	TLSv1.2	142	Application Data
38	9.790580	10.10.10.109	10.10.10.10	TLSv1.2	179	Application Data
39	9.792727	10.10.10.10	10.10.10.109	TLSv1.2	87	Application Data
40	9.792945	10.10.10.109	10.10.10.10	TLSv1.2	545	Application Data

```
Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  > Random: 5c0ae7f7a7fbcdd4f4fad9079a189ae73298d7edd2b8f329...
    Session ID Length: 32
    Session ID: e125000075afe3a4e22fe49294ea4f40ff34b67d431ed731...
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
    Compression Method: null (0)
    Extensions Length: 9
  > Extension: extended_master_secret (len=0)
  > Extension: renegotiation_info (len=1)
Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 746
    Certificates Length: 743
  Certificates (743 bytes)
      Certificate Length: 740
    Certificate: 308202e0308201c8a0030201020210340ddf7682826b864a... (id-at-commonName=WINDCAC.EE.LAB)
      signedCertificate
          version: v3 (2)
          serialNumber: 0x340ddf7682826b864a14ba76e3b5d778
        > signature (sha256WithRSAEncryption)
        issuer: rdnSequence (0)
          > rdnSequence: 1 item (id-at-commonName=WINDCAC.EE.LAB)
        > validity
        > subject: rdnSequence (0)
        > subjectPublicKeyInfo
```

47.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

47.2.3 Test 4

The evaluator will ensure that any modification done to the channel data is detected by the OS and the connection is refused.

47.2.3.1 Setup

The Remote Administration method available is the *Remote Desktop Connection*. The necessary scenario to perform the tests described in the assurance activities of the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2019)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux 2018)

  These machines are in the same network.

© 2019 Microsoft Corporation. All rights reserved.

To be able to use the *Remote Administration*, the following steps shall be performed in the Server Machine.

- Go to Server Manager application, click *Manage* then, *Add Roles and Features*
- Click on *Next*.
- Select *Role-based or feature-based installation* and click on *Next*.
- Mark the checkbox with the option: *Select a server from the server pool* and choose the server with the IP of the Server Machine and click on *Next*.
- Select *Remote Desktop Services*. Press *Add features* and click on *Next*.
- Click on *Next*.
- Press *Install*.

To record the establishment of the Remote Administration connection, the Server Machine shall have a network packet analyzer application installed (e.g. *Wireshark*).

To perform this test, a MITM Machine shall be used. This machine shall be between the Server Machine and the Client Machine, acting as a switch and modifying the packets that match with certain patterns (depending the remote administration method used).

- Remote Desktop Connection:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"tcp port 3389",
                "regex":".*\\x17\\x03\\x03.*"
            },
    "replacement":{
        "regex":"(\\x17\\x03\\x03.{2})(.{1})",
        "replacement":"$1\\x41"
        }
    }
    ]
}
```

The Client Machine shall have enabled the secure configuration according to the section 5 of the ***Operational Guidance*** document. In addition, a network packet analyzer application shall be installed (e.g. *Wireshark*).

47.2.3.2 Procedure

To invalidate the channel data, two different approaches shall be followed:

- **Approach 1:** Modifying the packet before establishing the connection.
- **Approach 2:** Modifying the packet after the connection is established.

To complete this test assurance activity, the following steps shall be performed:

© 2019 Microsoft Corporation. All rights reserved.

- In the Server and Client Machines, open a network packet analyzer application (e.g. *Wireshark*) to verify that the remote connection is established correctly.
- Start the MITM tool that will modify all the traffic that matches the first pattern showed above.
- Open the *Remote Desktop Connection* application. (*WIN+r*, then type *mstsc.exe*).
- In the computer name, type the server name (e.g. *HPELITEX2*) and in username type the user name of the server(e.g. *EE\administrator*).

- Click *Connect*.
- Type the password in the new prompt.

© 2019 Microsoft Corporation. All rights reserved.

- Accept the certificate sent by the server by clicking on *Yes*

© 2019 Microsoft Corporation. All rights reserved.

- In the Client and Server machine, the *Wireshark* application shall be open and capturing the traffic to verify that the connection is not performed correctly.
- Repeat the previous process, but this time the modification will be performed once the connection is established.

47.2.3.3 Results

The evaluator has performed these tests on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2016 with Windows 10 Home x86 October 2018 Update (v1809).
- Microsoft Windows Hyper-V Server 2019 with Windows Server 2019 Standard (Desktop Experience) (v1809).
- Dell PowerEdge R740 with Windows Server 2019 Datacenter (Desktop Experience) (v1809).

The evaluator has obtained the same results in all tested platforms.

The evaluator has performed two different approaches and the obtained results are described as follows:

For the first approach the packet is modified before the connection is established when the server sends the first *Application Data* message. This packet is modified by the MITM as it can be observed in the following images:

- The packet before the modification:

No.	Time	Source	Destination	Protocol	Length	Info
18	3.881113900	20.20.20.1	20.20.20.10	TLSv1.2	232	Client Hello
19	3.882970100	20.20.20.10	20.20.20.1	TLSv1.2	890	Server Hello, Certificate, Server Hello Done
20	3.882980700	20.20.20.10	10.10.10.10	TLSv1.2	890	Server Hello, Certificate, Server Hello Done
21	3.890211000	10.10.10.10	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
22	3.890583400	20.20.20.1	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
23	3.894915100	20.20.20.10	20.20.20.1	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
24	3.894929400	20.20.20.10	10.10.10.10	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
43	12.990640300	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
44	12.991655600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
45	12.996979000	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
46	12.997016300	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert

```
> Frame 43: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.10.10.10, Dst: 20.20.20.10
> Transmission Control Protocol, Src Port: 53427, Dst Port: 3389, Seq: 566, Ack: 929, Len: 101
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 96
      Encrypted Application Data: 9dd92096d559c96986d17e0222f9f458c1e16583aedabcd2...
```

The MITM tool detects this packet and modifies its content.

- The packet after the modification:

No.	Time	Source	Destination	Protocol	Length	Info
18	3.881113900	20.20.20.1	20.20.20.10	TLSv1.2	232	Client Hello
19	3.882970100	20.20.20.10	20.20.20.1	TLSv1.2	890	Server Hello, Certificate, Server Hello Done
20	3.882980700	20.20.20.10	10.10.10.10	TLSv1.2	890	Server Hello, Certificate, Server Hello Done
21	3.890211000	10.10.10.10	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
22	3.890583400	20.20.20.1	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
23	3.894915100	20.20.20.10	20.20.20.1	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
24	3.894929400	20.20.20.10	10.10.10.10	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
43	12.990640300	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
44	12.991655600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
45	12.996979000	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
46	12.997016300	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert

```
> Frame 44: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 20.20.20.1, Dst: 20.20.20.10
> Transmission Control Protocol, Src Port: 53427, Dst Port: 3389, Seq: 566, Ack: 929, Len: 101
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 96
      Encrypted Application Data: eeee2096d559c96986d17e0222f9f458c1e16583aedabcd2...
```

© 2019 Microsoft Corporation. All rights reserved.

- Once the modification is performed, the Client Machine sends a *RST, ACK* packet, closing the connection established:

No.	Time	Source	Destination	Protocol	Length	Info
21	3.890211000	10.10.10.10	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
22	3.890583400	20.20.20.1	20.20.20.10	TLSv1.2	398	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
23	3.894915100	20.20.20.10	20.20.20.1	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
24	3.894929400	20.20.20.10	10.10.10.10	TLSv1.2	131	Change Cipher Spec, Encrypted Handshake Message
25	3.915428100	10.10.10.10	20.20.20.10	TCP	56	53427 → 3389 [ACK] Seq=566 Ack=929 Win=2101248 Len=0
26	3.915693100	20.20.20.1	20.20.20.10	TCP	56	53427 → 3389 [ACK] Seq=566 Ack=929 Win=2101248 Len=0
43	12.990640300	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
44	12.991655600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
45	12.996979000	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
46	12.997016300	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert
47	12.998013100	10.10.10.10	20.20.20.10	TCP	56	53427 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
48	12.999829800	20.20.20.1	20.20.20.10	TCP	56	53427 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0

- In addition, the *Remote Desktop* application shows the following error message:

For the second approach, the packet is modified once the connection is established.

- When the server sends an *Application Data* message, the MITM tool starts to work and the modification is performed. This packet is modified by the MITM tool as it can be observed in the following images:
- The packet before the modification:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
3211	52.733389700	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
3212	52.733620600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
3213	52.735406500	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
3214	52.735412900	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert
3215	52.739532500	10.10.10.10	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3216	52.739718800	20.20.20.1	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3217	52.746249400	10.10.10.10	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3218	52.746474100	20.20.20.1	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3219	52.747269200	20.20.20.10	20.20.20.1	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3220	52.747366000	20.20.20.10	10.10.10.10	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3221	52.747780400	10.10.10.10	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
3222	52.747952600	20.20.20.1	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0

```
> Frame 3211: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.10.10.10, Dst: 20.20.20.10
> Transmission Control Protocol, Src Port: 53434, Dst Port: 3389, Seq: 566, Ack: 929, Len: 101
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 96
      Encrypted Application Data: 97238efb6d4edf2d94b133cb40546a5f6dd3e8a629b2477c...
```

The MITM detects this packet and modifies it.

- The packet after the modification:

No.	Time	Source	Destination	Protocol	Length	Info
3211	52.733389700	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
3212	52.733620600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
3213	52.735406500	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
3214	52.735412900	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert
3215	52.739532500	10.10.10.10	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3216	52.739718800	20.20.20.1	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3217	52.746249400	10.10.10.10	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3218	52.746474100	20.20.20.1	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3219	52.747269200	20.20.20.10	20.20.20.1	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3220	52.747366000	20.20.20.10	10.10.10.10	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3221	52.747780400	10.10.10.10	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
3222	52.747952600	20.20.20.1	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0

```
> Frame 3212: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 20.20.20.1, Dst: 20.20.20.10
> Transmission Control Protocol, Src Port: 53434, Dst Port: 3389, Seq: 566, Ack: 929, Len: 101
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Application Data Protocol: Application Data
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 96
      Encrypted Application Data: eeee8efb6d4edf2d94b133cb40546a5f6dd3e8a629b2477c...
```

- Once the modification is performed, the Client Machine sends a *RST, ACK* packet, closing the connection established:

No.	Time	Source	Destination	Protocol	Length	Info
3211	52.733389700	10.10.10.10	20.20.20.10	TLSv1.2	157	Application Data
3212	52.733620600	20.20.20.1	20.20.20.10	TLSv1.2	157	Application Data
3213	52.735406500	20.20.20.10	20.20.20.1	TLSv1.2	109	Encrypted Alert
3214	52.735412900	20.20.20.10	10.10.10.10	TLSv1.2	109	Encrypted Alert
3215	52.739532500	10.10.10.10	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3216	52.739718800	20.20.20.1	20.20.20.10	TCP	56	53434 → 3389 [RST, ACK] Seq=667 Ack=982 Win=0 Len=0
3217	52.746249400	10.10.10.10	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3218	52.746474100	20.20.20.1	20.20.20.10	TCP	68	53435 → 3389 [SYN, ECN, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3219	52.747269200	20.20.20.10	20.20.20.1	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3220	52.747366000	20.20.20.10	10.10.10.10	TCP	68	3389 → 53435 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1
3221	52.747780400	10.10.10.10	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
3222	52.747952600	20.20.20.1	20.20.20.10	TCP	56	53435 → 3389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0

- In addition, the *Remote Desktop* application shows the following error message:

© 2019 Microsoft Corporation. All rights reserved.

47.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4**.

47.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TRP.1.1.

© 2019 Microsoft Corporation. All rights reserved.

48 FAU_GEN.1/WLAN

The assurance activity for the **FAU_GEN.1/WLAN** requirement is stated as follows:

> The evaluator shall check the operational guidance and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the EP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 2.
>
> The evaluator shall in particular ensure that the operational guidance is clear in relation to the contents for failed cryptographic events. In Table 2, information detailing the cryptographic mode of operation and a name or identifier for the object being encrypted is required. The evaluator shall ensure that name or identifier is sufficient to allow an administrator reviewing the audit log to determine the context of the cryptographic operation (for example, performed during a key negotiation exchange, performed when encrypting data for transit) as well as the non-TOE endpoint of the connection for cryptographic failures relating to communications with other IT systems.
>
> The evaluator shall also make a determination of the administrative actions that are relevant in the context of this EP. The TOE may contain functionality that is not evaluated in the context of this EP because the functionality is not specified in an SFR. This functionality may have administrative aspects that are described in the operational guidance. Since such administrative actions will not be performed in an evaluated configuration of the TOE, the evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the EP, which thus form the set of "all administrative actions". The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.
>
> - **Test 1:** The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records in accordance with the assurance activities associated with the functional requirements in this EP. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.
>
> Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

© 2019 Microsoft Corporation. All rights reserved.

48.1 Documentation Review activity

48.1.1 Findings

The ***Security Target*** document, defines in its section **5.1.1.1 Audit Data Generation(FAU_GEN.1) and FAU_GEN.1(WLAN)**, the following auditable events:

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/WLAN	None.	
FCS_CKM.1/WLAN	None.	
FCS_CKM.2/WLAN	None.	
FCS_TLSC_EXT.1/WLAN	Failure to establish an EAP-TLS session. Establishment/termination of an EAP-TLS session.	Reason for failure. Non-TOE endpoint of connection.
FIA_PAE_EXT.1	None.	
FIA_X509_EXT.2/WLAN	None.	
FIA_X509_EXT.4/WLAN	Attempts to load certificates. Attempts to revoke certificates.	None.
FMT_SMF_EXT.1/WLAN	None.	
FPT_TST_EXT.1/WLAN	Execution of this set of TSF self-tests. [***detected integrity violation***].	[***The TSF binary file that caused the integrity violation***].
FTA_WSE_EXT.1	All attempts to connect to access points.	Identity of access point being connected to as well as success and failures (including reason for failure).
FTP_ITC_EXT.1/WLAN	All attempts to establish a trusted channel.	Identification of the non-TOE endpoint of the channel.

Table 11 WLAN EP Audit Events

Moreover, the evaluator has reviewed the section **7.2 Audit events - WLAN client extended package** of the ***Operational Guidance***, where it is included a table with all the events related to the WLAN client extended package requirements implemented by the TOE. The content of this table matches with the selection performed by the vendor in the ***Security Target*** document as it can be seen in the image above.

© 2019 Microsoft Corporation. All rights reserved.

48.1.2 Verdict

The evaluator has reviewed the ***Security Target*** document and has ensured that every auditable event type selected in the ***Security Target*** document is included in the ***Operational Guidance*** document.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

48.2 Test Activity

48.2.1 Test 1

For this case, all the auditable events will be obtained during the test execution of each security functional requirement.

48.2.1.1 Setup

The necessary setup for obtaining each auditable event will be described in the *Setup* section of the requirements shown in the above table.

48.2.1.2 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms. All the audit data generation results are obtained and described in each test defined in the table above. In addition, the audit events will be shown below for the sake of a better readability.

FCS_TLSC_EXT.1/WLAN

- Failure to establish an EAP-TLS session (36888, 11, 30)

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A fatal alert was generated and sent to the remote endpoint. This may result in termination of
                       the connection. The TLS protocol defined fatal alert code is 48.

                          Target name:

                       The TLS alert registry can be found at
                       http://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-6
Id                   : 36888
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 0
Opcode               : 0
Keywords             : -9223372036854775808
RecordId             : 3339
ProviderName         : Schannel
ProviderId           : 1f678132-5938-4686-9fdc-c8ff68f15c85
LogName              : System
ProcessId            : 648
ThreadId             : 2228
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-21-3509454584-2400802258-3823911923-1103
TimeCreated          : 11/26/2018 3:19:10 PM
ActivityId           : 7f7880ea-857d-0001-1b81-787f7d85d401
RelatedActivityId    :
ContainerLog         : System
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty}
```

```
Message              : For more details for this event, please refer to the "Details" section
Id                   : 11
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 11
Opcode               : 2
Keywords             : 4611686018427387907
RecordId             : 19563
ProviderName         : Microsoft-Windows-CAPI2
ProviderId           : 5bbca4a8-b209-48dc-a8c7-b23d3e5216fb
LogName              : Microsoft-Windows-CAPI2/Operational
ProcessId            : 648
ThreadId             : 2228
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-21-3509454584-2400802258-3823911923-1103
TimeCreated          : 11/26/2018 3:19:10 PM
ActivityId           : 7f7880ea-857d-0001-1b81-787f7d85d401
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-CAPI2/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Stop
TaskDisplayName      : Build Chain
KeywordsDisplayNames : {Path Validation, Path Discovery}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message               : For more details for this event, please refer to the "Details" section
Id                    : 30
Version               : 0
Qualifiers            :
Level                 : 2
Task                  : 30
Opcode                : 0
Keywords              : 4611686018427387905
RecordId              : 19564
ProviderName          : Microsoft-Windows-CAPI2
ProviderId            : 5bbca4a8-b209-48dc-a8c7-b23d3e5216fb
LogName               : Microsoft-Windows-CAPI2/Operational
ProcessId             : 648
ThreadId              : 2228
MachineName           : DESKTOP-JLDC4RV.lab.ee
UserId                : S-1-5-21-3509454584-2400802258-3823911923-1103
TimeCreated           : 11/26/2018 3:19:10 PM
ActivityId            : 7f7880ea-857d-0001-1b81-787f7d85d401
RelatedActivityId     :
ContainerLog          : Microsoft-Windows-CAPI2/Operational
MatchedQueryIds       : {}
Bookmark              : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName      : Error
OpcodeDisplayName     : Info
TaskDisplayName       : Verify Chain Policy
KeywordsDisplayNames  : {Path Validation}
Properties            : {System.Diagnostics.Eventing.Reader.EventProperty}
```

- Establishment/termination of an EAP-TLS session (36880, 1793)

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A TLS client handshake completed successfully. The negotiated cryptographic parameters are as
                       follows.

                          Protocol version: TLS 1.2
                          CipherSuite: 0x2F
                          Exchange strength: 2048 bits
                          Context handle: 0x1f04b58d050
                          Target name:
                          Local certificate subject name:
                          Remote certificate subject name:
Id                   : 36880
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 0
Opcode               : 0
Keywords             : -9223372036854775808
RecordId             : 3304
ProviderName         : Schannel
ProviderId           : 1f678132-5938-4686-9fdc-c8ff68f15c85
LogName              : System
ProcessId            : 648
ThreadId             : 816
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 2:59:38 PM
ActivityId           : 7f7880ea-857d-0001-1b81-787f7d85d401
RelatedActivityId    :
ContainerLog         : System
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : A TLS Security Context handle is being deleted.

                           Context handle: 0x26ceb2ae4c0
                           Target name:
Id                   : 1793
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 28672
Opcode               : 1
Keywords             : -9223372036854775808
RecordId             : 1079
ProviderName         : Microsoft-Windows-Schannel-Events
ProviderId           : 91cc1150-71aa-47e2-ae18-c96e61736b6f
LogName              : Microsoft-Windows-Schannel-Events/Perf
ProcessId            : 636
ThreadId             : 672
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/27/2018 2:14:04 PM
ActivityId           : d2a40b4f-8621-0002-be0b-a4d22186d401
RelatedActivityId    :
ContainerLog         : C:\Windows\System32\Winevt\Logs\Microsoft-Windows-Schannel-Events%4Perf.etl
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Start
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty}
```

FPT_TST_EXT.1/WLAN

- Execution of this set of this set of TSF self-tests (20)

```
Message              : The last shutdown's success status was true. The last boot's success status was true.
Id                   : 20
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 31
Opcode               : 0
Keywords             : -9223372036854775808
RecordId             : 21774
ProviderName         : Microsoft-Windows-Kernel-Boot
ProviderId           : 15ca44ff-4d7a-4baa-bba5-0998955e531e
LogName              : System
ProcessId            : 4
ThreadId             : 8
MachineName          : LAB-EE.lab.ee
UserId               : S-1-5-18
TimeCreated          : 26-Nov-18 01:54:41 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : system
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Info
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty}
```

In addition, the evaluator has also reviewed the event with id 3033 which provides more information about the file with an invalid signature.

© 2019 Microsoft Corporation. All rights reserved.

FTA_WSE_EXT.1

- All attempts to connect to access points (8001, 8002, 8003, 11004, 11005, 11006, 12013)

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : WLAN AutoConfig service has successfully connected to a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       PHY Type: 802.11n
                       Authentication: WPA2-Enterprise
                       Encryption: AES-CCMP
                       802.1x Enabled: Yes
                       Hidden: false

Id                   : 8001
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24010
Opcode               : 190
Keywords             : -9223372036317903360
RecordId             : 1200
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 4992
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 2:59:38 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Success
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : WLAN AutoConfig service failed to connect to a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       Failure Reason:The specific network is not available.
                       RSSI: -44

Id                   : 8002
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 24010
Opcode               : 191
Keywords             : -9223372036317903360
RecordId             : 1322
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 5952
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:19:12 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Failure
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : WLAN AutoConfig service has successfully disconnected from a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       Reason: The network is disconnected by the user.

Id                   : 8003
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24010
Opcode               : 192
Keywords             : -9223372035781032448
RecordId             : 1202
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 4992
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:04:19 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Disconnect
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Wireless security stopped.

                       Network Adapter: Qualcomm Atheros QCA61x4A Wireless Network Adapter
                       Interface GUID: {ff919ea5-731a-41fd-86a1-b327096d1317}
                       Local MAC Address: [redacted]
                       Network SSID: Radius
                       BSS Type: Infrastructure
                       Security Hint: The operation was successful.

Id                   : 11004
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24012
Opcode               : 197
Keywords             : -9223372036854774272
RecordId             : 385
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 4176
ThreadId             : 124
MachineName          : WINDOWS-BA57EDP.SecEval.local
UserId               : S-1-5-18
TimeCreated          : 12/6/2018 4:50:07 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Stop
TaskDisplayName      : MsmSecurity
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Wireless security succeeded.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Local MAC Address: [redacted]
                       Network SSID: Radius
                       BSS Type: Infrastructure

Id                   : 11005
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24012
Opcode               : 198
Keywords             : -9223372036854774272
RecordId             : 1199
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 3916
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 2:59:38 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Success
TaskDisplayName      : MsmSecurity
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Wireless security failed.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Local MAC Address: [redacted]
                       Network SSID: Radius
                       BSS Type: Infrastructure
                       Peer MAC Address: [redacted]
                       Reason: Explicit Eap failure received
                       Error: 0x80420400

Id                   : 11006
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 24012
Opcode               : 199
Keywords             : -9223372036854774272
RecordId             : 1320
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 3160
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:19:10 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Failure
TaskDisplayName      : MsmSecurity
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Wireless 802.1x authentication failed.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Local MAC Address: [redacted]
                       Network SSID: Radius
                       BSS Type: Infrastructure
                       Peer MAC Address: [redacted]
                       Identity: jdoe@lab.ee
                       User: jdoe
                       Domain: LAB
                       Reason: Explicit Eap failure received
                       Error: 0x80420400
                       EAP Reason: 0x80420400
                       EAP Root cause String: Windows cannot connect to this network\nThe root certificate on the
                       server required for authentication cannot be found in the certificate store. As a result, this
                       computer will not trust the network.

                       EAP Error: 0x80420400

Id                   : 12013
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 24014
Opcode               : 205
Keywords             : -9223372036854774272
RecordId             : 1319
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 8596
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:19:10 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Failure
TaskDisplayName      : OneXAuthentication
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

FTP_ITC_EXT.1/WLAN

- All attempts to establish a trusted channel (8001, 8003)

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : WLAN AutoConfig service has successfully connected to a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       PHY Type: 802.11n
                       Authentication: WPA2-Enterprise
                       Encryption: AES-CCMP
                       802.1x Enabled: Yes
                       Hidden: false

Id                   : 8001
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24010
Opcode               : 190
Keywords             : -9223372036317903360
RecordId             : 1200
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 4992
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 2:59:38 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Success
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : WLAN AutoConfig service has successfully disconnected from a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       Reason: The network is disconnected by the user.

Id                   : 8003
Version              : 0
Qualifiers           :
Level                : 4
Task                 : 24010
Opcode               : 192
Keywords             : -9223372035781032448
RecordId             : 1202
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 4992
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:04:19 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Information
OpcodeDisplayName    : Disconnect
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

For all the previous events, the generated audit log entries include all the fields required by FAU_GEN.1.2/WLAN defined in the ***Security Target*** document.

48.2.1.3 Verdict

As the result above states, the related events have been correctly generated and they include all the information defined in the ***Security Target*** document.

Due to this, the evaluator considers that the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

© 2019 Microsoft Corporation. All rights reserved.

48.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FAU_GEN.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

49 FCS_CKM.1/WLAN

The assurance activity for the **FCS_CKM.1/WLAN** requirement is stated as follows:

> The evaluator shall verify that the TSS describes how the primitives defined and implemented by this EP are used by the TOE in establishing and maintaining secure connectivity to the wireless clients.
>
> The TSS shall also provide a description of the developer's method(s) of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also any third-party testing that is performed:
>
> - **Test 1:** The evaluator shall configure the access point so the cryptoperiod of the session key is 1 hour. The evaluator shall successfully connect the TOE to the access point and maintain the connection for a length of time that is greater than the configured cryptoperiod. The evaluator shall use a packet capture tool to determine that after the configured cryptoperiod, a re-negotiation is initiated to establish a new session key. Finally, the evaluator shall determine that the renegotiation has been successful and the client continues communication with the access point.
> - **Test 2:** The evaluator shall perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless LAN access point:
> - Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or access point.
> - Step 2: The evaluator shall configure the TOE to communicate with a WLAN access point using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key. The pre-shared key is only used for testing.
> - Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and the access point, and allow the TOE to authenticate, associate, and successfully complete the 4 way handshake with the client.
> - Step 4: The evaluator shall set a timer for 1 minute, at the end of which the evaluator shall disconnect the TOE from the wireless network and stop the sniffer.
> - Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK from the 4-way handshake frames and pre-shared key as specified in IEEE 802.11-2012.
> - Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and access point after the 4-way handshake successfully completed, and without the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the PTK to decrypt the data portion of the packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.
> - Step 7: The evaluator shall repeat Step 6 for the next 2 data frames between the TOE and access point and without frame control value 0x4208.

© 2019 Microsoft Corporation. All rights reserved.

49.1 Documentation Review activity

49.1.1 Findings

The evaluator has reviewed the section **6.2.3.2 Wireless Networking** of the ***Security Target*** document. This section states that Windows can use PRF-384 to generate AES 128-bit keys or PRF-704 to generate AES 256-bit keys using Windows RBG. In addition to this, the same section also states that Windows complies with the IEE 802.11-2012 and IEE 802.11ac-2013 standards and is able to interoperate with devices that implement the standard. All the evaluated platforms count with certificates from the Wi-Fi alliance that provides compliance for FIPS-Approved algorithms, like the *AES-CCMP*, *AES-CCMP-256* and *AES-GCMP-256*, as selected for the FCS_COP.1 requirement.

The Wi-Fi Alliance certificates for each platform are the following:

- WFA77936 for Mirosoft Surface Go.
- WFA74101 for Microsoft Surface Book 2.
- WFA71259 for Microsoft Surface Pro LTE.
- WFA71261 Microsoft Surface Laptop.
- WFA66287 for Dell Latitude 5290.
- WFA62354 for Dell Latitude 12 Rugged.
- WFA61820 for Samsung Galaxy Book 10.6 (Qualcomm Atheros QCA6174 adapter).
- WFA61581 for Samsung Galaxy Book 12 (Qualcomm Atheros QCA61x4A adapter).
- WFA66287 for HP Elite x2 1013 G3 (Intel 8265 adapter).
- WFA66287 for HP EliteBook x360 1030 G2 (Intel 8265 adapter).

The "model number" shown in the Wi-Fi Alliance certificates is used to identify the product. This varies from vendor to vendor based on their business. Some vendors will have different label for product and model number while others use the same for both. Microsoft uses either the name of the device or the name of the adapter on the device for its certified devices.

As specified in the Wi-Fi Test Suite Version 10.9.0, document, which is a software platform developed by the Wi-Fi Alliance to support certification program development and device certification, the FIPS-Approved algorithms: *AES-CCMP*, *AES-CCMP-256* and *AES-GCMP-256* are thoroughly tested during the Wi-Fi Alliance certification process.

Finally, according to section **6.2.1 Cryptographic Algorithms and Operations** of the ***Security Target*** document, Windows shall be running in FIPS validated mode. In this mode, all used algorithms must be FIPS approved. The evaluator has followed instructions included in section **5.2.5 FIPS 140 Approved mode** of the ***Operational Guidance*** document for setting this mode.

49.1.2 Verdict

The evaluator considers that the TSS provides enough information related to what primitives are used to establishing and maintaining security in wireless connections. In addition to this, TSS provides information about FIPS and Wi-Fi certifications.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

49.2 Test Activity

49.2.1 Test 1

The evaluator will attempt to verify that a re-negotiation to establish a new key exists after a defined cryptoperiod. In this case, a cryptoperiod of 1 minute for testing purposes will be used.

49.2.1.1 Setup

In this case, the evaluator has used the architecture defined for the *Setup* for *Test 1* in FMT_SMF_EXT.1/WLAN with the following elements and roles:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Kali Linux (sniffing machine) using aircrack suite

49.2.1.2 Procedure

To complete this test assurance activity, the evaluator shall establish a Session Timeout value of 60 minutes in the NPS Server.

In order to do this, the evaluator has performed the following steps:

49.2.1.2.1 Windows NPS Server

- Log in to NPS Server
- Open Network Policy Server, Expand Policies Tree, Click on Network Policies, Select Secure Wireless Connection and then right click and Properties:

© 2019 Microsoft Corporation. All rights reserved.

- Now, in constraints tab, select Session Timeout Option and set timeout in minutes (60):

49.2.1.2.2 WiFi Client

In order to avoid problems related to client disconnection by energy saving, this feature shall be disabled.

49.2.1.2.3 Kali Linux Sniffing Machine

© 2019 Microsoft Corporation. All rights reserved.

Before connecting to the Radius network, sniffing machine shall be prepared as shown below:

- Enable Monitor mode: airmon-ng start wlan0
- Start Capturing packets: airodump-ng wlan0mon -w Radius -c 7 –bssid <Wireless MAC>

Then, the evaluator shall establish a connection from the client to the network Radius and it will continue capturing packets during the established cryptoperiod (1 hour).

Observe the results.

49.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has obtained evidences related to the renegotiation after expired cryptoperiod as can be seen below:

No.	Time	Source	Destination	Protocol	Length	Info
322	15.157727151	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	100	Success
325	15.162975704	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	213	Key (Message 1 of 4)
327	15.172058825	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	247	Key (Message 3 of 4)
1386	77.528093850	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	101	Request, Identity
1389	77.554888420	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	101	Request, Identity
1392	77.600481356	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	102	Request, TLS EAP (EAP-TLS)
1395	77.608744255	BelkinIn_d7:5f:04	Microsof_2d:55:83	TLSv1.2	271	Encrypted Handshake Message
1399	77.622820314	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	100	Success
1402	77.624903123	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	213	Key (Message 1 of 4)
1404	77.629229332	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	247	Key (Message 3 of 4)

The evaluator has observed, how after the defined cryptoperiod, a new re-negotiation starts again.

49.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

49.2.2 Test 2

The evaluator will attempt to capture data frames after after the 4-way handshake has been successfully completed. Using the PTK the evaluator will decrypt data frames in order to see ASCII-readable text.

For testing purposes, the evaluator will use a passphrase to generate 256 bits PMK.

© 2019 Microsoft Corporation. All rights reserved.

49.2.2.1 Setup

In this case, an special configuration shall be configured as shown below:

- AP should be configured in WPA2 PSK
- Client should connect to previously created network
- Linux Sniffing machine with Aircrack suite installed

49.2.2.2 Procedure

According to 802.11-2012 PTK is generated by concatenation of the following attributes: PMK, AP nonce (ANonce), STA nonce (SNonce), AP MAC address, and STA MAC address as shown in the following picture:

The evaluator will use wireshark for decrypting data packets as mentioned in the tests. In order to assure that wireshark is decrypting data correctly, The evaluator has analyzed wireshark source code and has used an independent program that is able to obtain the PTK.

Reviewing Wireshark 2.6.3 public source code, PTK calculation following 802.11-2012 is located at /epan/crypt/dot11decrypt.c in *Dot11DecryptRsnaPrfX* as shown below:

```
static void
Dot11DecryptRsnaPrfX(
DOT11DECRYPT_SEC_ASSOCIATION *sa,
const UCHAR pmk[32],
const UCHAR snonce[32],
const INT x, /* for TKIP 512, for CCMP 384 */
UCHAR *ptk)
{
    UINT8 i;
    UCHAR R[100];
```

© 2019 Microsoft Corporation. All rights reserved.

```
    INT offset=sizeof("Pairwise key expansion");
    UCHAR output[80]; /* allow for sha1 overflow. */

    memset(R, 0, 100);
    memcpy(R, "Pairwise key expansion", offset);
    /* Min(AA, SPA) || Max(AA, SPA) */
    if (memcmp(sa->saId.sta, sa->saId.bssid, DOT11DECRYPT_MAC_LEN) < 0)
    {
        memcpy(R + offset, sa->saId.sta, DOT11DECRYPT_MAC_LEN);
        memcpy(R + offset+DOT11DECRYPT_MAC_LEN, sa->saId.bssid,
            DOT11DECRYPT_MAC_LEN);
    }
    else
    PMK{
        memcpy(R + offset, sa->saId.bssid, DOT11DECRYPT_MAC_LEN);
        memcpy(R + offset+DOT11DECRYPT_MAC_LEN, sa->saId.sta,
            DOT11DECRYPT_MAC_LEN);
    }
    offset+=DOT11DECRYPT_MAC_LEN*2;
    /* Min(ANonce,SNonce) || Max(ANonce,SNonce) */
    if( memcmp(snonce, sa->wpa.nonce, 32) < 0 )
    {
        memcpy(R + offset, snonce, 32);
        memcpy(R + offset + 32, sa->wpa.nonce, 32);
    }
    else
    {
        memcpy(R + offset, sa->wpa.nonce, 32);
        memcpy(R + offset + 32, snonce, 32);
    }
    offset+=32*2;
    for(i = 0; i < (x+159)/160; i++)
    {
        R[offset] = i;
        if (ws_hmac_buffer(GCRY_MD_SHA1, &output[HASH_SHA1_LENGTH * i], R, 100,
            pmk, 32)) {
            return;
        }
    }
    memcpy(ptk, output, x/8);
}
```

In order to verify that Wireshark key calculation is correct, the evaluator will use a python script that can calculate PMK and PTK from required fields including:

- SSID name
- Password
- Constant
- AP MAC
- Client MAC

© 2019 Microsoft Corporation. All rights reserved.

- ANonce
- SNonce

```
from Crypto.Cipher import AES
import hmac, hashlib, binascii, pbkdf2

def xPRF512(key, A, B):
    blen = 128
    i = 0
    R = ''
    while i <= ((blen * 8 + 159) / 160):
        hmacsha1 = hmac.new(key, A + chr(0x00) + B + chr(i), hashlib.sha1)
        i += 1
        R = R + hmacsha1.hexdigest()
        return bytearray(R[:blen].decode("hex"))


ssid = raw_input('SSID: ')
passPhrase = raw_input('Password: ')
A = 'Pairwise key expansion'

APmac = raw_input('AP MAC: ')
Clientmac = raw_input('Client MAC: ')
ANonce = raw_input('ANonce: ')
SNonce = raw_input('SNonce: ')

B =
    min(APmac, Clientmac)+max(APmac, Clientmac)+min(ANonce, SNonce)+max(ANonce, SNonce)
pmk = pbkdf2.PBKDF2(passPhrase, ssid, 4096).read(32)
ptk = xPRF512(pmk, A, B)
print "PMK:\t\t", binascii.b2a_hex(pmk), "\n"
print "PTK:\t\t", binascii.b2a_hex(ptk), "\n"
```

Where:

- AP Nonce can be obtained from the first EAPOL message:

© 2019 Microsoft Corporation. All rights reserved.

```
102 4.433665   BelkinIn_0c:f7:e0   IntelCor_bf:c7:19   EAPOL   133 Key (Message 1 of 4)
104 4.438286   IntelCor_bf:c7:19   BelkinIn_0c:f7:e0   EAPOL   157 Key (Message 2 of 4)
106 4.440319   BelkinIn_0c:f7:e0   IntelCor_bf:c7:19   EAPOL   189 Key (Message 3 of 4)
108 4.443406   IntelCor_bf:c7:19   BelkinIn_0c:f7:e0   EAPOL   133 Key (Message 4 of 4)

▸ Frame 102: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
▸ IEEE 802.11 QoS Data, Flags: ......F.
▸ Logical-Link Control
▾ 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 95
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 1]
  ▸ Key Information: 0x008a
    Key Length: 16
    Replay Counter: 1
    WPA Key Nonce: 03975ab85589fed82eed355c3f7a6874a896212f3010536b...
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000

0000  88 02 3a 01 00 28 f8 bf  c7 19 58 ef 68 0c f7 e0   ..:..(.. ..X.h...
0010  58 ef 68 0c f7 e0 00 00  00 00 aa aa 03 00 00 00   X.h..... ........
0020  88 8e 02 03 00 5f 02 00  8a 00 10 00 00 00 00 00   ....._.. ........
0030  00 00 01 03 97 5a b8 55  89 fe d8 2e ed 35 5c 3f   .....Z.U .....5\?
0040  7a 68 74 a8 96 21 2f 30  10 53 6b b0 44 bb bc 78   zht..!/0 .Sk.D..x
0050  e4 32 ed 00 00 00 00 00  00 00 00 00 00 00 00 00   .2...... ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00                                     .....
```

- SNonce can be obtained from the second EAPOL message:

```
102 4.433665   BelkinIn_0c:f7:e0   IntelCor_bf:c7:19   EAPOL   133 Key (Message 1 of 4)
104 4.438286   IntelCor_bf:c7:19   BelkinIn_0c:f7:e0   EAPOL   157 Key (Message 2 of 4)
106 4.440319   BelkinIn_0c:f7:e0   IntelCor_bf:c7:19   EAPOL   189 Key (Message 3 of 4)
108 4.443406   IntelCor_bf:c7:19   BelkinIn_0c:f7:e0   EAPOL   133 Key (Message 4 of 4)

▸ Frame 104: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits)
▸ IEEE 802.11 QoS Data, Flags: .......T
▸ Logical-Link Control
▾ 802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: Key (3)
    Length: 119
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 2]
  ▸ Key Information: 0x010a
    Key Length: 0
    Replay Counter: 1
    WPA Key Nonce: 3bbf52f89f0552190c4a59183445a46a20d1ef6543b4a095...
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000

0000  88 01 3a 01 58 ef 68 0c  f7 e0 00 28 f8 bf c7 19   ..:.X.h. ...(....
0010  58 ef 68 0c f7 e0 00 00  00 00 aa aa 03 00 00 00   X.h..... ........
0020  88 8e 01 03 00 77 02 01  0a 00 00 00 00 00 00 00   .....w.. ........
0030  00 00 01 3b bf 52 f8 9f  05 52 19 0c 4a 59 18 34   ...;.R.. .R..JY.4
0040  45 a4 6a 20 d1 ef 65 43  b4 a0 95 ec b6 07 cb 50   E.j ..eC .......P
0050  54 08 c4 00 00 00 00 00  00 00 00 00 00 00 00 00   T....... ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 b0 26 af 79 71  fb 59 90 f2 bf cb fd 81   ....&.yq .Y......
0080  ea 0b a3 00 18 30 16 01  00 00 0f ac 04 01 00 00   .....0.. ........
0090  0f ac 04 01 00 00 0f ac  02 3c 00 00 00            ........ .<...
```

- Observe the results.

49.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

As seen previously, Wireshark is able to decrypt packets following IEEE802.11-2012 standard. In order to do so, the evaluator has followed the next steps:

- View => Check "Wireless Toolbar"
- Click on 802.11 Preferences

- A popup windows will appear and then click "Edit"

- Click on '+' button

© 2019 Microsoft Corporation. All rights reserved.

- Click "+" Button and select "wpa-pwd", put the AP password in Key field and press "OK"

© 2019 Microsoft Corporation. All rights reserved.

At this point, the evaluator will be able to see decrypted traffic.

The evaluator has selected a packet with starting header "88 42" different from "08 42" and can see now several information related to this packet:

- Now expanding IEEE 802.11 Qos Data and expand CCMP parameters, the evaluator has identified TK (part of the PTK used for decrypting packets) that is the same that has been calculated using the Python script:

```
:~/Code/Python$ python ptk.py
('PTK Size: ', 64)
ptk:            70e79d96e5027e6e0618ca4a2b67aa2f16585034e3f329c105e00a87c785812d
581543e436f866e251ecb1c9e4257ed69c24d0c185bbfd411c56fda9fbdc9ad4
```

Wireshark calculated TK (part of previously calculated PTK) can be seen below:

© 2019 Microsoft Corporation. All rights reserved.

```
1505 11.231426   192.168.2.100   192.168.2.144   HTTP   3088 ContinuationContinuation
1506 11.231426   192.168.2.100   192.168.2.144   HTTP   1564 Continuation
1507 11.231426   192.168.2.100   192.168.2.144   HTTP   3088 ContinuationContinuation

▸ Frame Control Field: 0x8842
  .000 0000 0011 0000 = Duration: 48 microseconds
  Receiver address: IntelCor_bf:c7:19 (00:28:f8:bf:c7:19)
  Transmitter address: BelkinIn_0c:f7:e0 (58:ef:68:0c:f7:e0)
  Destination address: IntelCor_bf:c7:19 (00:28:f8:bf:c7:19)
  BSS Id: BelkinIn_0c:f7:e0 (58:ef:68:0c:f7:e0)
  STA address: IntelCor_bf:c7:19 (00:28:f8:bf:c7:19)
  .... .... .... 0000 = Fragment number: 0
  0001 0000 1001 .... = Sequence number: 265
▸ Qos Control: 0x0080
▾ CCMP parameters
    CCMP Ext. Initialization Vector: 0x000000000108
    Key Index: 0
    [TK: 581543e436f866e251ecb1c9e4257ed6]
    [PMK: faf757d147ab2abd0d423a736e45bf46a84f449b24f21520f173349fcc9eb1b3]
▸ IEEE 802.11 Aggregate MSDU

0000  88 42 30 00 00 28 f8 bf  c7 19 58 ef 68 0c f7 e0   .B0..(.. ..X.h...
0010  24 5e be 11 cb e7 90 10  80 00 08 01 00 20 00 00   $^...... ..... ..
0020  00 00 f3 65 b2 0b 2b bb  7d 7a cc de ba eb 75 cb   ...e..+. }z....u.
0030  89 70 e5 de 82 d6 56 25  1e 58 c6 88 29 cc 73 4f   .p....V% .X..).sO
0040  57 6a 8c ec 93 5c 57 99  3c 31 3f bb d8 4e a8 30   Wj...\W. <1?..N.0
0050  8f 1e e5 50 73 48 f2 b1  a0 05 7e e8 17 2f cf 1f   ...PsH.. ..~../..
0060  9d 84 ec 3d f0 cc db 4c  95 87 c5 d1 f6 af 36 11   ...=...L ......6.
0070  e0 86 8e 4f e3 2c 70 8e  ae 16 a5 c5 09 a8 f4 b6   ...O.,p. ........
0080  a7 9c 21 8d 32 17 8e 59  44 18 9b 13 f5 0b 28 6f   ..!.2..Y D.....(o
0090  3b 09 16 13 53 1a a8 65  5e 3b 24 3b 84 fe d6 bb   ;...S..e ^;$;....
00a0  d2 6a 75 a0 7b 0d 0f c5  de f8 bc a5 7f 5e db 8c   .ju.{... .....^..
00b0  3c c3 a7 5b 13 d5 56 5e  45 a4 44 7b c8 33 9b bf   <..[..V^ E.D{.3..
00c0  e0 59 60 60 01 7d d7 1a  04 01 48 83 f5 6a 97 02   .Y``.}.. ..H..j..
00d0  ff bf 4d 14 e5 3a 24 50  ef 93 35 48 13 22 3c 86   ..M..:$P ..5H."<.
00e0  58 75 d2 42 6a 63 7a 8d  1a 70 24 40 e8 cb a2 b6   Xu.Bjcz. .p$@....
00f0  c7 d4 92 b2 02 9b bd 03  ef a6 9f ef af 31 10 7a   ........ .....1.z
0100  cd 94 a4 a2 6b 2e ca 74  38 46 ca 7c ac 16 25 06   ....k..t 8F.|..%.
0110  3b 8d 07 ba fe 2c 8a 81  7f 06 e4 52 15 18 53 bd   ;....,.. ...R..S.
0120  cc 9f 77 80 64 b8 a5 0b  69 e8 4c 81 db 08 0c 5f   ..w.d... i.L....

Frame (3088 bytes)   Decrypted CCMP data (3046 bytes)
```

Now clicking on “Decrypted CCMP data” Tab on the bottom left of the window, the evaluator will be able to see decrypted package including ASCII characters:

```
00c0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00d0  20 20 20 20 20 20 20 20  20 3c 69 6e 70 75 74 20                <input
00e0  6e 61 6d 65 3d 22 5f 5f  52 65 71 75 65 73 74 56   name="__ RequestV
00f0  65 72 69 66 69 63 61 74  69 6f 6e 54 6f 6b 65 6e   erificat ionToken
0100  22 20 74 79 70 65 3d 22  68 69 64 64 65 6e 22 20   " type=" hidden"
0110  76 61 6c 75 65 3d 22 43  66 44 4a 38 47 42 4a 67   value="C fDJ8GBJg
0120  4f 54 6e 4b 44 78 41 6f  4d 76 44 52 7a 5a 63 42   OTnKDxAo MvDRzZcB
0130  5f 64 62 51 75 37 4d 47  76 77 53 61 74 49 62 6d   _dbQu7MG vwSatIbm
0140  62 50 35 38 6d 43 54 6f  69 6e 5a 38 55 4d 76 4f   bP58mCTo inZ8UMvO
0150  56 50 4c 57 71 2d 42 4f  68 33 51 73 57 39 6b 54   VPLWq-BO h3QsW9kT
0160  35 77 79 79 77 66 55 74  4a 71 30 36 75 78 52 4b   5wyywfUt Jq06uxRK
0170  79 45 79 35 53 62 53 59  64 74 61 4d 6f 68 6e 78   yEy5SbSY dtaMohnx
0180  63 51 6a 2d 52 5a 63 4d  2d 75 66 49 76 53 46 43   cQj-RZcM -ufIvSFC
0190  6a 5f 6d 45 38 37 41 71  64 4e 77 49 2d 68 30 61   j_mE87Aq dNwI-h0a
01a0  70 5f 78 6c 41 4e 33 75  79 75 41 44 56 33 7a 35   p_xlAN3u yuADV3z5
01b0  36 52 4a 66 37 2d 65 37  5a 45 44 79 31 34 39 59   6RJf7-e7 ZEDy149Y
01c0  71 44 35 69 55 66 5f 34  2d 38 52 36 7a 6e 45 6b   qD5iUf_4 -8R6znEk
01d0  4c 4d 6d 4c 30 36 48 48  71 58 35 4f 77 22 20 2f   LMmL06HH qX5Ow" /
01e0  3e 3c 2f 66 6f 72 6d 3e  0d 0a 20 20 20 20 20 20   ></form> ..

Frame (3088 bytes)   Decrypted CCMP data (3046 bytes)
```

49.2.2.4 Verdict

© 2019 Microsoft Corporation. All rights reserved.

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

49.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_CKM.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

50 FCS_CKM.2/WLAN

The assurance activity for the **FCS_CKM.2/WLAN** requirement is stated as follows:

> The evaluator shall check the TSS to ensure that it describes how the GTK is unwrapped prior to being installed for use on the TOE using the AES implementation specified in this EP.
>
> - **Test 1:** The evaluator shall configure the access point so the cryptoperiod of the session key is 1 hour. The evaluator shall successfully connect the TOE to the access point and maintain the connection for a length of time that is greater than the configured cryptoperiod. The evaluator shall use a packet capture tool to determine that after the configured cryptoperiod, a re-negotiation is initiated to establish a new session key. Finally, the evaluator shall determine that the renegotiation has been successful and the client continues communication with the access point.
> - **Test 2:** The evaluator shall perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless LAN access point:
> - Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or access point.
> - Step 2: The evaluator shall configure the TOE to communicate with a WLAN access point using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key. The pre-shared key is only used for testing.
> - Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and the access point, and allow the TOE to authenticate, associate, and successfully complete the 4 way handshake with the client.*
> - Step 4: The evaluator shall set a timer for* 1 minute, at the end of which the evaluator shall disconnect the TOE from the wireless network and stop the sniffer.
> - Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK and GTK from the 4-way handshake frames and pre-shared key as specified in IEEE 802.11-2012.
> - Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and access point after the 4-way handshake successfully completed, and with the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the GTK to decrypt the data portion of the selected packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.
> - Step 7: The evaluator shall repeat Step 6 for the next 2 data frames with frame control value 0x4208.

© 2019 Microsoft Corporation. All rights reserved.

50.1 Documentation Review activity

50.1.1 Findings

The evaluator has reviewed the section **6.2.3.2 Wireless Networking** of the ***Security Target*** document. This section states that Windows implements key wrapping and unwrapping according to the NIST SP 800-38F specification (the "KW" mode) and so unwraps the Wi-Fi Group Temporal Key (GTK) which was sent by the access point without exposing GTK to the network. In addition to this, it states that Windows complies with the IEE 802.11-2012 and IEE 802.11ac-2013 standards and is able to interoperate with devices that implement the standard.

According to section **5.2.5 FIPS 140 Approved mode** and following the ***Operational Guidance***, Windows shall be running in FIPS validated mode. In this mode, all used algorithms must be FIPS approved.

50.1.2 Verdict

The evaluator considers that the TSS provides enough information related to what primitives are used to establishing and maintaining security in wireless connections. In addition to this, TSS provides information about FIPS and Wi-Fi certifications.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

50.2 Test Activity

50.2.1 Test 1

The evaluator will attempt to capture data frames after after the 4-way handshake has been successfully completed. Using the GTK the evaluator will decrypt data frames in order to see ASCII-readable text.

50.2.1.1 Setup

In this case, we are going to use the architecture and the captured file defined at FCS_CKM.1/WLAN with the following elements and roles:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Kali Linux (sniffing machine) using aircrack suite

© 2019 Microsoft Corporation. All rights reserved.

50.2.1.2 Procedure

To complete this test assurance activity, we will use the previously captured file in **FCS_CKM.1/WLAN**.

The evaluator will use wireshark for decrypting data packets as mentioned in the tests.

Reviewing Wireshark 2.6.3 public source code, PTK calculation following 802.11-2012 for AES-CCMP is located at /epan/crypt/dot11decrypt.c in *Dot11DecryptWPABroadcastKey* as shown below:

```
} else if (key_version == DOT11DECRYPT_WPA_KEY_VER_AES_CCMP){
/* AES CCMP key */

guint8 key_found;
guint8 key_length;
guint16 key_index;
guint8 *decrypted_data;
/* Unwrap the key; the result is key_bytes_len in length */
decrypted_data = AES_unwrap(decryption_key, 16, szEncryptedKey, key_bytes_len);
/* With WPA2 what we get after Broadcast Key decryption is an actual RSN
    structure.
The key itself is stored as a GTK KDE
WPA2 IE (1 byte) id = 0xdd, lengtow in the serverh (1 byte), GTK OUI (4 bytes),
    key index (1 byte) and 1 reserved byte. Thus we have to
pass pointer to the actual key with 8 bytes offset */
key_found = FALSE;
key_index = 0;
/* Parse Key data until we found GTK KDE */
/* GTK KDE = 00-0F-AC 01 */
while(key_index < (key_bytes_len - 6) && !key_found){
guint8 rsn_id;
guint32 type;
/* Get RSN ID */
rsn_id = decrypted_data[key_index];
type = ((decrypted_data[key_index + 2] << 24) +
(decrypted_data[key_index + 3] << 16) +
(decrypted_data[key_index + 4] << 8) +
(decrypted_data[key_index + 5]));
if (rsn_id == 0xdd && type == 0x000fac01) {
key_found = TRUE;
} else {
key_index += decrypted_data[key_index+1]+2;
}
}
```

Observe the results

50.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

© 2019 Microsoft Corporation. All rights reserved.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

Using the previous captured packets from FCS_CKM.1/WLAN and decrypting packets as shown in FCS_CKM.1/WLAN *Test 2*, the evaluator has found packets starting with 08 42:

```
130 4.619009    IntelCor_bf:c7:19    Broadcast              ARP        36 Who has 192.168.2.1? Tell 192.168.2.144
131 4.619531    IntelCor_bf:c7:19    BelkinIn_0c:f7:e0      802.11     26 QoS Null function (No data), SN=0, FN=0, Flag
132 4.619531                         IntelCor_bf:c7:19 (00:2802.11     10 Acknowledgement, Flags=........

    0... .... = Order flag: Not strictly ordered
.000 0000 0000 0000 = Duration: 0 microseconds
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
Transmitter address: BelkinIn_0c:f7:e0 (58:ef:68:0c:f7:e0)
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
Source address: IntelCor_bf:c7:19 (00:28:f8:bf:c7:19)
BSS Id: BelkinIn_0c:f7:e0 (58:ef:68:0c:f7:e0)
STA address: Broadcast (ff:ff:ff:ff:ff:ff)
.... .... .... 0000 = Fragment number: 0
1000 1010 1110 .... = Sequence number: 2222
▾CCMP parameters
  CCMP Ext. Initialization Vector: 0x0000000008B6
  Key Index: 1
  [GTK: 94b5a15594d7d00ad114af9e2317859f]
▸Logical-Link Control
▸Address Resolution Protocol (request)

0000  08 42 00 00 ff ff ff ff  ff ff 58 ef 68 0c f7 e0   .B...... ..X.h...
0010  00 28 f8 bf c7 19 e0 8a  b6 08 00 60 00 00 00 00   .(...... ...`....
0020  9f 39 53 96 e9 55 6e 96  5c 0b ca 43 3a 48 df 74   .9S..Un. \..C:H.t
0030  bd f5 59 9d 43 ae 8a 09  e4 ce 80 c3 0e d8 c0 f8   ..Y.C... ........
0040  48 8f 67 44 74 21 82 90  b8 bd 7f 77               H.gDt!.. ...w
```

The evaluator has seen that the GTK has been calculated and now is able to decrypt this broadcast message in order to see the ARP query in clear text:

```
▾Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: IntelCor_bf:c7:19 (00:28:f8:bf:c7:19)
  Sender IP address: 192.168.2.144
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.2.1

0000  aa aa 03 00 00 00 08 06  00 01 08 00 06 04 00 01   ........ ........
0010  00 28 f8 bf c7 19 c0 a8  02 90 00 00 00 00 00 00   .(...... ........
0020  c0 a8 02 01                                        ....
```

50.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

© 2019 Microsoft Corporation. All rights reserved.

50.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_CKM.2/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

51 FCS_TLSC_EXT.1/WLAN

The assurance activity for the **FCS_TLSC_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.
>
> The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.
>
> The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).
>
> The evaluator shall check that the OPE guidance contains instructions for the administrator to configure the list of Certificate Authorities that are allowed to sign certificates used by the authentication server that will be accepted by the TOE in the EAP-TLS exchange, and instructions on how to specify the algorithm suites that will be proposed and accepted by the TOE during the EAP-TLS exchange.
>
> The evaluator shall write, or the ST author shall provide, an application for the purposes of testing TLS.
>
> The evaluator shall also perform the following tests:
>
> - **Test 1**: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
> - **Test 2**: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not Documentation Review activity established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.
> - **Test 3**: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA certificate while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.

© 2019 Microsoft Corporation. All rights reserved.

- **Test 4**: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.
- **Test 5**: The evaluator shall perform the following modifications to the traffic:
  - Change the TLS version selected by the server in the Server Hello to a non- supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.
  - Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.
  - Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
  - Modify the signature block in the Server's Key Exchange handshake message and verify that the client rejects the connection after receiving the Server Key Exchange message.
  - Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.
  - Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the client denies the connection.

51.1 Documentation Review activity

51.1.1 Findings

The evaluator has reviewed the section **6.3.2 Available EAP-TLS ciphersuites** of the ***Operational Guidance*** document. This section states that Windows can use different ciphersuites including among others those requested:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5430
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5430

Complete set of TLS cipher suites implemented can be found here: Cipher Suites in TLS/SSL (Schannel SSP)

These cipher suites match with the selected ones in the requirement definition. The Cipher suites listed above are a subset of the ones implemented in the Schannel library, as it can be checked in the previous URL.

© 2019 Microsoft Corporation. All rights reserved.

In addition, sections **6.3.3 Configuring with MDM**, **6.3.4 Configuring with PowerShell** and **6.3.5 Configuring with group policy** of the ***Operational Guidance*** document describes how to configure these TLS cipher suites. In addition to this, the information can be found in the following MSDN link included in the ***Operational Guidance***: How to restrict the use of certain cryptographic algorithms and protocols in Schannel.dll

Selection of TLS cipher suites in the TLS handshake process is performed according to the order defined which can be configured as described in the links listed above.

51.1.2 Verdict

The evaluator considers that the TSS provides enough information related to what cipher suites are available.

The ***Operational Guidance*** includes information regarding how to configure correctly the TOE for testing purposes including ciphersuite selection and associated restrictions.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

51.2 Test Activity

51.2.1 Test 1

The evaluator will attempt to force the client to use the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5430
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5430

51.2.1.1 Setup

In this case, the evaluator will use the general NPS architecture composed by the following elements:

> - Windows NPS Server (EAP-TLS)
> - Windows Client using domain user (FIPS Enabled)
> - AP WPA2-Enterprise mode
> - W10WLANClientAutomator.ps1
> - W10WLANServerAutomator.ps1

© 2019 Microsoft Corporation. All rights reserved.

51.2.1.2 Procedure

To complete this test assurance activity, the evaluator has developed a suite of scripts to configure the server and the client according to the test to be performed.

The server is configured with only one ciphersuite and its corresponding certificate (RSA/ECDSA) and the client is configured with the corresponding user certificate (RSA/ECDSA) to perform the connection.

These are some server functions about the *Test 1* for FCS_TLSC_EXT.1/WLAN in W10WLANServerAutomator.ps1:

Main menu:

```
# FCS_TLSC_EXT.1 — Test 1
function Fcs—Tlsc—Ext—1—1 {
    Write—Host "============================"
    Write—Host "FCS_TLSC_EXT.1/WLAN — Test 1"
    Write—Host "============================"
    Write—Host "Please select ciphersuite to set:"
    Write—Host "1. TLS_RSA_WITH_AES_128_CBC_SHA"
    Write—Host "2. TLS_RSA_WITH_AES_256_CBC_SHA"
    Write—Host "3. TLS_RSA_WITH_AES_128_CBC_SHA256"
    Write—Host "4. TLS_RSA_WITH_AES_256_CBC_SHA256"
    Write—Host "5. TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"
    Write—Host "6. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256"
    Write—Host "7. TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384"

    $input = Read—Host "Please select"
    switch ($input) {
        '1' { AddRSAConfig | setCipherSuite "TLS_RSA_WITH_AES_128_CBC_SHA" }
        '2' { AddRSAConfig | setCipherSuite "TLS_RSA_WITH_AES_256_CBC_SHA" }
        '3' { AddRSAConfig | setCipherSuite "TLS_RSA_WITH_AES_128_CBC_SHA256" }
        '4' { AddRSAConfig | setCipherSuite "TLS_RSA_WITH_AES_256_CBC_SHA256" }
        '5' { AddECDSAConfig | setCipherSuite
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384" }
        '6' { AddECDSAConfig | setCipherSuite
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256" }
        '7' { AddECDSAConfig | setCiphersuite
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384" } #c024
        Default {}
    }
}
```

Default RSA server configuration:

```
# Adding RSA default configuration
function AddRSAConfig {
    Write—Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write—Warning "[I]: Adding the server certificate with serverAuth"
    AddServerCert ".\WLAN\NPS_OK\NPS_OK_SERVER_AUTH_CERT.pfx"
    Write—Warning "[I]: Importing the default NPS configuration"
    Import—NpsConfiguration —Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
```

© 2019 Microsoft Corporation. All rights reserved.

```
  WLAN\NPS_OK\NPS_SERVER_AUTH_CONFIG.xml"
}
```

Default ECDSA server configuration:

```
# Adding ECDSA default configuration
function AddECDSAConfig {
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the ECDSA server certificate"
    AddServerCert
        ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_256_SERVER_CERT.pfx"
    Write-Warning "[I]: Importing the ECDSA NPS configuration"
    Import-NpsConfiguration -Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
  WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_ECDSA_256_CONFIG.xml"
}
```

These are some client functions about the *Test 1* for FCS_TLSC_EXT.1/WLAN in W10WLANClientAutomator.ps1:

Main menu:

```
# FCS_TLSC_EXT.1 - Test 1
function Fcs-Tlsc-Ext-1-1 {
    Write-Host "============================"
    Write-Host "FCS_TLSC_EXT.1/WLAN - Test 1"
    Write-Host "============================"
    Write-Host "Please select ciphersuite to set:"
    Write-Host "1. RSA"
    Write-Host "2. ECDSA"

    $input = Read-Host "Please select"
    switch ($input) {
        '1' { AddRSAUserConfig | JDoe_Good }
        '2' { AddECDSAUserConfig }
        Default {}
    }
}
```

Default RSA client configuration:

```
# Default RSA User configuration
function AddRSAUserConfig {
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the user certificate"
}
```

Default ECDSA client configuration:

```
# Default ECDSA User Configuration
function AddECDSAUserConfig {
    Write-Warning "[I]: Adding the CA"
```

© 2019 Microsoft Corporation. All rights reserved.

```
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the ECDSA user certificate"
    AddCert ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_256_USER_CERT.pfx"
}
```

51.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

Using the captured packets, the evaluator has tested all the ciphersuites. In the following capture, the sample for TLS_RSA_WITH_AES_128_CBC_SHA can be seen below:

No.	Time	Source	Destination	Protocol	Length	Info
1	-14.243684	10.10.20.25	10.10.20.40	RADIUS	222	Access-Request id=39
4	-14.055657	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=39
5	-14.032734	10.10.20.25	10.10.20.40	RADIUS	386	Access-Request id=40
7	-13.993224	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=40
8	-13.978223	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=41
9	-13.977166	10.10.20.40	10.10.20.25	RADIUS	392	Access-Challenge id=41
11	-13.878774	10.10.20.25	10.10.20.40	RADIUS	266	Access-Request id=42
12	-13.878109	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=42
13	-13.874688	10.10.20.25	10.10.20.40	RADIUS	861	Access-Request id=43
14	-10.861791	10.10.20.25	10.10.20.40	RADIUS	861	Access-Request id=43, Duplicate Request
17	-6.474833	10.10.20.40	10.10.20.25	RADIUS	211	Access-Challenge id=43
18	-6.458920	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=44
19	-6.457929	10.10.20.40	10.10.20.25	RADIUS	260	Access-Accept id=44

```
Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
    Random: 5bfbd01aaee97ec90c96e685b06bb527ee29a0087f3fd686...
    Session ID Length: 32
    Session ID: 45140000f95431bbac98e617efecc247e22bb3b6687e60f6...
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
```

51.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

51.2.2 Test 2

The evaluator will attempt to establish a connection with a server providing a valid server certificate containing Server Authentication Purpose. Then, the evaluator will use a valid server certificate without Server Authentication purpose and will verify that the connection is not established.

© 2019 Microsoft Corporation. All rights reserved.

51.2.2.1 Setup

In this case, the evaluator will use the general NPS architecture composed by the following elements:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- W10WLANClientAutomator.ps1
- W10WLANServerAutomator.ps1

51.2.2.2 Procedure

To complete this test assurance activity, the evaluator has developed a suite of scripts to configure the server and the client according to the test to be performed:

For the first part of the test, the NPS server is configured with a RSA certificate with ServerAuth purpose.

For the second part of the test, the NPS server is configured with a RSA certificate without ServerAuth purpose.

In both tests, the client is configured with a RSA user certificate.

These are some server functions about the *Test 2* for FCS_TLSC_EXT.1/WLAN in W10WLANServerAutomator.ps1:

Main menu

```
# FCS_TLSC_EXT.1 - Test 2
function Fcs-Tlsc-Ext-1-2 {
    Write-Host "============================"
    Write-Host "FCS_TLSC_EXT.1/WLAN - Test 2"
    Write-Host "============================"
    Write-Host "Please select option to test:"
    Write-Host "1. Certificate containing ServerAuth purpose"
    Write-Host "2. Certificate not containing ServerAuth purpose"

    $input = Read-Host "Please select"
    switch ($input) {
        '1' { setServerAuthConfig }
        '2' { setNoServerAuthConfig }
        Default {}
    }
}
```

Configuration for NPS Server with ServerAuth purpose

```
FCS_TLSC_EXT.1/WLAN - Test 2.1
function setServerAuthConfig {
 Write-Warning "[I]: Setting the configuration to test FCS_TLSC_EXT.1/WLAN -
     Test 2.1"
 setCipherSuite "TLS_RSA_WITH_AES_128_CBC_SHA"
 Write-Warning "[I]: Adding the CA"
```

© 2019 Microsoft Corporation. All rights reserved.

```
 AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
 Write-Warning "[I]: Adding the server certificate with serverAuth"
 AddServerCert ".\WLAN\NPS_OK\NPS_OK_SERVER_AUTH_CERT.pfx"
 Write-Warning "[I]: Importing the default NPS configuration"
 Import-NpsConfiguration -Path "C:\Users\Administrator\Desktop\certs\scripts_w10\
 WLAN\NPS_OK\NPS_SERVER_AUTH_CONFIG.xml"
}
```

Configuration for NPS Server without ServerAuth purpose

```
# FCS_TLSC_EXT.1/WLAN - Test 2.2
function setNoServerAuthConfig {
 Write-Warning "[I]: Setting the configuration to test FCS_TLSC_EXT.1/WLAN -
     Test 2.2"
 setCipherSuite "TLS_RSA_WITH_AES_128_CBC_SHA"
 Write-Warning "[I]: Adding the CA"
 AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
 Write-Warning "[I]: Adding the server certificate without serverAuth"
 AddServerCert ".\WLAN\NPS_OK\NPS_OK_EMAIL_REPLICATION_CERT.pfx"
 Write-Warning "[I]: Importing the NPS configuration with a certificate without
     serverAuth"
 Import-NpsConfiguration -Path "C:\Users\Administrator\Desktop\certs\scripts_w10\
 WLAN\NPS_OK\NPS_EMAIL_REPLICATION_CONFIG.xml"
}
```

These are some server functions about the *Test 2* for FCS_TLSC_EXT.1/WLAN in W10WLANClientAutomator.ps1:

Sets the default CA

```
# Default RSA User configuration
function AddRSAUserConfig {
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the user certificate"
}
```

Imports the certificate for user 'John Doe'

```
# Imports the good certificate for John Doe
function JDoe_Good {
    AddCA ".\FTP_ITC_EXT_1_1\CA.pfx"
    AddCert ".\FTP_ITC_EXT_1_1\jdoe.pfx"
}
```

51.2.2.3 Procedure

To complete this test assurance activity, the evaluator shall perform the following steps:

Valid certificate with Server Authentication Purpose

© 2019 Microsoft Corporation. All rights reserved.

- Establish the connection from the TOE
- Observe the results

Valid certificate without Server Authentication Purpose

- Try to establish the connection from the TOE
- Observe the results

51.2.2.4 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has followed has tested valid general configuration of Environment and has established connection with a valid certificate with Server Authentication purpose as shown below:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=81
2	0.009047	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=81
3	0.013315	10.10.20.25	10.10.20.40	RADIUS	349	Access-Request id=82
5	0.025150	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=82
6	0.078226	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=83
7	0.078860	10.10.20.40	10.10.20.25	RADIUS	392	Access-Challenge id=83
9	1.010177	10.10.20.25	10.10.20.40	RADIUS	269	Access-Request id=84
10	1.011574	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=84
11	1.033398	10.10.20.25	10.10.20.40	RADIUS	257	Access-Request id=85
12	1.047668	10.10.20.40	10.10.20.25	RADIUS	211	Access-Challenge id=85
13	1.064827	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=86
14	1.065590	10.10.20.40	10.10.20.25	RADIUS	260	Access-Accept id=86

```
> subject: rdnSequence (0)
> subjectPublicKeyInfo
v extensions: 10 items
  > Extension (id-ms-certificate-template)
  v Extension (id-ce-extKeyUsage)
      Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
    v KeyPurposeIDs: 2 items
        KeyPurposeId: 1.3.6.1.5.5.7.3.1 (id-kp-serverAuth)
        KeyPurposeId: 1.3.6.1.5.5.7.3.2 (id-kp-clientAuth)
```

The evaluator has loaded the NPS configuration file with valid certificate and invalid purpose and has verified that connection is not established.

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.40	RADIUS	222	Access-Request id=0
2	0.014164	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=0
3	0.019632	10.10.20.25	10.10.20.40	RADIUS	418	Access-Request id=1
5	0.030312	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=1
6	0.044827	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=2
7	0.045368	10.10.20.40	10.10.20.25	RADIUS	369	Access-Challenge id=2
8	0.094502	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=3
9	0.097037	10.10.20.40	10.10.20.25	RADIUS	86	Access-Reject id=3

```
extensions: 10 items
  Extension (id-ms-certificate-template)
  Extension (id-ce-extKeyUsage)
    Extension Id: 2.5.29.37 (id-ce-extKeyUsage)
    KeyPurposeIDs: 1 item
      KeyPurposeId: 1.3.6.1.4.1.311.21.19 (id-ms-ds-email-replication)
```

51.2.2.5 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

51.2.3 Test 3

The evaluator will attempt to establish a server ciphersuite that does not match with the server certificate by modifying packets in real time.

51.2.3.1 Setup

In this case, the evaluator will use the general NPS architecture composed by the following elements:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- W10WLANClientAutomator.ps1
- W10WLANServerAutomator.ps1

For modifying packets, the following Rehtse configuration file will be used:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\xc0\\x24.*"
            },
            "replacement":{
                "regex":"\\xc0\\x24",
```

© 2019 Microsoft Corporation. All rights reserved.

```
                "replacement":"\\x00\\x2f"
            }
        }
    ]
}
```

This snippet will change default server ciphersuite from TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) to TLS_RSA_WITH_AES_128_CBC_SHA (0x002f).

51.2.3.2 Procedure

To complete this test assurance activity, the evaluator has developed a suite of scripts to configure the server and the client according to the test to be performed:

These are some server functions about the *Test 3* for FCS_TLSC_EXT.1/WLAN in W10WLANServerAutomator.ps1:

Default NPS Server configuration

```
# Default Configuration
function Fcs-Tlsc-Ext-Default {
    Write-Warning "[I]: Setting the default configuration"
    setCiphersuite "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384"
    AddECDSAConfig
}


# Adding ECDSA default configuration
function AddECDSAConfig {
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the ECDSA server certificate"
    AddServerCert
        ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_256_SERVER_CERT.pfx"
    Write-Warning "[I]: Importing the ECDSA NPS configuration"
    Import-NpsConfiguration -Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
  WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_ECDSA_256_CONFIG.xml"
}
```

These are some server functions about the *Test 3* for FCS_TLSC_EXT.1/WLAN in W10WLANClientAutomator.ps1:

```
# Default ECDSA User Configuration
function AddECDSAUserConfig {
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the ECDSA user certificate"
    AddCert ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_256_USER_CERT.pfx"
}
```

© 2019 Microsoft Corporation. All rights reserved.

- Run Rehtse with appropriate config.json file
- Capture packets using Wireshark
- Observe the results

51.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has identified that TOE disconnects after receiving the server's certificate. In the following captures, the modified ciphersuite can be seen before and after the modification is performed:

No.	Time	Source	Destination	Protocol	Length	Info
80	102.467346080	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=146
81	102.467308354	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=146
82	102.469943480	10.10.20.40	10.10.20.25	RADIUS	1419	Access-Challenge id=146
83	102.493128288	10.10.20.40	10.10.10.1	RADIUS	1419	Access-Challenge id=146

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▸ Random: 5bfd2b645bb633dad74237f474d02c091f6a0bc232931cc3...
    Session ID Length: 32
    Session ID: c4000000902d2a0814c6e2a37af8050e068d182911445e57...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

No.	Time	Source	Destination	Protocol	Length	Info
80	102.467346080	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=146
81	102.467308354	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=146
82	102.469943480	10.10.20.40	10.10.20.25	RADIUS	1419	Access-Challenge id=146
83	102.493128288	10.10.20.40	10.10.10.1	RADIUS	1419	Access-Challenge id=146

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▸ Random: 5bfd2b645bb633dad74237f474d02c091f6a0bc232931cc3...
    Session ID Length: 32
    Session ID: c4000000902d2a0814c6e2a37af8050e068d182911445e57...
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

51.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

© 2019 Microsoft Corporation. All rights reserved.

Therefore, the **PASS** verdict is assigned to **Test 3**.

51.2.4 Test 4

The evaluator will configure the server forcing it to use TLS_NULL_WITH_NULL_NULL ciphersuite and will verify that TOE rejects the connection.

51.2.4.1 Setup

In this case, the evaluator will use the general NPS architecture composed by the following elements:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- W10WLANClientAutomator.ps1
- W10WLANServerAutomator.ps1

For modifying packets, the following Rehtse configuration file will be used:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\xc0\\x24.*"
            },
            "replacement":{
                "regex":"\\xc0\\x24",
                "replacement":"\\x00\\x00"
            }
        }
    ]
}
```

51.2.4.2 Procedure

The procedure is the same as defined in **Test 3**.

51.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has identified that TOE disconnects after receiving the ciphersuite. In the following captures, the modified ciphersuite can be seen before and after the modification is performed:

No.	Time	Source	Destination	Protocol	Length	Info
7	10.474335841	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=169
8	10.484637254	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=169
10	10.515725335	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=170
11	10.516900860	10.10.20.40	10.10.20.25	RADIUS	1418	Access-Challenge id=170
12	10.474288940	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=169
13	10.485450396	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=169
14	10.515703198	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=170
15	10.517332636	10.10.20.40	10.10.10.1	RADIUS	1418	Access-Challenge id=170

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▸ Random: 5bfd2f09a7a008d7a58221815095ff32fec62af91efa4efe...
    Session ID Length: 32
    Session ID: 391e0000829260ad756a5afa76e06a54d153f1f56dd84da2...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
```

No.	Time	Source	Destination	Protocol	Length	Info
7	10.474335841	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=169
8	10.484637254	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=169
10	10.515725335	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=170
11	10.516900860	10.10.20.40	10.10.20.25	RADIUS	1418	Access-Challenge id=170
12	10.474288940	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=169
13	10.485450396	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=169
14	10.515703198	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=170
15	10.517332636	10.10.20.40	10.10.10.1	RADIUS	1418	Access-Challenge id=170

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▸ Random: 5bfd2f09a7a008d7a58221815095ff32fec62af91efa4efe...
    Session ID Length: 32
    Session ID: 391e0000829260ad756a5afa76e06a54d153f1f56dd84da2...
    Cipher Suite: TLS_NULL_WITH_NULL_NULL (0x0000)
    Compression Method: null (0)
    Extensions Length: 9
  ▸ Extension: extended_master_secret (len=0)
  ▸ Extension: renegotiation_info (len=1)
```

51.2.4.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4**.

51.2.5 Test 5

The evaluator will perform the following modifications by using MITM tool:

© 2019 Microsoft Corporation. All rights reserved.

- Change the TLS version to an unsupported version
- Modify server nonce in Server Hello
- Modify server ciphersuite after Client Hello
- Modify signature block in Server's Key Exchange
- Modify Server finished handshake
- Send a garbled message after ChangeCipherSpec Message

51.2.5.1 Setup

In this case, the evaluator will use the general NPS architecture composed by the following elements:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- W10WLANClientAutomator.ps1
- W10WLANServerAutomator.ps1

In order to force the server to perform traffic modifications, the evaluator will use Rehtse tool with different configurations for each test.

51.2.5.2 Procedure

To complete this test assurance activity, the evaluator shall perform the following steps:

In all cases, the initial procedure is the same as defined in **Test 3**.

a) Change TLS version

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\x00\\x00\\x51\\x03\\x03.*"
            },
            "replacement":{
                "regex":"\\x00\\x00\\x51\\x03\\x03",
                "replacement":"\\x00\\x00\\x51\\x03\\x04"
            }
        }
    ]
}
```

In this case, the last bytes are related to TLS Version 1.2 (0x0303) and it will be changed by TLS version 1.3 (0x0304).

© 2019 Microsoft Corporation. All rights reserved.

b) Modify Server Nonce

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\x03\\x03\\x5b.*"
            },
            "replacement":{
                "regex":"\\x03\\x03\\x5b",
                "replacement":"\\x03\\x03\\xee"
            }
        }
    ]
}
```

The configuration will change the first part of the server nonce (Date Time) to another value.

c) Modify Cipher Suite after Client Hello

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\xc0\\x24.*"
            },
            "replacement":{
                "regex":"\\xc0\\x24",
                "replacement":"\\x00\\x2f"
            }
        }
    ]
}
```

This snippet will change default server ciphersuite from TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) to TLS_RSA_WITH_AES_128_CBC_SHA (0x002f).

d) Modify signature block in Server's Key Exchange

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
```

© 2019 Microsoft Corporation. All rights reserved.

```
                "regex":".*\\xde\\x2c\\x04\\x03.*"
            },
            "replacement":{
                "regex":"(\\xde\\x2c\\x04\\x03.{6})(.{2})",
                "replacement":"$1\\xee\\xee"
            }
        }
    ]
}
```

Where the evaluator is changing data inside Signature.

e) Modify server finished handshake

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\x4f\\x06\\x03.*"
            },
            "replacement":{
                "regex":"(\\x4f\\x06\\x03.{1})(.{1})",
                "replacement":"$1\\xee"
            }
        }
    ]
}
```

It will modify the first byte of server finished handshake.

f) Send garbled message after ChangeCipherSpec message

The evaluator has used the following configuration file:

```
{
    "debuglevel":1,
    "patterns":[
        {
            "match":{
                "bpf":"udp port 1812",
                "regex":".*\\x14\\x03\\x03\\x00\\x01\\x01.*"
            },
            "replacement":{
                "regex":"\\x14\\x03\\x03\\x00\\x01\\x01",
                "replacement":"\\x14\\x03\\x03\\x00\\x01\\xee"
            }
        }
    ]
}
```

© 2019 Microsoft Corporation. All rights reserved.

It changes the message by adding an extra byte at the end.

51.2.5.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

- 5.a Invalid TLS Version

No.	Time	Source	Destination	Protocol	Length	Info
140	408.085680797	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=194
142	408.094645087	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=194
143	408.085638578	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=194
144	408.095426643	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=194
145	408.103717207	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=195
146	408.103759816	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=195
147	408.108859057	10.10.20.40	10.10.10.1	RADIUS	1419	Access-Challenge id=195
148	408.107430935	10.10.20.40	10.10.20.25	RADIUS	1419	Access-Challenge id=195

```
▼ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.3 (0x0304)
    Random: 5bfd31312f39b25c4bd51ff07ba9e4de6aa70cef326ba9af...
    Session ID Length: 32
    Session ID: e0390000bd42631f7099f1069c4af2ecbf4b5b38f545a88e...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Compression Method: null (0)
    Extensions Length: 9
  ▶ Extension: extended_master_secret (len=0)
  ▶ Extension: renegotiation_info (len=1)
```

- 5.b Invalid Server Nonce

No.	Time	Source	Destination	Protocol	Length	Info
9	17.313240499	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=222
10	17.313299059	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=222
11	17.344705783	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=222
12	17.355962348	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=223
13	17.359500758	10.10.20.40	10.10.20.25	RADIUS	1418	Access-Challenge id=223
14	17.345728248	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=222
15	17.355918836	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=223
16	17.360893545	10.10.20.40	10.10.10.1	RADIUS	1418	Access-Challenge id=223
26	37.826837828	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=224

```
▼ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▼ Random: eefd3dc477b96b6cbb1cb02883f59833b87085c2377d0542...
      GMT Unix Time: Jan 21, 2097 03:30:28.000000000 CET
      Random Bytes: 77b96b6cbb1cb02883f59833b87085c2377d0542b7f2cc59...
    Session ID Length: 32
    Session ID: 4e0100009891892e9ff24a2e2e36b053a58e3c1b0f5eb35c...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Compression Method: null (0)
    Extensions Length: 9
  ▶ Extension: extended_master_secret (len=0)
  ▶ Extension: renegotiation_info (len=1)
```

- 5.c Modify Cipher Suite after Client Hello

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=236
2	0.008702732	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=236
3	0.018925916	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=237
4	0.022679268	10.10.20.40	10.10.20.25	RADIUS	1418	Access-Challenge id=237
6	-0.000040194	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=236
7	0.009596679	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=236
8	0.018885276	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=237
9	0.024173421	10.10.20.40	10.10.10.1	RADIUS	1418	Access-Challenge id=237

```
▼ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 81
    Version: TLS 1.2 (0x0303)
  ▼ Random: 5bfd3e4f28e222a5a1e04e97447624e9af39e6972e2998fe...
      GMT Unix Time: Nov 27, 2018 13:53:35.000000000 CET
      Random Bytes: 28e222a5a1e04e97447624e9af39e6972e2998fee6276a1c...
    Session ID Length: 32
    Session ID: 5521000050a5c50c90f9af8908f3618626e5c8ad69ee9fb3...
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Compression Method: null (0)
    Extensions Length: 9
  ▶ Extension: extended_master_secret (len=0)
  ▶ Extension: renegotiation_info (len=1)
```

- 5.d Modify Signature block in Server's Key Exchange

No.	Time	Source	Destination	Protocol	Length	Info
8	0.120634740	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=207
9	0.125607245	10.10.20.40	10.10.10.1	RADIUS	1417	Access-Challenge id=207
13	20.604000259	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=208
14	20.613027904	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=208

```
▼ Handshake Protocol: Server Key Exchange
    Handshake Type: Server Key Exchange (12)
    Length: 143
  ▼ EC Diffie-Hellman Server Params
      Curve Type: named_curve (0x03)
      Named Curve: secp256r1 (0x0017)
      Pubkey Length: 65
      Pubkey: 0476b7ffd8c53bb36659fb239e5674b461b05d1d22c415bf...
    ▶ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
      Signature Length: 70
      Signature: 30440220eeee0de910f11bbd16b86ed46b017b2bd1372cc7...
```

- 5.e Modify server finished handshake

No.	Time	Source	Destination	Protocol	Length	Info
31	20.555893781	10.10.20.25	10.10.20.40	RADIUS	419	Access-Request id=122
32	20.556744012	10.10.20.40	10.10.20.25	RADIUS	333	Access-Challenge id=122
33	20.583180011	10.10.20.25	10.10.20.40	RADIUS	364	Access-Request id=123
34	20.585162867	10.10.20.40	10.10.20.25	RADIUS	284	Access-Accept id=123
35	20.530747153	10.10.10.1	10.10.20.40	RADIUS	228	Access-Request id=121
36	20.540344304	10.10.20.40	10.10.10.1	RADIUS	132	Access-Challenge id=121
37	20.555877551	10.10.10.1	10.10.20.40	RADIUS	419	Access-Request id=122
38	20.557044361	10.10.20.40	10.10.10.1	RADIUS	333	Access-Challenge id=122
39	20.583161101	10.10.10.1	10.10.20.40	RADIUS	364	Access-Request id=123
40	20.585676371	10.10.20.40	10.10.10.1	RADIUS	284	Access-Accept id=123

```
▼ AVP: t=EAP-Message(79) l=6 Last Segment[1]
    Type: 79
    Length: 6
    EAP fragment: 03d7ee04
  ▼ Extensible Authentication Protocol
      Code: Success (3)
      Id: 215
      Length: 60932
0000  03 d7 ee 04                                        ....
```

- 5.f Send garbled message after ChangeCipherSpec message

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.010432020	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=105
5	0.037318559	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=106
6	0.040945790	10.10.20.40	10.10.20.25	RADIUS	1418	Access-Challenge id=106
8	0.070571739	10.10.20.25	10.10.20.40	RADIUS	112	Access-Request id=107
9	0.077229139	10.10.20.40	10.10.20.25	RADIUS	243	Access-Challenge id=107
10	0.037274917	10.10.10.1	10.10.20.40	RADIUS	387	Access-Request id=106
11	0.042082712	10.10.20.40	10.10.10.1	RADIUS	1418	Access-Challenge id=106
13	0.070537805	10.10.10.1	10.10.20.40	RADIUS	112	Access-Request id=107
14	0.077594024	10.10.20.40	10.10.10.1	RADIUS	243	Access-Challenge id=107
16	20.515081158	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=108

```
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message

0000  01 22 00 75 0d 80 00 00  00 6b 14 03 03 00 01 ee   ."u.... .k......
0010  16 03 03 00 60 d4 29 1b  b4 24 16 ac 64 fd cb 2d   ....`.). .$..d..-
0020  c6 3d a3 7b e3 2c 36 61  8a 9b bb 84 75 37 51 8b   .=.{.,6a ....u7Q.
0030  60 a5 49 06 bf de e2 8b  50 44 04 c4 9e 48 33 4d   `.I..... PD...H3M
0040  da d7 04 3f 84 bc 6d 63  cc fc 7f d1 cf 33 25 9a   ...?..mc .....3%.
0050  b7 9c 4a 74 fb 26 bc 42  35 73 fb 72 eb 71 93 4e   ..Jt.&.B 5s.r.q.N
0060  5f c6 51 0c f5 6b 6f fb  e2 46 4d a5 41 a5 bf 48   _.Q..ko. .FM.A..H
0070  d4 54 75 4f 4a                                     .TuOJ
```

51.2.5.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 5** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 5**.

51.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_TLSC_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

52 FCS_TLSC_EXT.2/WLAN

The assurance activity for the **FCS_TLSC_EXT.2/WLAN** requirement is stated as follows:

> The evaluator shall verify that the TSS describes the supported Elliptic Curves Extension and whether the required behavior is performed by default or may be configured.
>
> If the TSS indicates that the supported Elliptic Curves Extension must be configured to meet the requirement, the evaluator shall verify that the operational guidance includes instructions on configuration of the supported Elliptic Curves Extension.
>
> The evaluator shall perform the following test:
>
> **Test 1**: The evaluator shall configure a server to perform ECDHE key exchange using each of the TOE's supported curves and shall verify that the TOE successfully connects to the server.

52.1 Documentation Review activity

52.1.1 Findings

The evaluator has reviewed the section **6.3.2 Available EAP-TLS ciphersuites** of the ***Operational Guidance*** document. This section states that Windows can use different Elliptic Curves ciphersuites including among others those requested:

- secp256r1
- secp384r1
- secp521r1

Complete set of TLS cipher suites implemented can be found here: Cipher Suites in TLS/SSL (Schannel SSP)

These cipher suites match with the selected ones in the requirement definition. The Cipher suites listed above are a subset of the ones implemented in the Schannel library, as it can be checked in the previous URL.

52.1.2 Verdict

The evaluator considers that the TSS provides enough information related to what Elliptic Curves cipher suites are available.

The ***Operational Guidance*** document includes information regarding how to configure correctly the TOE for testing purposes including ciphersuite selection and associated restrictions.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

52.2 Test Activity

52.2.1 Test 1

The evaluator will attempt to force the client to use the following ciphersuites:

- secp256r1
- secp384r1
- secp521r1

52.2.1.1 Setup

In this case, the general NPS architecture composed by the following elements is used:

> - Windows NPS Server (EAP-TLS)
> - Windows Client using domain user (FIPS Enabled)
> - AP WPA2-Enterprise mode
> - W10WLANServerAutomator.ps1

52.2.1.2 Procedure

W10WLANServerAutomator.ps1 is a script that has several functions to perform different WLAN tests on the server side.

In this case, these are the functions that manage the FCS_TLSC_EXT.2/WLAN tests:

The main menu:

```
function Fcs-Tlsc-Ext-2 {
    Write-Host "============================"
    Write-Host "FCS_TLSC_EXT.2/WLAN - Test 1"
    Write-Host "============================"
    Write-Host "Please select curve to test:"
    Write-Host "1. secp256r1"
    Write-Host "2. secp384r1"
    Write-Host "3. secp521r1"

    $input = Read-Host "Please select"
    switch ($input) {
        '1' { setCurve256 }
        '2' { setCurve384 }
        '2' { setCurve521 }
        Default {}
    }

    ConnectTo $radiusNetwork
}
```

Each function to set the appropriate curve:

secp256r1 curve

© 2019 Microsoft Corporation. All rights reserved.

```
# FCS_TLSC_EXT.2/WLAN - Test 1 - secp256r1 curve
function setCurve256 {
    Write-Warning "[I]: Setting the configuration to test FCS_TLSC_EXT.1/WLAN -
        ECDSA 256"
    setCipherSuite "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the server ECDSA 256 certificate"
    AddServerCert
        ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_256_SERVER_CERT.pfx"
    Write-Warning "[I]: Importing the NPS configuration for ECDSA 256
        certificate"
    Import-NpsConfiguration -Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
  WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_ECDSA_256_CONFIG.xml"
}
```

secp384r1 curve

```
# FCS_TLSC_EXT.2/WLAN - Test 1 - secp384r1 curve
function setCurve384 {
    Write-Warning "[I]: Setting the configuration to test FCS_TLSC_EXT.1/WLAN -
        ECDSA 384"
    setCipherSuite "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"
    Write-Warning "[I]: Adding the CA"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the server ECDSA 384 certificate"
    AddServerCert
        ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_384_SERVER_CERT.pfx"
    Write-Warning "[I]: Importing the NPS configuration for ECDSA 384
        certificate"
    Import-NpsConfiguration -Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
  WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_ECDSA_384_CONFIG.xml"
}
```

secp521r1 curve

```
# FCS_TLSC_EXT.2/WLAN - Test 1 - secp521r1 curve
function setCurve521 {
    Write-Warning "[I]: Setting the configuration to test FCS_TLSC_EXT.1/WLAN -
        ECDSA 521"
    setCipherSuite "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"
    AddCA ".\WLAN\NPS_OK\NPS_OK_CA.pfx"
    Write-Warning "[I]: Adding the server ECDSA 521 certificate"
    AddServerCert
        ".\WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_OK_ECDSA_521_SERVER_CERT.pfx"
    Write-Warning "[I]: Importing the NPS configuration for ECDSA 521
        certificate"
    Import-NpsConfiguration -Path
        "C:\Users\Administrator\Desktop\certs\scripts_w10\
  WLAN\NPS_OK\WLAN_FCS_TLSC_EXT_2\NPS_ECDSA_521_CONFIG.xml"
```

© 2019 Microsoft Corporation. All rights reserved.

```
}
```

On the client side, an imported ECDSA certificate and a connection to Radius network are needed.

Observe the results.

52.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

Using the captured packets, the evaluator has tested all the ciphersuites.

The sample for secp256r1 can be seen below:

No.	Time	Source	Destination	Protocol	Length	Info
11	20.552781	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=225
12	20.562100	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=225
13	20.572197	10.10.20.25	10.10.20.40	RADIUS	387	Access-Request id=226
14	20.574871	10.10.20.40	10.10.20.25	RADIUS	1419	Access-Challenge id=226
16	20.600916	10.10.20.25	10.10.20.40	RADIUS	191	Access-Request id=227
18	23.601159	10.10.20.25	10.10.20.40	RADIUS	191	Access-Request id=227, Duplicate Request
20	29.601073	10.10.20.25	10.10.20.40	RADIUS	191	Access-Request id=227, Duplicate Request
21	32.184567	10.10.20.40	10.10.20.25	RADIUS	187	Access-Challenge id=227
22	32.198708	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=228
23	32.200360	10.10.20.40	10.10.20.25	RADIUS	284	Access-Accept id=228

```
    Session ID Length: 32
    Session ID: 39170000f8b6871c68fe9b1efa3a2afb33345eb5f694deba...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Compression Method: null (0)
    Extensions Length: 9
  > Extension: extended_master_secret (len=0)
  > Extension: renegotiation_info (len=1)
v Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1002
    Certificates Length: 999
  v Certificates (999 bytes)
      Certificate Length: 996
    v Certificate: 308203e0308202c8a003020102021377000000012157abbbb... (id-at-commonName=LAB-EE.lab.ee)
      v signedCertificate
          version: v3 (2)
          serialNumber: 0x7700000012157abbbbe6c2f895000000000012
        > signature (sha256WithRSAEncryption)
        > issuer: rdnSequence (0)
        > validity
        > subject: rdnSequence (0)
        v subjectPublicKeyInfo
          v algorithm (id-ecPublicKey)
              Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
            v ECParameters: namedCurve (0)
                namedCurve: 1.2.840.10045.3.1.7 (secp256r1)
```

The sample for secp384r1 can be seen below:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
15	16.791159	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=234
16	16.801945	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=234
17	16.818058	10.10.20.25	10.10.20.40	RADIUS	419	Access-Request id=235
18	16.831900	10.10.20.40	10.10.20.25	RADIUS	1478	Access-Challenge id=235
20	16.901250	10.10.20.25	10.10.20.40	RADIUS	191	Access-Request id=236
21	16.925666	10.10.20.40	10.10.20.25	RADIUS	187	Access-Challenge id=236
22	16.951529	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=237
23	16.952842	10.10.20.40	10.10.20.25	RADIUS	284	Access-Accept id=237

```
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
Compression Method: null (0)
Extensions Length: 9
> Extension: extended_master_secret (len=0)
> Extension: renegotiation_info (len=1)
v Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1031
    Certificates Length: 1028
  v Certificates (1028 bytes)
      Certificate Length: 1025
    v Certificate: 308203fd308202e5a003020102021377000000013a1fcf11f... (id-at-commonName=LAB-EE.lab.ee)
      v signedCertificate
          version: v3 (2)
          serialNumber: 0x7700000013a1fcf11f767cb258000000000013
        v signature (sha256WithRSAEncryption)
            Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
        > issuer: rdnSequence (0)
        > validity
        > subject: rdnSequence (0)
        v subjectPublicKeyInfo
          v algorithm (id-ecPublicKey)
              Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
            v ECParameters: namedCurve (0)
                namedCurve: 1.3.132.0.34 (secp384r1)
```

The sample for secp521r1 can be seen below:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
73	33.665454	10.10.20.25	10.10.20.40	RADIUS	228	Access-Request id=229
74	33.677071	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=229
75	33.722481	10.10.20.25	10.10.20.40	RADIUS	419	Access-Request id=230
77	33.738180	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=230
78	33.801374	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=231
79	33.801990	10.10.20.40	10.10.20.25	RADIUS	152	Access-Challenge id=231
81	33.858529	10.10.20.25	10.10.20.40	RADIUS	191	Access-Request id=232
82	33.889919	10.10.20.40	10.10.20.25	RADIUS	187	Access-Challenge id=232
83	33.925040	10.10.20.25	10.10.20.40	RADIUS	253	Access-Request id=233
84	33.926961	10.10.20.40	10.10.20.25	RADIUS	284	Access-Accept id=233

```
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
Compression Method: null (0)
Extensions Length: 9
> Extension: extended_master_secret (len=0)
> Extension: renegotiation_info (len=1)
v Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1069
    Certificates Length: 1066
  v Certificates (1066 bytes)
      Certificate Length: 1063
    v Certificate: 308204233082030ba003020102021377000000014023dcfb7... (id-at-commonName=LAB-EE.lab.ee)
      v signedCertificate
          version: v3 (2)
          serialNumber: 0x7700000014023dcfb7c6eda83d000000000014
        v signature (sha256WithRSAEncryption)
            Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
        > issuer: rdnSequence (0)
        > validity
        > subject: rdnSequence (0)
        v subjectPublicKeyInfo
          v algorithm (id-ecPublicKey)
              Algorithm Id: 1.2.840.10045.2.1 (id-ecPublicKey)
            v ECParameters: namedCurve (0)
                namedCurve: 1.3.132.0.35 (secp521r1)
```

In all the tests performed, the TOE connects successfully.

52.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

52.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FCS_TLSC_EXT.2/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

53 FIA_PAE_EXT.1/WLAN

The assurance activity for the **FIA_PAE_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall perform the following tests:
>
> - **Test 1:** The evaluator shall demonstrate that the TOE has no access to the test network. After successfully authenticating with an authentication server through a wireless access system, the evaluator shall demonstrate that the TOE does have access to the test network communication with the access point.
> - **Test 2:** The evaluator shall demonstrate that the TOE has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the TOE still being unable to access the test network.
> - **Test 3:** The evaluator shall demonstrate that the TOE has no access to the test network. The evaluator shall attempt to authenticate using an invalid authentication server certificate, such that the EAP-TLS negotiation fails. This should result in the TOE still being unable to access the test network.

53.1 Documentation Review activity

53.1.1 Findings

The related assurance activity does not define any action for this requirement.

53.1.2 Verdict

The related assurance activity does not define any action for this requirement. Therefore, the **PASS** verdict is assigned to the documentation review activity.

53.2 Test Activity

53.2.1 Test 1

The evaluator will attempt to verify that once connected, the TOE has access to the test network.

53.2.1.1 Setup

In this case, the evaluator has used the general NPS architecture with the following elements and roles:

© 2019 Microsoft Corporation. All rights reserved.

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Kali Linux (sniffing machine)

53.2.1.2 Procedure

- Check initial disconnected configuration
- Connect to Radius network
- Connect to IIS server in test network
- Observe the results

53.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

Initially, the TOE is disconnected as seen below:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has verified that TOE is disconnected by running "ipconfig /all" in a command shell:

```
Wireless LAN adapter Wi-Fi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Marvell AVASTAR Wireless-AC Network Controller
   Physical Address. . . . . . . . . : [redacted]
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
```

In addition to this, it is not possible to reach the internal test web server:

The evaluator has connected to Radius Network:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has verified that valid network address has been obtained:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Marvell AVASTAR Wireless-AC Network Controller
   Physical Address. . . . . . . . . : [redacted]
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f41a:2a5f:5016:96b6%15(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.10.100(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.1
   DHCPv6 IAID . . . . . . . . . . . : 263499307
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-23-7F-20-C3-D4-6E-0E-0D-FD-56
   DNS Servers . . . . . . . . . . . : 10.10.20.40
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

The evaluator has verified that internal test network web server is available when connected:

53.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

© 2019 Microsoft Corporation. All rights reserved.

53.2.2 Test 2

The evaluator will attempt to connect to the test network using a revoked certificate.

53.2.2.1 Setup

In this case, the evaluator has used the general NPS architecture with the following elements and roles:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Kali Linux (sniffing machine)

53.2.2.2 Procedure

The evaluator has revoked the user certificate.

In order to do so, connect to CA server is needed. Then, in Server Manager => Tools => Certification Authority, right click on "Issued Certificates" and revoke certificate:

A popup will appear, then "Yes":

© 2019 Microsoft Corporation. All rights reserved.

Publish CRL by right click on Revoked certificates => Publish

Force CRL update on client running the following command:

```
certutil -setreg chain\ChainCacheResyncFiletime @now
```

Restart the client, try to connect to Radius network and observe the results.

53.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

As seen previously, the evaluator has tried to connect to test network and has received an error:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

In addition to this, the evaluator has identified the error in captured traffic:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.40	RADIUS	222	Access-Request id=105
2	0.009631	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=105
3	0.020470	10.10.20.25	10.10.20.40	RADIUS	384	Access-Request id=106
5	0.022526	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=106
6	0.041589	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=107
7	0.042280	10.10.20.40	10.10.20.25	RADIUS	392	Access-Challenge id=107
9	0.081273	10.10.20.25	10.10.20.40	RADIUS	266	Access-Request id=108
10	0.082667	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=108
11	0.090649	10.10.20.25	10.10.20.40	RADIUS	861	Access-Request id=109
12	0.097917	10.10.20.40	10.10.20.25	RADIUS	143	Access-Challenge id=109
13	0.104192	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=110
14	0.104933	10.10.20.40	10.10.20.25	RADIUS	86	Access-Reject id=110

```
[Time from request: 0.007268000 seconds]
Attribute Value Pairs
  > AVP: t=Session-Timeout(27) l=6 val=30
  v AVP: t=EAP-Message(79) l=19 Last Segment[1]
      Type: 79
      Length: 19
      EAP fragment: 014c00110d8000000007150301000202 2c
    v Extensible Authentication Protocol
        Code: Request (1)
        Id: 76
        Length: 17
        Type: TLS EAP (EAP-TLS) (13)
      > EAP-TLS Flags: 0x80
        EAP-TLS Length: 7
      v Secure Sockets Layer
        v TLSv1.2 Record Layer: Alert (Level: Fatal, Description: Certificate Revoked)
            Content Type: Alert (21)
            Version: TLS 1.0 (0x0301)
            Length: 2
          v Alert Message
              Level: Fatal (2)
              Description: Certificate Revoked (44)
```

53.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

53.2.3 Test 3

The evaluator will attempt to connect to the test network using an invalid CA server certificate.

53.2.3.1 Setup

In this case, the evaluator has used the general NPS architecture with the following elements and roles:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)

© 2019 Microsoft Corporation. All rights reserved.

- AP WPA2-Enterprise mode
- Debian 9 (sniffing machine)
- Script to test the Test 3 (W10WLANClientAutomator.ps1)

53.2.3.2 Procedure

The evaluator has followed the next steps for trying to connect to test network with an invalid CA certificate:

- Execute W10WLANClientAutomator.ps1 as Administrator and select 3 in FIA_PAE_EXT.1/WLAN menu:

```
==================
FIA_PAE_EXT.1/WLAN
==================
Please select test:
1. Test 1
2. Test 2
3. Test 3
Please select: 3
WARNING: [I]: Removing good CA
WARNING: [I]: Importing fake CA


   PSParentPath: Microsoft.PowerShell.Security\Certificate::CurrentUser\Root

Thumbprint                                Subject
----------                                -------
2BE1A575347DF22C8C7EC5248170E76F4BFDEBE8  CN=lab-LAB-EE-CA, DC=lab, DC=ee
WARNING: [I]: Is the server ready?
Press Enter to continue...:
```

When the server was ready, the evaluator pressed enter button to connect to Radius network.

Observe the results.

53.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

As seen previously, the evaluator has tried to connect to the network and has received an error:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

In addition to this, the evaluator has identified the error in captured traffic:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.25	10.10.20.40	RADIUS	222	Access-Request id=111
2	0.010420	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=111
3	0.020477	10.10.20.25	10.10.20.40	RADIUS	384	Access-Request id=112
5	0.022384	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=112
6	0.038300	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=113
7	0.039539	10.10.20.40	10.10.20.25	RADIUS	392	Access-Challenge id=113
8	0.093198	10.10.20.25	10.10.20.40	RADIUS	261	Access-Request id=114
9	0.095036	10.10.20.40	10.10.20.25	RADIUS	86	Access-Reject id=114

```
> AVP: t=Called-Station-Id(30) l=26 val=60-38-E0-D7-5F-04:Radius
> AVP: t=NAS-Port-Type(61) l=6 val=Wireless-802.11(19)
> AVP: t=NAS-Port(5) l=6 val=0
> AVP: t=Calling-Station-Id(31) l=19 val=B4-AE-2B-2D-55-83
> AVP: t=Connect-Info(77) l=23 val=CONNECT 0Mbps 802.11b
> AVP: t=Acct-Session-Id(44) l=19 val=5B8E69E3-0000009D
> AVP: t=Framed-MTU(12) l=6 val=1400
v AVP: t=EAP-Message(79) l=19 Last Segment[1]
    Type: 79
    Length: 19
    EAP fragment: 022200110d8000000007150303000202 30
  v Extensible Authentication Protocol
      Code: Response (2)
      Id: 34
      Length: 17
      Type: TLS EAP (EAP-TLS) (13)
    > EAP-TLS Flags: 0x80
      EAP-TLS Length: 7
    v Secure Sockets Layer
      v TLSv1.2 Record Layer: Alert (Level: Fatal, Description: Unknown CA)
          Content Type: Alert (21)
          Version: TLS 1.2 (0x0303)
          Length: 2
        v Alert Message
            Level: Fatal (2)
            Description: Unknown CA (48)
```

53.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3**.

53.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FIA_PAE_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

54 FIA_X509_EXT.2/WLAN

The assurance activity for the **FIA_X509_EXT.2** requirement is stated as follows:

> The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.
>
> The evaluator shall check the administrative guidance to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions for configuring the operating environment so that the TOE can use the certificates.
>
> The evaluator shall perform the following test for each trusted channel:
>
> - **Test 1:** The evaluator shall demonstrate using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

54.1 Documentation Review activity

54.1.1 Findings

The evaluator has reviewed the section **6.4.1 X.509 Certificate Validation and Generation** of the ***Security Target*** document.

According to section **6.4.1 X.509 Certificate Validation and Generation** and following the ***Operational Guidance***, all windows components user a common system subcomponent for certificate validation a described in RFC 5280 including all applicable usage constraints such as Client and Server Authentication.

In addition to this, regarding certificate validation procedure, if certificate validation fails, or if TOE is not ables to check the validation status for a certificate, TOE will not establish a trusted network channel and will inform the user and seek their consent before establishing secure connection.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has reviewed the Windows Administrative Guide section **6.2.4 Certificate validation** and specifically section **6.2.4.2 Configuring certificate validation for EAP-TLS**.

Administrator can configure certificate validation as described in Extensible Authentication Protocol (EAP) Settings for network Access in section Smart Card or other certificate properties configuration items: Extensible Authentication Protocol (EAP) Settings for Network Access

54.1.2 Verdict

The evaluator considers that the TSS and AGD provides enough information related to which certificate ares used, determine if certificate validation is needed and the appropriate actions related to acceptance.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

54.2 Test Activity

54.2.1 Test 1

The evaluator will attempt to verify that when the TOE cannot verify the validity of the certificate, it will ask the user to explicitly accept it.

54.2.1.1 Setup

In this case, The evaluator has used the general NPS architecture with the following elements and roles:

> - Windows NPS Server (EAP-TLS)
> - Windows Client using domain user (FIPS Enabled)
> - AP WPA2-Enterprise mode
> - Kali Linux (sniffing machine)

54.2.1.2 Procedure

- Delete root CA certificate from TOE

> - Open Management Console => Certificates => Computer Certificates
> - Trusted Root Certification Authorities => Certificates => Select CA root certificate => Right button => Delete

© 2019 Microsoft Corporation. All rights reserved.

- Try to connect to Radius network
- Observe the results

54.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

When connecting, user acceptance is requested:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

54.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

54.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FIA_X509_EXT.2/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

55 FIA_X509_EXT.4/WLAN

The assurance activity for the **FIA_X509_EXT.4/WLAN** requirement is stated as follows:

> The evaluator shall examine the TSS to determine that it describes all certificate stores implemented that contain certificates used to meet the requirements of this EP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access.
>
> The evaluator shall perform the following tests for each function in the system that requires the use of certificates:
>
> - **Test 1:** The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.

55.1 Documentation Review activity

55.1.1 Findings

The evaluator has reviewed the section **6.4.2 Certificate Storage** of the ***Security Target***, where it is stated that each user has their own certificate store and there is a certificate store for computer account. Certificate store is managed by access control policy in Windows and only authorized administrator (like local administrator) can add or remove entries.

55.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the PASS verdict is assigned to the documentation review activity.

55.2 Test Activity

55.2.1 Test 1

In this case, user certificates are derived from CA root certificate. The evaluator will attempt to verify that when the TOE cannot verify the validity of the CA root certificate, it will ask the user to explicitly accept it.

55.2.1.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

The applicable setup for this test is the same as the one defined for *Test 1* in FIA_X509_EXT.2/WLAN.

55.2.1.2 Procedure

The evaluator shall carry out the same procedure as one defined for *Test 1* in FIA_X509_EXT.2/WLAN.

55.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms. Results can be seen in *Test 1* in FIA_X509_EXT.2/WLAN.

55.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

55.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FIA_X509_EXT.4/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

56 FMT_SMF_EXT.1/WLAN

The assurance activity for the **FMT_SMF_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall check to make sure that every management function mandated by the EP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.
>
> The evaluator shall test the TOE's ability to provide the management functions by configuring the TOE and testing each option listed in the requirement above.
>
> Note that the testing here may be accomplished in conjunction with the testing of other requirements, such as FCS_TLSC_EXT and FTA_WSE_EXT.

56.1 Documentation Review activity

56.1.1 Findings

The evaluator has reviewed the Windows ***Operational Guidance*** document sections **6.4 Managing network connections** and **6.5 Managing personal hotspots**.

The evaluator has verified that every management function required is included in Administrative Guide and it provides information about how to properly configure them.

56.1.2 Verdict

The evaluator considers that the ***Operational Guidance*** provides enough information related to managements functions included in the EP.

Therefore, the evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

56.2 Test Activity

56.2.1 Test 1

The evaluator will attempt to verify supported management functions required by the EP as described in the operational guidance.

56.2.1.1 Setup

© 2019 Microsoft Corporation. All rights reserved.

In order to perform tests related for this Assurance Activity, a full simulation environment for WPA2 Enterprise based on Microsoft Radius Server Implementation (Network Policy Server) should be deployed.

The scenario to perform the assurance activities according to the Protection Profile is composed of the following elements:

- Server Machine acting as Active Directory Domain Controller, Root Certificate Authority, Web Server and Network Policy and Access Services.
- Wi-Fi Client Machine (Windows 10 1809)
- MITM / Wi-Fi Sniffer with two physical network interfaces and a Wi-Fi interface.
- Linksys Access Point
- TLS Debian server

All devices are in different networks with the following configuration:

- Server Machine, IP = 10.10.20.40
- Client Machine, IP = 10.10.10.*
- Access Point, IP = 10.10.10.1
- TLS Debian server, IP = 10.10.20.30
- Sniffer Machine, IP-1 = 10.10.10.25 IP-2 = 10.10.20.25

Final configuration of full environment is as follows:

56.2.1.1.1 Server Machine Configuration

The server machine shall have at least the following roles enabled: Active Directory Domain Services, Active Directory Certificate Services and Network Policy and Access Services. These roles should be installed in correct order as described below.

Active Directory Domain Services and Active Directory Certificate Services

The setup can be found in FTP_ITC_EXT.1.1(TLS).

Network Policy and Access Services

© 2019 Microsoft Corporation. All rights reserved.

Network Policy and Access Services shall be installed and configured following the next steps:

- Open the Server Manager form the task bar.
- From the Server Manager Dashboard select "Manage" and then "Add Roles and Features".
- Click next until the "Select server roles" screen, then select "Network Policy and Access Services" and click "Next", accepting the suggested additional features.

- Click next until option Network Policy and Access Services is selected:

© 2019 Microsoft Corporation. All rights reserved.

- Click Next until Installation is completed and the Close
- Once installed, in Server Manager => Tools, you should click in "Network Policy Server"use . You should select option "RADIUS server for 802.1X Wireless or Wired Connections" and then "Configure 802.1X"

© 2019 Microsoft Corporation. All rights reserved.

- Select Secure Wireless connection and click Next
- In Specify 802.1X Switches, Click Add

© 2019 Microsoft Corporation. All rights reserved.

- Complete information related to the Access Point including
  - Name
  - IP (10.10.20.25)
  - Shared secret (this is the password that we will use later in the AP configuration)

© 2019 Microsoft Corporation. All rights reserved.

- In Specify 802.1X Switches, click Next
- In Configure an Authentication Method, select Microsoft: Smart Card or other certificate

© 2019 Microsoft Corporation. All rights reserved.

- Do not add any group in Specify User Groups and Click “Next”

© 2019 Microsoft Corporation. All rights reserved.

- In Configure traffic controls screen, click Next

© 2019 Microsoft Corporation. All rights reserved.

- And then click Finish

© 2019 Microsoft Corporation. All rights reserved.

For routing purposes, the following command shall be executed in an administrative shell:

```
route –p add 10.10.10.0 mask 255.255.255.0 10.10.20.25 metric 1
```

56.2.1.1.2 Access Point Configuration

Access point shall be configured in WPA2 Enterprise Mode according to data obtained from previous configuration steps including Server IP, connection type and shared secret.

- Network name: Radius
- Shared key: whatever (configured previously in NPS in server)
- Radius server: 10.10.20.40 (Server IP)
- Radius port: 1812

© 2019 Microsoft Corporation. All rights reserved.

- Channel: 8 (According to test procedures, you should choose a channel not occupied by other Wi-Fi networks)
- Enable Second Network with WPA2 Personal as shown below because it will be used during testing procedures as well.
- For routing traffic using the MITM machine, an static route shall be created

56.2.1.1.3 Man in the Middle Machine Configuration

The evaluator has configured a machine with the following configuration:

- 1 Physical Ethernet interfaces
- 1 Virtual network interface
- 1 Wi-Fi network interface (Supporting Monitor Mode)

Software:

- Debian 9
- RehtSe: https://github.com/JmFoces/Rehtse
- Aircrack-ng Suite: https://www.aircrack-ng.org/
- Wireshark: https://www.wireshark.org/

Initial script for network configuration:

```
ifconfig eth0 10.10.10.25 netmask 255.255.254.0
ifconfig vboxnet0 10.10.20.25 netmask 255.255.254.0
echo 1 > /proc/sys/net/ipv4/ip_forward
```

IP Tables configuration script for configuring MITM and allow RehtSe to modify packets can be seen below:

```
#!/bin/bash
set -e
sudo iptables -Z ; iptables -F ; iptables -t nat -F; iptables -t mangle -F
sudo iptables -A FORWARD -s 10.10.10.0/24 -p tcp -d 10.10.20.0/24 -j ACCEPT
sudo iptables -A FORWARD -s 10.10.10.0/24 -p udp -d 10.10.20.0/24 -j ACCEPT
sudo iptables -A FORWARD -s 10.10.20.0/24 -p udp -d 10.10.10.0/24 -j NFQUEUE
    --queue-num 0
sudo iptables -A FORWARD -s 10.10.20.0/24 -p tcp -d 10.10.10.0/24 -j NFQUEUE
    --queue-num 0
sudo iptables -t nat -A POSTROUTING -j MASQUERADE
```

56.2.1.1.4 Windows 10 Client Configuration

Firstly the client connected to LAN for joining the Domain and obtaining certificates will be configured. Secondly, the client shall be disconnected from the LAN and has to use the Wi-Fi to connect to the domain using Radius services.

These steps can be found in FTP_ITC_EXT.1.1(TLS).

© 2019 Microsoft Corporation. All rights reserved.

56.2.1.2 Procedure

- Review TOE network configuration.

56.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has followed instructions included in ***Operational Guidance*** to establish a connection using EAP-TLS in **6.4.8 Configuring a Wi-Fi connection profile with the Windows UI**:

- Open "Network and Sharing Center" and click on "Set up a new connection or network":

- Select "Manually connect to a wireless network":

© 2019 Microsoft Corporation. All rights reserved.

- Establish SSID name (Radius) and Security Type to WPA2-Enterprise:

© 2019 Microsoft Corporation. All rights reserved.

- Click on “Change connection settings”:

- The following popup windows will appear, click on “Security” Tab:

© 2019 Microsoft Corporation. All rights reserved.

- The following options appear in the Security Tab: > - Encryption Type > - Network Authentication Method
- For EAP-TLS, Microsoft: Smart Card or other certificate must be selected:

© 2019 Microsoft Corporation. All rights reserved.

- In Settings, it is possible to configure additional management functions including:
  - Use certificate or smart card
  - Verify server certificate

© 2019 Microsoft Corporation. All rights reserved.

- In Advance options, additional settings about certificates used in communication can be configured:

© 2019 Microsoft Corporation. All rights reserved.

Previous tests has been accomplished by the evaluator during the testing of other requirements in FCS_TLSC_EXT.1.1/WLAN.

- In Advance options of "Radius Wireless Network Properties", additional management functions can be configured:

  - PMK caching configuration
  - Enable/Disable IEEE 802.1X pre-authentication
  - FIPS compliance (mandatory)

© 2019 Microsoft Corporation. All rights reserved.

Additional management functions can be performed:

- Allow/Disallow connection by SSID:

```
netsh wlan add filter permission=block ssid="WiFi Network Name"
    networktype=infrastructure

netsh wlan add filter permission=allow ssid="WiFi Network Name"
    networktype=infrastructure
```

This test were accomplished during the testing performed in FTA_WSE_EXT.1/WLAN.

In addition to this, according to section **6.5 Managing personal hotspots** of the ***Operational Guidance***, configurations related to hotspot can be configured as well:

> - Settings > Network & Internet > Mobile Hotspot

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has verified that is not possible to share network when mobile hotspot is off.

56.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

56.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FMT_SMF_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

57 FPT_TST_EXT.1/WLAN

The assurance activity for the **FPT_TST_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.
>
> The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases.
>
> The evaluator shall ensure that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases.
>
> The evaluator shall perform the following tests:
>
> - **Test 1:** The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful.
> - **Test 2:** The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.

57.1 Documentation Review activity

57.1.1 Findings

The evaluator has reviewed the section **6.6.4 Windows Platform Integrity and Code Integrity** of the ***Security Target*** document, which describes the different stages of the boot process, providing information about which are the main files loaded in each step of the boot chain.

This information has allowed the evaluator to design a testing plan, which will be used during the test activity.

57.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the **PASS** verdict is assigned to the documentation review activity.

© 2019 Microsoft Corporation. All rights reserved.

57.2 Test Activity

57.2.1 Test 1

The evaluator will observe that the integrity mechanism does not flag any executable or internal library with integrity errors. Boot procedure has been described and tested in the *FPT_TST_EXT.1.1* requirement of the *General Purpose OS PP*.

57.2.1.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- The SignTool application must be available. This application comes within the Windows 10 1809 SDK.

57.2.1.2 Procedure

- Open a *cmd* terminal and go to the folder where *SignTool* is installed (typical path is: **%programfiles(x86)%\WindowsKits\10\bin\10.0.17763.0\x64**).
- Obtain the digital signature of the '*wlanapi.dll*' file, located in the System32 directory by typing the following command:

  ```
  signtool.exe verify /all /pa /v wlanapi.dll
  ```

  The signature will be shown in *Hash of file (sha256)* field.
- Reboot the TOE.
- Connect to Radius network.
- Observe the results.

57.2.1.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).
- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has obtained the same results for all the tested platforms.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has selected wlanapi.dll (*Microsoft Wireless LAN Api library*) and has verified that is correctly signed:

```
C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x64>signtool.exe verify /all /pa /v C:\Windows\System32\wlanapi.dll

Verifying: C:\Windows\System32\wlanapi.dll

Signature Index: 0 (Primary Signature)
Hash of file (sha256): 358B99861078BE27F50F0100723068F3F3E9AAEDDB098B958EF4704BB8245423

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Windows Production PCA 2011
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Mon Oct 19 19:51:42 2026
        SHA1 hash: 580A6F4CC4E4B669B9EBDC1B2B3E087B80D0678D

            Issued to: Microsoft Windows
            Issued by: Microsoft Windows Production PCA 2011
            Expires:   Sat Aug 11 21:23:35 2018
            SHA1 hash: 419E77AED546A1A6CF4DC23C1F977542FE289CF7

The signature is timestamped: Wed Apr 11 05:42:04 2018
Timestamp Verified by:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Time-Stamp PCA 2010
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Tue Jul 01 22:46:55 2025
        SHA1 hash: 2AA752FE64C49ABE82913C463529CF10FF2F04EE

            Issued to: Microsoft Time-Stamp Service
            Issued by: Microsoft Time-Stamp PCA 2010
            Expires:   Fri Sep 07 18:56:52 2018
            SHA1 hash: E4822E9D32366D7400320F1EC121B40DEB0C77D9

File has page hashes.


Successfully verified: C:\Windows\System32\wlanapi.dll

Number of signatures successfully Verified: 1
Number of warnings: 0
Number of errors: 0
```

The evaluator has connected to a wireless network and no integrity records have been created.

57.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

57.2.2 Test 2

The evaluator will modify the selected file and will try to obtain the evidence in the system log.

© 2019 Microsoft Corporation. All rights reserved.

57.2.2.1 Setup

Before the test execution, the following setup condition must be fulfilled:

- A hexadecimal editor (e.g. *HxD*) must be installed in the evaluated platforms to modify the binary files loaded during the boot process.
- A *WinPE* USB for both architectures must be available.

57.2.2.2 Procedure

- Create a copy of the system executable file '*wlanapi.dll*', which is going to be modified. This file is located in the *System32* directory.
- Open a *cmd* terminal and go to the folder where *SignTool* is installed (typical path is: **%programfiles(x86)%\WindowsKits\10\bin\10.0.17763.0\x64**).
- Obtain the digital signature of the '*wlanapi.dll*' file before the modification by typing the following command:

  ```
  signtool.exe verify /all /pa /v wlanapi.dll
  ```

  The signature will be shown in *Hash of file (sha256)* field.
- Modify any bytes of the file using a hexadecimal editor. (e.g. *HxD*).
- Obtain the digital signature of the '*wlanapi.dll*' file after the modification by typing the same command as before.
- Boot the TOE into the Windows Pre Environment using the *WinPE* USB and replace the original file with the modified one. Verify that there are no other copies of the file in the partition where the OS is installed (e.g. C:\Windows and C:\Windows\WinSxS). Otherwise, the OS will use them to repair itself.
- Restart the TOE.
- Connect to a wireless network.
- Observe the results.

57.2.2.3 Results

The evaluator has performed this test on the following platforms:

- Microsoft Surface Book 2 with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Microsoft Surface Laptop with Windows 10 Pro x64 October 2018 Update (v1809).
- Microsoft Surface Pro LTE with Windows 10 Enterprise x64 October 2018 Update (v1809).
- Dell Latitude 5290 with Windows 10 Pro x64 October 2018 Update (v1809).
- Dell Latitude 12 Rugged Tablet with Windows 10 Home x86 October 2018 Update (v1809).
- Samsung Galaxy Book 10.6" with Windows 10 Home x64 October 2018 Update (v1809).
- Samsung Galaxy Book 12" with Windows 10 Pro x64 October 2018 Update (v1809).
- HP Elite x2 1013 G3 Tablet with Windows 10 Enterprise x64 October 2018 Update (v1809).

© 2019 Microsoft Corporation. All rights reserved.

- HP EliteBook x360 1030 G2 with Windows 10 Pro x64 October 2018 Update (v1809).

The evaluator has obtained the same results for all the tested platforms.

The evaluator has selected wlanapi.dll (*Microsoft Wireless LAN Api library*) and has verified that initially is correctly signed:

```
C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x64>signtool.exe verify /all /pa /v C:\Windows\System32\wlanapi.dll

Verifying: C:\Windows\System32\wlanapi.dll

Signature Index: 0 (Primary Signature)
Hash of file (sha256): 358B99861078BE27F50F0100723068F3F3E9AAEDDB098B958EF4704BB8245423

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Windows Production PCA 2011
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Mon Oct 19 19:51:42 2026
        SHA1 hash: 580A6F4CC4E4B669B9EBDC1B2B3E087B80D0678D

            Issued to: Microsoft Windows
            Issued by: Microsoft Windows Production PCA 2011
            Expires:   Sat Aug 11 21:23:35 2018
            SHA1 hash: 419E77AED546A1A6CF4DC23C1F977542FE289CF7

The signature is timestamped: Wed Apr 11 05:42:04 2018
Timestamp Verified by:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Time-Stamp PCA 2010
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Tue Jul 01 22:46:55 2025
        SHA1 hash: 2AA752FE64C49ABE82913C463529CF10FF2F04EE

            Issued to: Microsoft Time-Stamp Service
            Issued by: Microsoft Time-Stamp PCA 2010
            Expires:   Fri Sep 07 18:56:52 2018
            SHA1 hash: E4822E9D32366D7400320F1EC121B40DEB0C77D9

File has page hashes.


Successfully verified: C:\Windows\System32\wlanapi.dll

Number of signatures successfully Verified: 1
Number of warnings: 0
Number of errors: 0
```

After that, the evaluator has modified the library using a hexadecimal editor:

© 2019 Microsoft Corporation. All rights reserved.

wlanapi.dll

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 F8 00 00 00  ............ø...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  45 26 45 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  E&Eprogram canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  52 5A F0 98 16 3B 9E CB 16 3B 9E CB 16 3B 9E CB  RZð˜.;žË.;žË.;žË
00000090  1F 43 0D CB 58 3B 9E CB 73 5D 9A CA 18 3B 9E CB  .C.ËX;žËs]šÊ.;žË
000000A0  73 5D 9D CA 15 3B 9E CB 73 5D 9B CA 01 3B 9E CB  s].Ê.;žËs]›Ê.;žË
000000B0  16 3B 9F CB EC 3F 9E CB 73 5D 9F CA 15 3B 9E CB  .;ŸËì?žËs]ŸÊ.;žË
000000C0  73 5D 9E CA 17 3B 9E CB 73 5D 93 CA 56 3B 9E CB  s]žÊ.;žËs]"ÊV;žË
000000D0  73 5D 63 CB 17 3B 9E CB 73 5D 61 CB 17 3B 9E CB  s]cË.;žËs]aË.;žË
000000E0  73 5D 9C CA 17 3B 9E CB 52 69 63 68 16 3B 9E CB  s]œÊ.;žËRich.;žË
000000F0  00 00 00 00 00 00 00 00 50 45 00 00 64 86 07 00  ........PE..d†..
00000100  9D 33 CE DE 00 00 00 00 00 00 00 00 F0 00 22 20  .3ÎÞ........ð."
```

And has verified that the hash has changed:

© 2019 Microsoft Corporation. All rights reserved.

```
C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x64>signtool.exe verify /all /pa /v E:\mod\wlanapi.dll

Verifying: E:\mod\wlanapi.dll

Signature Index: 0 (Primary Signature)
Hash of file (sha256): 6760017128E9D5E43AA77E1A882F8FFD4DCBA41B59745AE66C1B9BD65F5A8E58

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Windows Production PCA 2011
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Mon Oct 19 19:51:42 2026
        SHA1 hash: 580A6F4CC4E4B669B9EBDC1B2B3E087B80D0678D

            Issued to: Microsoft Windows
            Issued by: Microsoft Windows Production PCA 2011
            Expires:   Fri Jul 26 21:45:50 2019
            SHA1 hash: AE9C1AE54763822EEC42474983D8B635116C8452

The signature is timestamped: Sat Sep 15 02:30:07 2018
Timestamp Verified by:
    Issued to: Microsoft Root Certificate Authority 2010
    Issued by: Microsoft Root Certificate Authority 2010
    Expires:   Sat Jun 23 23:04:01 2035
    SHA1 hash: 3B1EFD3A66EA28B16697394703A72CA340A05BD5

        Issued to: Microsoft Time-Stamp PCA 2010
        Issued by: Microsoft Root Certificate Authority 2010
        Expires:   Tue Jul 01 22:46:55 2025
        SHA1 hash: 2AA752FE64C49ABE82913C463529CF10FF2F04EE

            Issued to: Microsoft Time-Stamp service
            Issued by: Microsoft Time-Stamp PCA 2010
            Expires:   Sat Nov 23 21:26:53 2019
            SHA1 hash: 5BA5294EAF18ACF8703C244F6A445BAD6FC7DB30

SignTool Error: WinVerifyTrust returned error: 0x80096010
        The digital signature of the object did not verify.

Number of signatures successfully Verified: 0
Number of warnings: 0
Number of errors: 1
```

The modified '*wlanapi.dll*' file was replaced using the WinPE environment and the system was rebooted.

Once the library was loaded (i.e. when trying to connect to Wi-Fi network), the following event was created:

© 2019 Microsoft Corporation. All rights reserved.

57.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

57.3 Final Verdict

Since all activities have been assigned a **PASS** verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FPT_TST_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

58 FTA_WSE_EXT.1/WLAN

The assurance activity for the **FTA_WSE_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall examine the TSS to determine that all of the attributes that can be used to specify acceptable networks (access points) are specifically defined.
>
> The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS.
>
> The evaluator shall ensure that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases.
>
> The evaluator shall also perform the following test for each attribute:
>
> - **Test 1:** The evaluator configures the TOE to allow a connection with a specific access point. The evaluator also configures the test environment such that the allowed access point and an access point that is not allowed are both "visible" to the TOE. The evaluator shall demonstrate that they can successfully establish a session with the allowed access point. The evaluator will then attempt to establish a session with the disallowed access point, and observe that the access attempt fails.
> - **Test 2:** The evaluator configures the TOE to allow a connection with a specific access point using EAP-TLS authentication (not only will the valid SSID be configured but the TOE will also be provided with certificates to complete the EAP-TLS authentication). The evaluator also configures the test environment such that an access point broadcasts the SSID the TOE has been configured to connect to but the authentication server does not have valid credentials.
>
> The evaluator will then attempt to establish a session with the valid SSID/invalid authentication server, and observe that the access attempt fails.

58.1 Documentation Review activity

58.1.1 Findings

The evaluator has reviewed the section **6.7 TOE Access** of the ***Security Target*** document, which describes that administrator can specify which Wi-Fi networks (SSID) a TOE may be connected to.

Also, the ***Operational Guidance*** document includes in its section **6.4 Managing network connections** instructions for configuring allowed Wi-Fi networks.

This information has allowed the evaluator to design a testing plan, which will be used during the test activity.

© 2019 Microsoft Corporation. All rights reserved.

58.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the PASS verdict is assigned to the documentation review activity.

58.2 Test Activity

58.2.1 Test 1

The evaluator will use two different Wi-Fi networks with different SSID: Radius and PSK and will block access to PSK. Then, the evaluator will check that it is not possible to connect to the PSK SSID.

58.2.1.1 Setup

In this case, the evaluator has used the general NPS architecture with the following elements and roles:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Kali Linux (sniffing machine)

58.2.1.2 Procedure

- Disallow PSK network
- Try to connect to PSK network
- Observe the results

58.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has executed the following command to connect to PSK network:

```
netsh wlan connect name=PSK
```

The evaluator has successfully connected to PSK network:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

Then, evaluator has blocked PSK SSID by running the following command:

```
netsh wlan add filter permission=block ssid=PSK networktype=infrastructure
```

At this point, network is not shown in available networks:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has tried to run the command again, but connection fails and a new event is created related to the connection error:

```
Message              : WLAN AutoConfig service failed to connect to a wireless network.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Connection Mode: Manual connection with a profile
                       Profile Name: Radius
                       SSID: Radius
                       BSS Type: Infrastructure
                       Failure Reason:The specific network is not available.
                       RSSI: -44

Id                   : 8002
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 24010
Opcode               : 191
Keywords             : -9223372036317903360
RecordId             : 1322
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 5952
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:19:12 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Failure
TaskDisplayName      : AcmConnection
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

58.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

© 2019 Microsoft Corporation. All rights reserved.

58.2.2 Test 2

The evaluator will use the original access point and will use a different authentication server created using the instructions for creating the original one.

58.2.2.1 Setup

In this case, the evaluator has used the general NPS architecture with the following elements and roles:

- Fake CA and fake server certificate in Windows NPS Server (EAP-TLS)
- Fake NPS configuration
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode
- Debian 9 (sniffing machine)

58.2.2.2 Procedure

- Import the fake CA, the fake server certificate and the fake NPS configuration
- Observe results

58.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has imported all the fake configuration:

© 2019 Microsoft Corporation. All rights reserved.

```
Administrator: Windows PowerShell
==================
FTA_WSE_EXT.1/WLAN
==================
Please select test:
1. Good NPS Server
2. Malicious NPS Server
Please select: 2
WARNING: [I]: Setting the malicious NPS Server
WARNING: [I]: Setting TLS_RSA_WITH_AES_128_CBC_SHA...
WARNING: [I]: Adding the malicious CA


   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\Root

Thumbprint                                Subject
----------                                -------
2BE1A575347DF22C8C7EC5248170E76F4BFDEBE8  CN=lab-LAB-EE-CA, DC=lab, DC=ee
WARNING: [I]: Adding the malicious server certificate


   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                                Subject
----------                                -------
5A1CFA35398371331FA0D716EA97C93DBAD2D343
WARNING: [I]: Importing malicious NPS configuration
```

The evaluator has tried to connect to Radius network and has obtained an error:

© 2019 Microsoft Corporation. All rights reserved.

© 2019 Microsoft Corporation. All rights reserved.

The evaluator has captured the traffic and the error is shown:

No.	Time	Source	Destination	Protocol	Length	Info
22	16.915714	10.10.20.25	10.10.20.40	RADIUS	222	Access-Request id=162
23	16.922487	10.10.20.40	10.10.20.25	RADIUS	132	Access-Challenge id=162
24	16.927758	10.10.20.25	10.10.20.40	RADIUS	384	Access-Request id=163
26	16.937935	10.10.20.40	10.10.20.25	RADIUS	52	Access-Challenge id=163
27	16.978880	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=164
28	16.980473	10.10.20.40	10.10.20.25	RADIUS	392	Access-Challenge id=164
29	17.029027	10.10.20.25	10.10.20.40	RADIUS	250	Access-Request id=165
30	17.030972	10.10.20.40	10.10.20.25	RADIUS	86	Access-Reject id=165

```
> Frame 30: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: PcsCompu_85:46:ae (08:00:27:85:46:ae), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
> Internet Protocol Version 4, Src: 10.10.20.40, Dst: 10.10.20.25
> User Datagram Protocol, Src Port: 1812, Dst Port: 40934
v RADIUS Protocol
    Code: Access-Reject (3)
    Packet identifier: 0xa5 (165)
    Length: 44
    Authenticator: 0f498a853179693a1a228c53416b04c9
    [This is a response to a request in frame 29]
    [Time from request: 0.001945000 seconds]
  v Attribute Value Pairs
    v AVP: t=EAP-Message(79) l=6 Last Segment[1]
        Type: 79
        Length: 6
        EAP fragment: 04280004
      v Extensible Authentication Protocol
          Code: Failure (4)
          Id: 40
          Length: 4
```

And an event error ID 12013 is created:

© 2019 Microsoft Corporation. All rights reserved.

```
Message              : Wireless 802.1x authentication failed.

                       Network Adapter: Marvell AVASTAR Wireless-AC Network Controller
                       Interface GUID: {b4f1d04a-32db-49e3-b033-795a571cabe0}
                       Local MAC Address: [redacted]
                       Network SSID: Radius
                       BSS Type: Infrastructure
                       Peer MAC Address: [redacted]
                       Identity: jdoe@lab.ee
                       User: jdoe
                       Domain: LAB
                       Reason: Explicit Eap failure received
                       Error: 0x80420400
                       EAP Reason: 0x80420400
                       EAP Root cause String: Windows cannot connect to this network\nThe root certificate on the
                       server required for authentication cannot be found in the certificate store. As a result, this
                       computer will not trust the network.

                       EAP Error: 0x80420400

Id                   : 12013
Version              : 0
Qualifiers           :
Level                : 2
Task                 : 24014
Opcode               : 205
Keywords             : -9223372036854774272
RecordId             : 1319
ProviderName         : Microsoft-Windows-WLAN-AutoConfig
ProviderId           : 9580d7dd-0379-4658-9870-d5be7d52d6de
LogName              : Microsoft-Windows-WLAN-AutoConfig/Operational
ProcessId            : 3116
ThreadId             : 8596
MachineName          : DESKTOP-JLDC4RV.lab.ee
UserId               : S-1-5-18
TimeCreated          : 11/26/2018 3:19:10 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : Microsoft-Windows-WLAN-AutoConfig/Operational
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Error
OpcodeDisplayName    : Failure
TaskDisplayName      : OneXAuthentication
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty...}
```

58.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

58.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FTA_WSE_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

59 FTP_ITC_EXT.1/WLAN

The assurance activity for the **FTP_ITC_EXT.1/WLAN** requirement is stated as follows:

> The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to an access point in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specification. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.
>
> The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to the access point, and that it contains recovery instructions should a connection be unintentionally broken.
>
> The evaluator shall perform the following tests:
>
> - **Test 1:** The evaluators shall ensure that the TOE is able to initiate communications with an access point using the protocols specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.
> - **Test 2:** The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
> - **Test 3:** The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.
> - **Test 4:** The evaluators shall physically interrupt the connection from the TOE to the access point (e.g., moving the TOE host out of range of the access point, turning the access point off). The evaluators shall ensure that subsequent communications are appropriately protected, at a minimum in the case of any attempts to automatically resume the connection or connect to a new access point.
>
> Further assurance activities are associated with the specific protocols.

59.1 Documentation Review activity

59.1.1 Findings

The evaluator has reviewed the sections **6.2.3.1 TLS, HTTPS, EAP-TLS, DTLS** and **6.2.3.2 Wireless Networking** of the ***Security Target*** document, which describes the TLS RFCs implemented in the TOE as well as the supported ciphersuites and Wi-Fi standards compliance.

The TSS identifies the following cipher suites used during the evaluated configuration:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

© 2019 Microsoft Corporation. All rights reserved.

- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

Full list of supported ciphersuites are described in the following document: Cipher Suites in TLS/SSL (Schannel SSP).

In addition to this, TSS describes how the TOE natively implements IEEE 802.11-2012 and IEEE 802.11ac-2013 for providing secure wireless local area networking (Wi-Fi). Additionally, in section **6.2.3.2 Wireless Networking**, it describes how the TOE can use PRF-384 in WPA2 Wi-Fi sessions and can generate AES 128-bit keys or PRF-704 to generate AES-256 bit keys both using the WIndows RBG. Key wrapping and unwrapping are performed according to the NIST SP 800-38F specification for securing GTK.

Moreover, the ***Operational Guidance*** document includes in its section **6.4.8 Configuring a Wi-Fi connection profile with the Windows UI** instructions about how to establish connection to an access point using EAP-TLS. In addition to this, section **6.4.7 Selecting a secure Wi-Fi connection with the Windows UI** states that when connection is unintentionally broken, TOE will automatically attempt to reconnect when it becomes available again.

This information has allowed the evaluator to design a testing plan, which will be used during the test activity.

59.1.2 Verdict

The evaluator considers that, the findings obtained during the documentation review activity demonstrate the fulfilment of the requirements established in the assurance activity section. Hence, the PASS verdict is assigned to the documentation review activity.

59.2 Test Activity

59.2.1 Test 1

The evaluator will try to successfully establish connection with AP using the protocols specified below.

59.2.1.1 Setup

The applicable setup for this test is the same as the one defined for *Test 1* in FCS_TLSC_EXT.1/WLAN.

59.2.1.2 Procedure

The evaluator shall carry out the same procedure as one defined for *Test 1* in FCS_TLSC_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

59.2.1.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms. This test was performed in conjunction with FCS_TLSC_EXT.1/WLAN. Results can be seen in *Test 1* in FCS_TLSC_EXT.1/WLAN

59.2.1.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 1** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 1**.

59.2.2 Test 2

The evaluator will review that no plaintext data is sent using authorized communication channels.

59.2.2.1 Setup

The applicable setup for this test is the same as the one defined for *Test 2* in FCS_CKM.1/WLAN.

59.2.2.2 Procedure

The evaluator shall carry out the same procedure as one defined for *Test 2* in FCS_CKM.1/WLAN.

59.2.2.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms. This test was performed in conjunction with FCS_CKM.1/WLAN. Results can be seen in *Test 2* in FCS_CKM.1/WLAN.

59.2.2.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 2** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 2**.

© 2019 Microsoft Corporation. All rights reserved.

59.2.3 Test 3

The evaluator will review that modification of the channel data is detected by the TOE.

59.2.3.1 Setup

The applicable setup for this test is the same as the one defined for for *Test 5* in FCS_TLSC_EXT.1/WLAN.

59.2.3.2 Procedure

The evaluator shall carry out the same procedure as one defined for for *Test 5* in FCS_TLSC_EXT.1/WLAN.

59.2.3.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms. This test was performed in conjunction with FCS_TLSC_EXT.1/WLAN. Results can be seen in for *Test 5* in FCS_TLSC_EXT.1/WLAN.

59.2.3.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 3** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 3**.

59.2.4 Test 4

The evaluator will physically disconnect the AP and will review that communications are appropriately protected.

59.2.4.1 Setup

In this case, we are going to use the general NPS architecture composed by the following elements:

- Windows NPS Server (EAP-TLS)
- Windows Client using domain user (FIPS Enabled)
- AP WPA2-Enterprise mode

© 2019 Microsoft Corporation. All rights reserved.

59.2.4.2 Procedure

- Connect to Radius Network
- After 1 minute, switch off AP
- Observe the results

59.2.4.3 Results

The evaluator has performed these tests on all the platforms listed in the ***Security Target*** document.

For Windows Server editions and Hyper-V versions, the wireless extended profile has not been tested because is out of the scope.

The evaluator has obtained the same results for all the tested platforms.

The evaluator has successfully connected to test network:

No.	Time	Source	Destination	Protocol	Length	Info
62	1.589852116	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	182	Association Response, SN=372, FN=0, Flags=....R...C
66	1.592407782	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	101	Request, Identity
73	1.614369860	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	102	Request, TLS EAP (EAP-TLS)
78	1.625986713	BelkinIn_d7:5f:04	Microsof_2d:55:83	TLSv1.2	271	Server Hello, Change Cipher Spec, Encrypted Handshake Message
84	1.638120111	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	100	Success
87	1.640014071	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	213	Key (Message 1 of 4)
92	1.650368954	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	247	Key (Message 3 of 4)
99	1.686691557	Microsof_2d:55:83	IPv6mcast_01:00:02	802.11	257	Data, SN=374, FN=0, Flags=.p....F.C
108	1.819011634	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=376, FN=0, Flags=.p....F.C
109	1.819650589	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=377, FN=0, Flags=.p....F.C
110	1.821232165	Microsof_2d:55:83	IPv6mcast_ff:16:96:…	802.11	170	Data, SN=378, FN=0, Flags=.p....F.C
111	1.822226824	Microsof_2d:55:83	IPv6mcast_02	802.11	154	Data, SN=379, FN=0, Flags=.p....F.C
112	1.826100240	Microsof_2d:55:83	IPv6mcast_16	802.11	182	Data, SN=380, FN=0, Flags=.p....F.C
122	2.325523877	Microsof_2d:55:83	IPv6mcast_16	802.11	182	Data, SN=386, FN=0, Flags=.p....F.C
128	2.696459701	Microsof_2d:55:83	IPv6mcast_01:00:02	802.11	257	Data, SN=391, FN=0, Flags=.p....F.C
136	2.818518012	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=393, FN=0, Flags=.p....F.C
137	2.819357578	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=394, FN=0, Flags=.p....F.C
138	2.820733429	Microsof_2d:55:83	IPv6mcast_01	802.11	178	Data, SN=395, FN=0, Flags=.p....F.C

```
▸ Frame 109: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
▸ Radiotap Header v0, Length 54
▸ 802.11 radio information
▸ IEEE 802.11 Data, Flags: .p....F.C
▾ Data (44 bytes)
    Data: ec7484f64d0501ff8555f1a92aded948409c029cf9a10f2b...
    [Length: 44]

0000  00 00 36 00 2f 40 40 a0  20 08 00 00 00 00 00 00   ··6·/@@· ····· ··
0010  43 64 81 5a 00 00 00 00  10 02 8f 09 a0 00 d6 00   Cd·Z···· ····· ··
0020  00 00 00 00 00 00 00 00  e5 64 81 5a 00 00 00 00   ····· ·· ·d·Z···
0030  16 00 11 03 d6 00 08 42  00 00 ff ff ff ff ff ff   ····· ·· ·B····· ··
0040  60 38 e0 d7 5f 04 b4 ae  2b 2d 55 83 90 17 0d 00   `8··_··· +-U····· ··
0050  00 60 00 00 00 00 ec 74  84 f6 4d 05 01 ff 85 55   ·`····· ·t··M····U
0060  f1 a9 2a de d9 48 40 9c  02 9c f9 a1 0f 2b ed 0d   ··*··H@· ····· +··
0070  02 b1 6f f0 a2 d1 f7 1a  40 97 06 7b ec 19 70 16   ··o····· @··{··p·
0080  d0 96 e0 9a 15 1e                                  ····· ··
```

Once switched off the AP, the TOE stops sending Data messages and tries to reach the Radius network sending Probe Request messages to the SSID:

© 2019 Microsoft Corporation. All rights reserved.

No.	Time	Source	Destination	Protocol	Length	Info
881	29.483208633	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=812, FN=0, Flags=.p....F.C
922	31.872595491	Microsof_2d:55:83	IPv4mcast_7f:ff:fa	802.11	271	Data, SN=841, FN=0, Flags=.p....F.C
932	32.756239893	Microsof_2d:55:83	IPv6mcast_01:00:02	802.11	257	Data, SN=850, FN=0, Flags=.p....F.C
959	35.402869174	Microsof_2d:55:83	BelkinIn_d7:5f:03	802.11	166	QoS Data, SN=190, FN=0, Flags=.p..R..TC
974	36.197153003	Microsof_2d:55:83	BelkinIn_d7:5f:03	802.11	183	QoS Data, SN=193, FN=0, Flags=.p..R..TC
975	37.102974260	Microsof_2d:55:83	Broadcast	802.11	137	Probe Request, SN=2949, FN=0, Flags=.....C, SSID=Radius
977	37.134538643	Microsof_2d:55:83	Broadcast	802.11	137	Probe Request, SN=2951, FN=0, Flags=.....C, SSID=Radius
992	39.589342048	Microsof_2d:55:83	Broadcast	802.11	137	Probe Request, SN=2987, FN=0, Flags=.....C, SSID=Radius
993	39.590346646	Microsof_2d:55:83	Broadcast	802.11	137	Probe Request, SN=2988, FN=0, Flags=.....C, SSID=Radius
999	42.554188869	Microsof_2d:55:83	Broadcast	802.11	131	Probe Request, SN=3017, FN=0, Flags=.....C, SSID=Wildcard (Broadcast)
1357	104.516102243	Microsof_2d:55:83	Broadcast	802.11	131	Probe Request, SN=3155, FN=0, Flags=.....C, SSID=Wildcard (Broadcast)
1358	104.518643926	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	339	Probe Response, SN=189, FN=0, Flags=.....C, BI=100, SSID=Radius
1359	104.519556675	Microsof_2d:55:83	Broadcast	802.11	131	Probe Request, SN=3156, FN=0, Flags=.....C, SSID=Wildcard (Broadcast)
1360	104.522479277	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	339	Probe Response, SN=189, FN=0, Flags=....R...C, BI=100, SSID=Radius
1362	104.525543317	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	339	Probe Response, SN=190, FN=0, Flags=.....C, BI=100, SSID=Radius
1365	104.598729129	Microsof_2d:55:83	Broadcast	802.11	131	Probe Request, SN=3157, FN=0, Flags=.....C, SSID=Wildcard (Broadcast)
1366	104.601249492	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	339	Probe Response, SN=192, FN=0, Flags=.....C, BI=100, SSID=Radius
1367	104.602093652	Microsof_2d:55:83	Broadcast	802.11	131	Probe Request, SN=3158, FN=0, Flags=.....C, SSID=Wildcard (Broadcast)
1368	104.604659093	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	339	Probe Response, SN=192, FN=0, Flags=....R...C, BI=100, SSID=Radius
1426	107.242017684	Microsof_2d:55:83	BelkinIn_d7:5f:04	802.11	88	Authentication, SN=3181, FN=0, Flags=.....C
1428	107.263901214	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	88	Authentication, SN=233, FN=0, Flags=.....C
1430	107.265723162	Microsof_2d:55:83	BelkinIn_d7:5f:04	802.11	179	Association Request, SN=3182, FN=0, Flags=.....C, SSID=Radius
1432	107.291927811	BelkinIn_d7:5f:04	Microsof_2d:55:83	802.11	182	Association Response, SN=234, FN=0, Flags=.....C
1435	107.303849296	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	101	Request, Identity
1440	107.323566354	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	101	Request, Identity
1498	110.363358380	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	102	Request, TLS EAP (EAP-TLS)
1512	110.376001637	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	102	Request, TLS EAP (EAP-TLS)
1518	110.382221001	BelkinIn_d7:5f:04	Microsof_2d:55:83	TLSv1.2	271	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message
1523	110.396657432	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAP	100	Success
1526	110.398544752	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	213	Key (Message 1 of 4)
1531	110.408621837	BelkinIn_d7:5f:04	Microsof_2d:55:83	EAPOL	247	Key (Message 3 of 4)
1542	110.452223430	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=273, FN=0, Flags=.p....F.C
1566	110.832160845	Microsof_2d:55:83	Broadcast	802.11	134	Data, SN=279, FN=0, Flags=.p....F.C

```
▸ Frame 1566: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
▸ Radiotap Header v0, Length 54
▸ 802.11 radio information
▸ IEEE 802.11 Data, Flags: .p....F.C
▾ Data (44 bytes)
    Data: ca75cec1327c9711cace0a93ae4d636a4e4eb4bec773b96f...
    [Length: 44]

0000  00 00 36 00 2f 40 40 a0  20 08 00 00 00 00 00 00   ··6·/@@· ····· ··
0010  9e ae 00 61 00 00 00 00  10 02 8f 09 a0 00 d6 00   ···a···· ····· ··
0020  00 00 00 00 00 00 00 00  40 af 00 61 00 00 00 00   ····· ·· @··a····
0030  16 00 11 03 d6 00 08 42  00 00 ff ff ff ff ff ff   ····· ·· B ····· ··
0040  60 38 e0 d7 5f 04 b4 ae  2b 2d 55 83 70 11 07 00   `8··_··· +-U·p···
0050  00 60 00 00 00 00 ca 75  ce c1 32 7c 97 11 ca ce   ·`····· ·u··2|····
0060  0a 93 ae 4d 63 6a 4e 4e  b4 be c7 73 b9 6f 54 cc   ···McjNN ···s·oT·
0070  7f 2d d3 5e 58 3f 7a 66  63 aa 5d 86 f2 3d cb 8f   ·-·^X?zf c·]··=··
0080  f6 16 cd a8 0e 2b                                  ····· +
```

After a period of time, the TOE identifies the network again and sends encrypted Data messages.

59.2.4.4 Verdict

The evaluator considers that, the results obtained from the test activity demonstrate the fulfilment of the **Test 4** requirements established in the assurance activity section.

Therefore, the **PASS** verdict is assigned to **Test 4**.

59.3 Final Verdict

Since all activities have been assigned a PASS verdict, the evaluator considers that the requirements established in the assurance activity are properly fulfilled. Therefore, the **PASS** verdict is assigned to FTP_ITC_EXT.1/WLAN.

© 2019 Microsoft Corporation. All rights reserved.

60 Evaluation Tools

- BinSkim, a verification tool which allows analyzing binaries to ensure that they have been built in compliance with Microsoft's Security Development Lifecycle (SDL).
- **CAVP**. FIPS tool used to test the cryptographic algorithm included in the TOE.
- **ee-tls-tool**. Test suite for TLS and X509 Common Criteria evaluations. It acts as an OpenSSL TLS server configured according to each test. A modified version based on tls-cc-tools has been used.
- **HxD**. Freeware Hex Editor and Disk Editor.
- **MITM Tool**. This software is a set of tools based on: br_netfiler + Iptables + NFQueue + BPF + Regex that allows modifying network packets on the fly. It is used to work as a MITM to modify the content of the network packages. More info at: Rehtse
- **PCPTool**. An utility which allows using TPM-related functionality.
- **SignTool**. A command line tool that provides the ability to sign files and verify signatures in files. It is distributed with the Windows 10 Software Development Kit (SDK).
- **Visual Studio 2017**. An integrated development environment from Microsoft.
- **VMMap**. VMMap is a process virtual and physical memory analysis utility.
- **WebClient** and **WebServer**. Internal tools that work as DTLS server and client.
- **Windows 10 Assessment and Deployment Kit (ADK)**. To create a WinPE USB.
- **Wireshark**. A network protocol analyser for macOS, Unix and Windows.

© 2019 Microsoft Corporation. All rights reserved.

61 Acronyms

- **AGD**. Administrative Guidance Document
- **API**. Application Programming Interface
- **CAVP**. Cryptographic Algorithm Validation Program
- **CC**. Common Criteria
- **CVE**. Common Vulnerabilities and Exposures
- **FIPS**. Federal Information Processing Standard
- **HTTPS**. HyperText Transfer Protocol Secure
- **HW**. Hardware
- **IKE**. Internet Key Exchange
- **IPSec**. Internet Protocol Security
- **ISAKMP**. Internet Security Association and Key Management Protocol
- **IT**. Information Technology
- **MITM**. Man In The Middle
- **OCSP**. Online Certificate Status Protocol
- **OS**. Operative System
- **PP**. Protection Profile
- **SFR**. Security Functional Requirement
- **SHA**. Secure Hash Algorithm
- **SSH**. Secure Shell
- **ST**. Security Target
- **SW**. Software
- **SSL**. Secure Socket Layer
- **TLS**. Transport Layer Security
- **TOE**. Target of Evaluation
- **TPM**. Trusted Platform Module
- **TSF**. TOE Security Functions
- **TSFI**. TOE Security Function Interface
- **TSS**. TOE Summary Specification
- **USB**. Universal Serial Bus
- **VPN**. Virtual Private Network

© 2019 Microsoft Corporation. All rights reserved.

62 Glossary

62.1 Common Criteria Terms

- **Common Criteria (CC)**. Common Criteria for Information Technology Security Evaluation.
- **Common Evaluation Methodology (CEM)**. Common Evaluation Methodology for Information Technology Security Evaluation.
- **Protection Profile (PP)**. An implementation-independent set of security requirements for a category of products.
- **Security Target (ST)**. A set of implementation-dependent security requirements for a specific product.
- **Target of Evaluation (TOE)**. The product under evaluation. In this case, the Operating System as described in section and its supporting documentation.
- **TOE Security Functionality (TSF)**. The security functionality of the product under evaluation.
- **TOE Summary Specification (TSS)**. A description of how a TOE satisfies the SFRs in a ST
- **Security Functional Requirement (SFR)**. A requirement for security enforcement by the TOE.
- **Security Assurance Requirement (SAR)**. A requirement to assure the security of the TOE
- **Extended Package (EP)**. An implementation-independent set of security requirements for a category of products, which extends those in a Protection Profile.

62.2 Technology Terms

- **Address Space Layout Randomization (ASLR)**. An anti-exploitation feature, which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.
- **Administrator**. An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system, which cannot be overridden by non-administrator users.
- **Application (app)**. Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation.
- **Application Programming Interface (API)**. A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.
- **Credential**. Data that establishes the identity of a user, e.g. a cryptographic key or password.

© 2019 Microsoft Corporation. All rights reserved.

- **Critical Security Parameters (CSP)**. Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module.
- **Data At Rest (DAR) Protection**. Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping.
- **Data Execution Prevention (DEP)**. An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.
- **Developer**. An entity that writes OS software. For the purposes of this document, vendors and developers are the same.
- **Host-based Firewall**. A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS.
- **Operating System (OS)**. Software that manages physical and logical resources and provides services for applications. The terms *TOE* and *OS* are interchangeable in this document.
- **Personally Identifiable Information (PII)**. Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.
- **Sensitive Data**. Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author.
- **User**. A user is subject to configuration policies applied to the operating system by administrators. On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.
- **Secure Shell (SSH)**. Cryptographic network protocol for initiating text-based shell sessions on remote systems.