MS-W10-I-003

Microsoft Windows 10 & Microsoft Windows Server 2012 R2 Operating System

18-03-2016

Version 1.3

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
FAX +34 916238772

Printed copy not supported.
Document approved by the E&E technical manager: José Emilio Rico Martínez

Version History

Version	Date	Summary of changes
1.0	December 28 , 2015	Initial version
1.1	December 31, 2015	Updates according new TSS content in [ST007]
1.2	January 29, 2016	Updates according new evidences
1.3	March 18, 2016	Algorithms implementation

Table of Contents

1. Background

Product	Windows Operating Systems (OS): • Microsoft Windows 10 Home Edition (32-bit and 64-bit versions) • Microsoft Windows 10 Pro Edition (32-bit and 64-bit versions) • Microsoft Windows 10 Enterprise Edition (32-bit and 64-bit versions) • Microsoft Windows Server 2012 R2 Standard Edition • Microsoft Windows Server 2012 R2 Datacenter Edition TOE Build: • Windows 10: build 10.0.10240 • Windows Server 2012 R2: build 6.3.9600
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 R4
CEM Version	CEM v 3.1 R4
Protection Profile	NIAP - Protection Profile for General Purpose Operating Systems, Version: 4.1, 20160309 and application notes as defined in [MS-WS10-I-000].
Evaluation Level	In conformance with [GPOSPP41]
Security Target	Microsoft Windows 10 Security Target version 1.0, March 18, 2016

2. Overview

The current document contains the obtained results by E&E laboratory after performing a Common Criteria evaluation taking into account the security requirements defined in the 'General-Purpose Operating System' Protection Profile ([GPOSPP41]) and the information provided in the security target (Microsoft Windows 10 Security Target version 1.0), for the following operating systems:

- Microsoft Windows 10 Home Edition (32-bit and 64-bit versions)
- Microsoft Windows 10 Pro Edition (32-bit and 64-bit versions)
- Microsoft Windows 10 Enterprise Edition (32-bit and 64-bit versions)
- Microsoft Windows Server 2012 R2 Standard Edition
- Microsoft Windows Server 2012 R2 Datacenter Edition

The build tested for all Windows 10 versions has been [version 10.0.10240] and for all Windows Server 2012 R2 versions [version 6.3.9600]. Before starting the testing, all critical updates as of October 31, 2015 were applied for all operating systems.

The hardware platforms used during the evaluation are listed below:

- Microsoft Surface Pro 3
- Microsoft Surface 3
- Windows Server 2012 R2 Hyper-V
- HP Pro x612 Notebook PC
- Dell Optiplex 755
- Microsoft Surface Book

The next table summarizes the combination between hardware platforms and operating system versions used for the testing:

Evaluated Testing Platforms
Dell Optiplex 755 with Windows 10 x86 Pro Edition
Dell Optiplex 755 with Windows 10 x86 Enterprise Edition
Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
HP Pro x2 612 with Windows 10 x64 Pro Edition
Surface 3 with Windows 10 x64 Enterprise Edition
Surface 3 Pro with Windows 10 x64 Enterprise Edition
Surface Book with Windows 10 x64 Enterprise Edition
Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The EE evaluation team has carried out a deep study about the operating system versions and the differences between them, concluding that not all the security requirements need to be

tested in all OS versions, given that the functionality is exactly the same. All test cases have been performed at least in Windows 10 x86, Windows 10 x64 and Windows Server 2012 R2. Each of the following test cases specifies the exactly testing platforms used.

The following information is included for each SFR:

1. Description of the assurance activity
2. Documentation review activity for both TSS and guidance including the corresponding verdict if applicable.
3. Testing activity. For each test case, the setup, the procedure for its execution, the obtained results (the expected results are specified in the assurance activity description) and verdict are included.

2.1. Executive summary

The following table summarizes the results obtained for each security requirement defined in the security target (Microsoft Windows 10 Security Target version 1.0, March 18, 2016), taking into account the information provided in the Operational Guidance (*Windows 10 and Server 2012 R2 GP OS Operational Guidance version 0.09, January 13, 2016)*.

Security Functional Requirement			
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 (lab testing for SP 800-56B)
FCS_CKM_EXT.3.1	PASS	PASS	PASS
FCS_COP.1 .1(SYM)	PASS	PASS	Covered by FIPS certification
			PASS (lab testing for AES-KW in WS)
FCS_COP.1 .1(HASH)	PASS	PASS	Covered by FIPS certification
FCS_COP.1 .1(SIGN)	PASS	PASS	Covered by FIPS certification
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_TLS_EXT.1.1	PASS	PASS	PASS
FCS_TLS_EXT.1.2	PASS	PASS	PASS
FCS_TLS_EXT.1.3	PASS	PASS	PASS
FCS_TLS_EXT.2.1	PASS	PASS	PASS
FCS_TLS_EXT.3.1	PASS	PASS	PASS
FCS_TLSC_EXT.4.1	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
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	PASS	PASS	PASS
FTP_TRP.1	PASS	PASS	PASS

3. FAU_GEN.1

3.1. Assurance activity

The assurance activity for FAU_GEN.1.1 requirement states as follows:

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type selected in the ST is included.

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. This should include all instance types of an event specified. 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.

The assurance activity for FAU_GEN.1.2 requirement states as follows:

The evaluator shall check the administrative guide and ensure that it 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 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 shall 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

3.2. Documentation review activity

3.2.1. Findings

The security target defines in its section **5.1.1.1 Audit Data Generation (FAU_GEN.1)** the following auditable events:

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),***
- ***Program initiations (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].***
-]

The evaluator has reviewed the operational guidance, which includes in its section **3.1 Audit Events** a table with all the auditable events. The content of this table matches with the selection performed by the vendor in the security target. For example, the following image shows the auditable events defined in the operational guidance and related to **User and Group management events.**

User and Group management events (Successful and unsuccessful add, delete, modify, suspend, lock)	Windows Logs/Security: add user: 4720 add user to group: 4732 delete user: 4726 delete user from group: 4733 add group: 4731 delete group: 4734 modify group: 4735 modify user account: 4738 disable user: 4725

The security target 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.

Additionally, the operational guidance 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 4732 (add user to group).

4732	Windows Logs/Security Subcategory: User Account Management	A member was added to a security-enabled group.	Logged: <Date and time of event> Member SID: <SID of user account> Group SID: <SID of group> Account Name: <name of user account> Group Name: <Name of group> Group SID Keywords: <Outcome as Success or Failure>

3.2.2. Verdict

The evaluator has reviewed the operational guidance and has ensured that every auditable event type selected in the security target is included in the operational guidance. Additionally, the format of every auditable event is also defined, including at least the fields defined in the security target (date and time, type of the event, subject identity and outcome).

Based on that, the evaluator considers that the above evidences 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.

3.3. Test Activity

Since there are several auditable events, the evaluator has split this test activity into different subtests in order to ensure appropriate coverage. The evaluator has also developed a script for some of these subtests.

The auditable events defined in the security target are the following, including the *eventID* defined in the administrator guide:

1. Start-up and shutdown of the audit functions (*eventIDs:* 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 (3038, 3077, 8020, 8022)
10. System reboot, restart and shutdown events (1100, 4608)
11. Kernel module loading and unloading events (3038, 3004)
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 subtests and the auditable events:

Scripts\Events	1	2	3	4	5	6	7	8	9	10	11	12	13
Test 1 - Startup, shutdown and logon events	X	X								X		X	
Test 2 - Privileges or role escalation and audit and log data access events				X			X						
Test 3 - User and group management events						X							
Test 4 - File and object events			X		X								
Test 5 - Cryptographic verification events								X					
Test 6 - Program initiations events									X				
Test 7- Kernel module loading events									X		X		
Test 8 - Lock and unlock account													X

Following sections contain the setup conditions, a procedure and the results obtained and verdict for each subtest defined above.

3.3.1. Test 1 - Startup, shutdown and logon events

3.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

3.3.1.2. Procedure

The evaluator has developed a script in order to make the subtest execution easier. The script shows the audit log related to the start-up and shutdown of the audit functions, system restart and shutdown events and success and failure authentication events.

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

1. Run a PowerShell terminal as administrator and type the following commands to enable the audit functions related to the system events and the logon events:

   "*auditpol /set /category:"System" /success:enable /failure:enable"*
   "auditpol /set /subcategory:"Logon" /success:enable /failure:enable"

2. Clear the previous event log by typing the following command: "*wevtutil cl Security*"

3. Restart the computer. After that, try to logon as an administrator using an invalid password.

4. Try to logon again, but this time using the correct password.

5. Run as administrator the script *FAU_GEN.1 - Startup&Shutdown&Logon.ps1*. To do this, type the following command in a PowerShell terminal: "*.\FAU_GEN.1 - Startup&Shutdown&Logon.ps1*".

6. Observe the results.

3.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

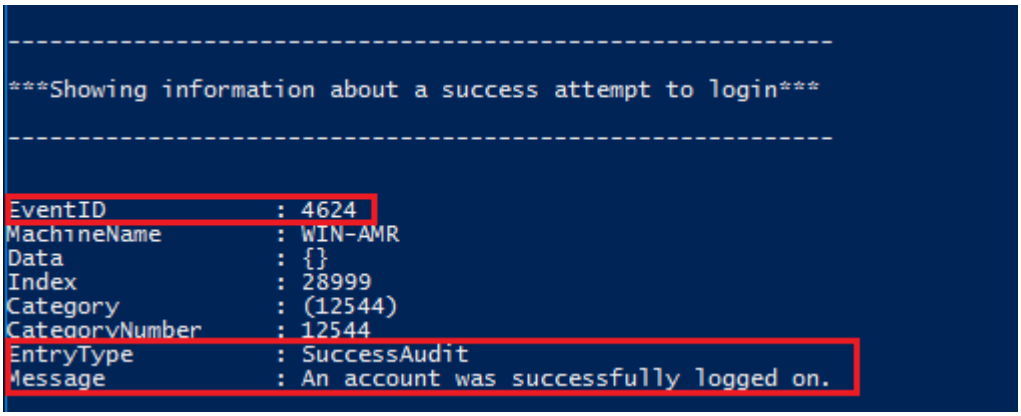
The evaluator has obtained the same results for all the tested platforms during the test execution. The following screenshots show that the audit events related to the startup and shutdown of the audit functions and successful and failed authentication events has been generated properly. Additionally, the generated audit log entries include the fields defined in the security target.

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

```
***Showing information about shutdown event***

------------------------------------------------------------


EventID             : 1100
MachineName         : WIN-AMR
Data                : {}
Index               : 28937
Category            : (103)
CategoryNumber      : 103
EntryType           : SuccessAudit
Message             : The event logging service has shut down.
Source              : Microsoft-Windows-Eventlog
ReplacementStrings  : {}
InstanceId          : 1100
TimeGenerated       : 10/28/2015 10:27:06 AM
TimeWritten         : 10/28/2015 10:27:06 AM
UserName            :
Site                :
Container           :



------------------------------------------------------------

***Showing information about startup event***

------------------------------------------------------------


EventID             : 4608
MachineName         : WIN-AMR
Data                : {}
Index               : 28948
Category            : (12288)
CategoryNumber      : 12288
EntryType           : SuccessAudit
Message             : Windows is starting up.

                      This event is logged when LSASS.EXE starts and the auditing subsystem is initialized.
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {}
InstanceId          : 4608
TimeGenerated       : 10/28/2015 10:27:37 AM
TimeWritten         : 10/28/2015 10:27:37 AM
UserName            :
Site                :
Container           :
```

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

```
------------------------------------------------------------

***Showing information about a fail attempt to login***

------------------------------------------------------------


EventID             : 4625
MachineName         : WIN-AMR
Data                : {}
Index               : 28987
Category            : (12544)
CategoryNumber      : 12544
EntryType           : FailureAudit
Message             : An account failed to log on.
```



3.3.1.4. Verdict

As the result above stated, the audit events related to the startup and shutdown of the audit functions and successful and failed authentication events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 1 - Startup, shutdown and logon events**.

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

3.3.2.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- User account with user name *user1* shall exist. This user shall belong to the default *Users* group. The password for this account must be *p@ss1234.*
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

3.3.2.2. Procedure

The evaluator has developed a script in order to make the subtest execution easier. The behavior of this script is as follows: first of all, the audit function related to the privileges use events is enabled. After that, two attempts to access the audit data are carried out, the first one using a user who does not have the administrator permissions and the second one using a user who has the administrator permissions.

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

1. Run as administrator the script *FAU_GEN.1 - AuditData&Priveleges.ps1*. To do this, type the following command in a PowerShell terminal: ".*FAU_GEN.1 - AuditData&Priveleges.ps1*".

2. Observe the results.

3.3.2.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

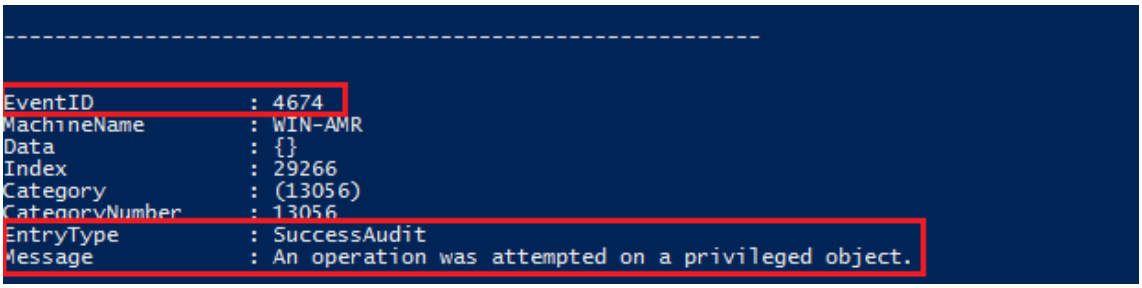
The evaluator has obtained the same result for all the tested platforms. During the test execution, an attempt to access audit data is performed using a user who does not have the administrator permissions. The result of this action is as follows:

```
WARNING: Requested registry access is not allowed.
Press Enter to continue...:
```

After that, the attempted is repeated but this time using a user with administrator rights.

Finally, the audit log entries show both the success and the failure attempt to access audit data. Additionally, the generated audit log entries include the fields defined in the security target:

```
------------------------------------------------------------


EventID            : 4673
MachineName        : WIN-AMR
Data               : {}
Index              : 29195
Category           : (13056)
CategoryNumber     : 13056
EntryType          : FailureAudit
Message            : A privileged service was called.
```



3.3.2.4. Verdict

As the result above stated, the audit events related to the privileges events and access data events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 2 - Privileges or role escalation and audit and log data events**.

3.3.3. Test 3 - User and group management events

3.3.3.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- User account with user name *userTest* shall not exist.
- Local group with name *groupTest* shall not exist.
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

3.3.3.2. Procedure

The evaluator has developed a script in order to make easier this subtest execution. The behavior of this script is as follows: first of all, the audit functions related to the account management events are enabled. After that the following operations are carried out: create a new local account (with name *userTest*), modify the user account description, disable the user account, enable the user account, create a new local group (with name *groupTest*), modify the group description, add the user created above to the group, remove the user from the group, delete the group *groupTest* and finally delete user *userTest*.

The following steps must be performed in order to check that the TOE audits these events.

1. Run as administrator the script *FAU_GEN.1 - Users&Groups.ps1*. To do this, type the following command in a PowerShell terminal: ".*FAU_GEN.1 - Users&Groups.ps1*".

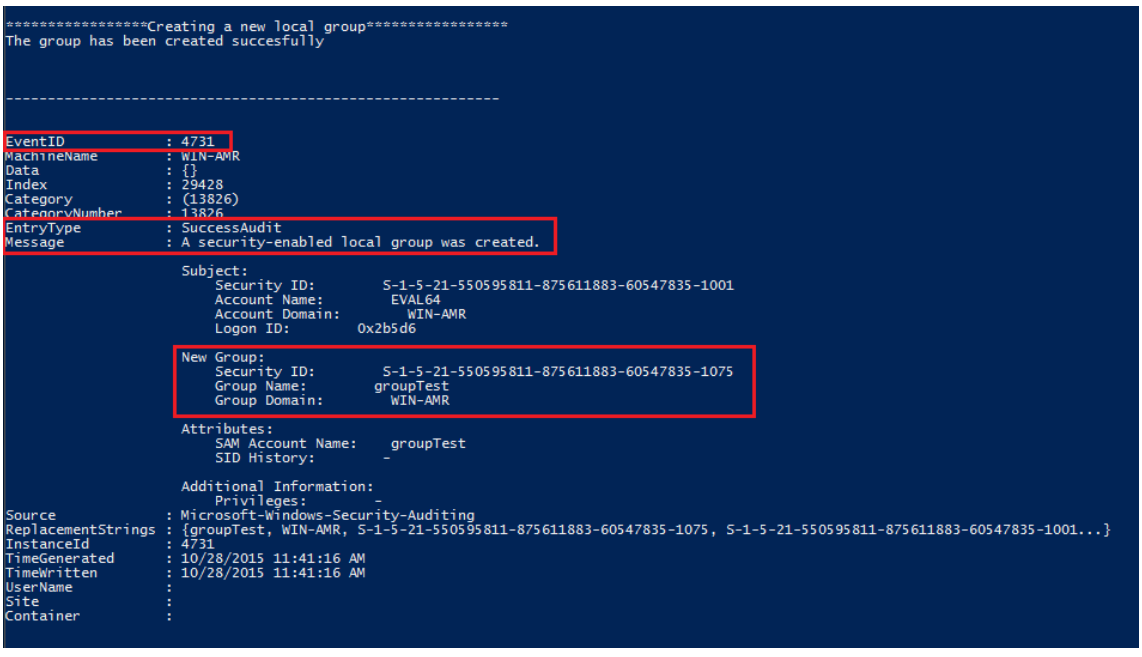
2. Observe the results.

3.3.3.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same result for all the tested platforms. During the test execution the following operations have been performed: add user, add group, add user to group, delete user, delete group, delete user from group, modify user, modify group, enable user and disable user. All of these operations have generated their related audit events. Since there are several audit log entries, the following screenshot shows an example which is related to local group creation operation.



As it can be observed, the audit log entry includes, among the other information, the required fields defined in the security target (Date and time of the event, type of the event, subject identity and outcome).

3.3.3.4. Verdict

As the result above stated, the audit events related to the user and group management events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 3 - User and group management events**.

3.3.4. Test 4 - File and object events

3.3.4.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- The script and the rest of the used files during the test execution (e.g. *common* folder which contains a script to get the audit logs) shall be stored in the root folder.
- User account with user name *user1* shall exist. This user shall belong to the default *Users* group. The password for this account must be *p@ss1234.*
- 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. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

3.3.4.2. Procedure

The evaluator has developed a script in order to make easier this subtest execution. The behavior of this script is as follows: first of all, a new empty file is created in a temporary folder (*C:\TEMP\file.txt*) and then, 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, and an audit rule is added to the created file in order 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 attempts to access the file are performed, the first one using a user who does not belong to the *Administrators* group and the second one using a user who belongs to the *Administrators* group.

The following steps must be performed in order to check that the TOE audits these events.

1. Run as administrator the script *FAU_GEN.1 - Fileobject&AdminRoot.ps1*. To do this, type the following command in a PowerShell terminal: ".*FAU_GEN.1 - Fileobject&AdminRoot.ps1*".

2. Observe the results.

3.3.4.3. Results

The evaluator has performed this test on the following evaluated platforms:

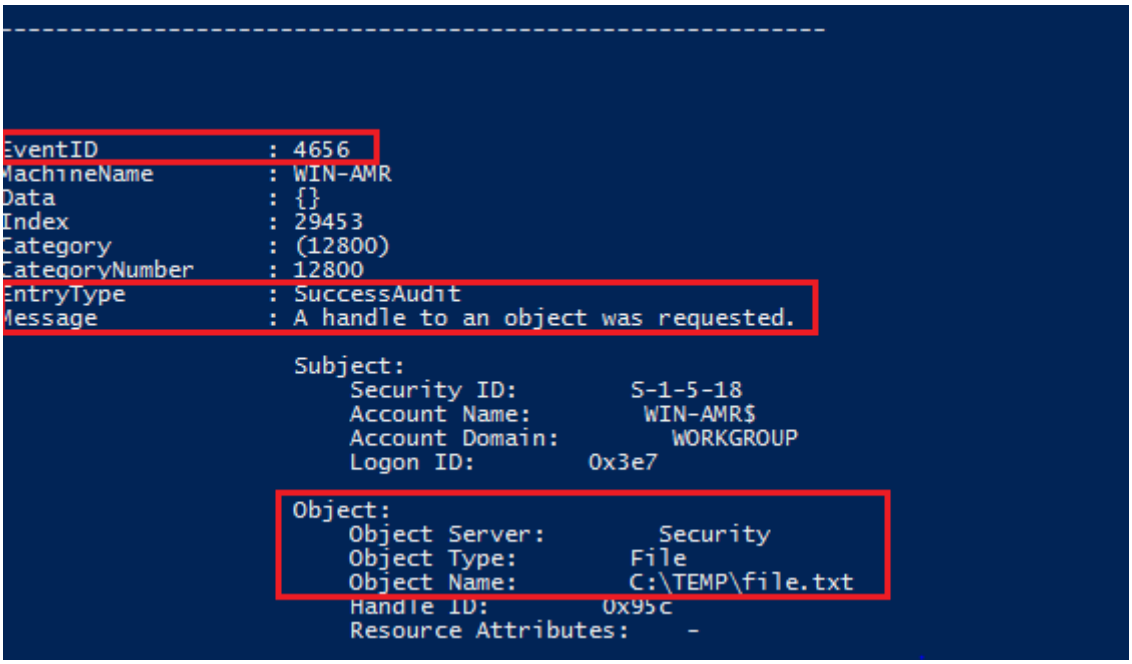
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same result for all the tested platforms. During the test execution, an attempt to access a file is performed using a user who does not have permissions to access it. The result of this action is as follows:

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

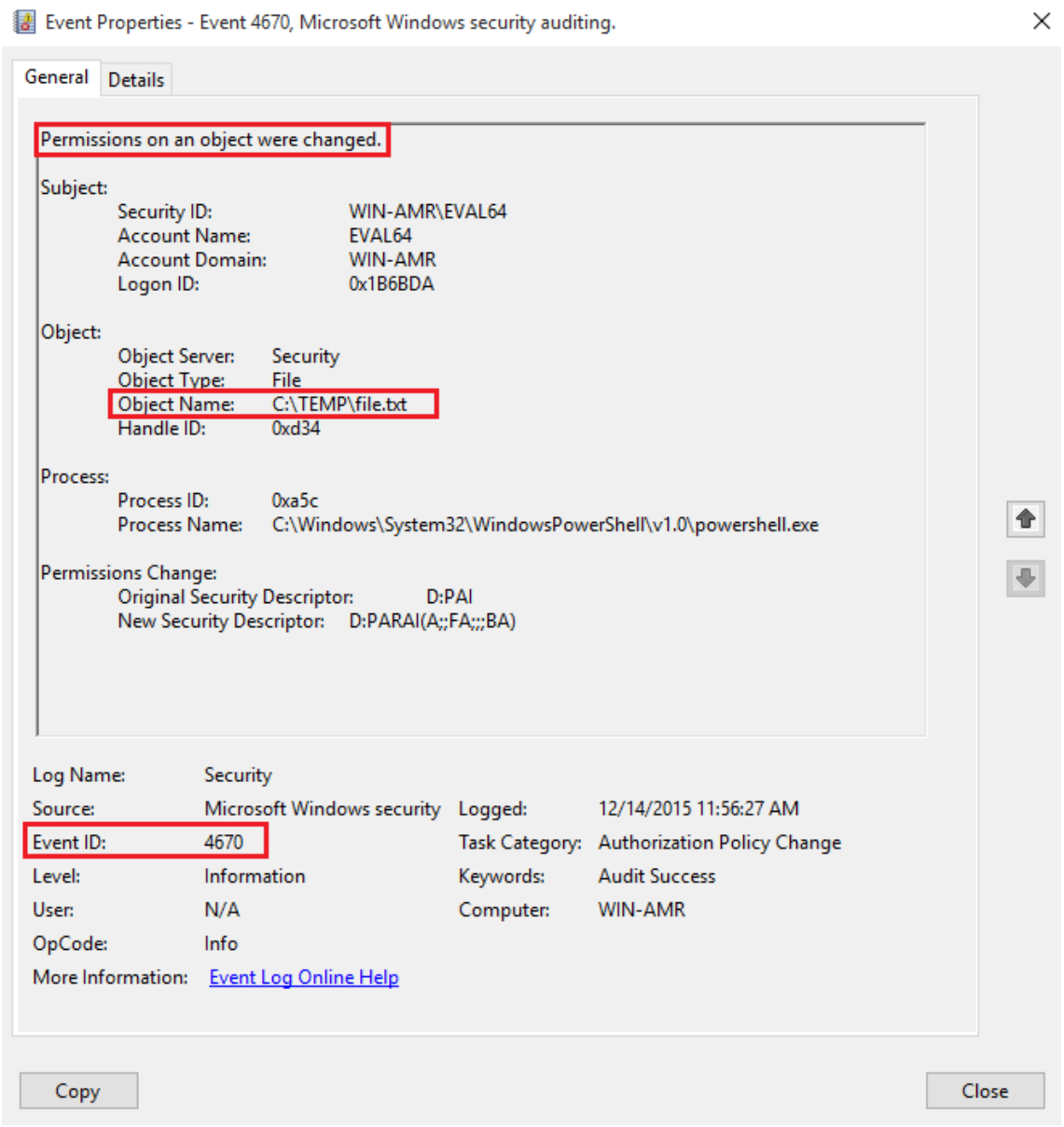
Finally, after accessing the file with a user with permissions, the audit log entries show both the failed and the successful attempt to access the file:

```
------------------------------------------------------------


EventID             : 4656
MachineName         : WIN-AMR
Data                : {}
Index               : 174314
Category            : (12800)
CategoryNumber      : 12800
EntryType           : FailureAudit
Message             : A handle to an object was requested.

                    Subject:
                        Security ID:        S-1-5-21-550595811-875611883-60547835-1076
                        Account Name:       user1
                        Account Domain:     WIN-AMR
                        Logon ID:           0x36c374

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



As it can be observed, the audit log entries include the required fields defined in the security target (Date and time of the event, type of the event, subject identity and outcome).

Additionally, the evaluator has reviewed all the generated events in the *Event Viewer* tool, and has observed that the event 4670 is also generated, which is related to permission modifications on an object.



3.3.4.4. Verdict

As the result above stated, the audit events related to the user and group management events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 4 - File and object events**.

3.3.5. Test 5 - Cryptographic verification events

3.3.5.1. Setup

The following tools must be installed in the platform in order to allow the evaluator perform these tests:

- 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. WinHex.
- A valid update file must be downloaded in the tested platform. To do this, the evaluator shall carry out the following steps:

    1. Open Internet Explorer and browse to *http://catalog.update.microsoft.com*
    2. In the search box type "Windows 10" or "Windows Server 2012 R2" depends on the operating system of the platform which is being tested. A list of available update will be shown.
    3. 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, add the update to the basket.
    4. Finally, click in *View Basket* and after that, click in *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).

3.3.5.2. Procedure

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

1. Create a copy of the update file, which is going to be modified.
2. Open the update file in WinHex editor, modify any bytes and save it.
3. Finally, attempt to install the modified update file and observe that the operating system rejects the operation.
4. Open *Event Viewer* and go to the *Windows Logs -> Setup* section. A new event shall be generated with ID 3 and source WUSA (Windows Update Standalone Installer).

5. After that, attempt to install the original update file. The update shall be installed successfully.

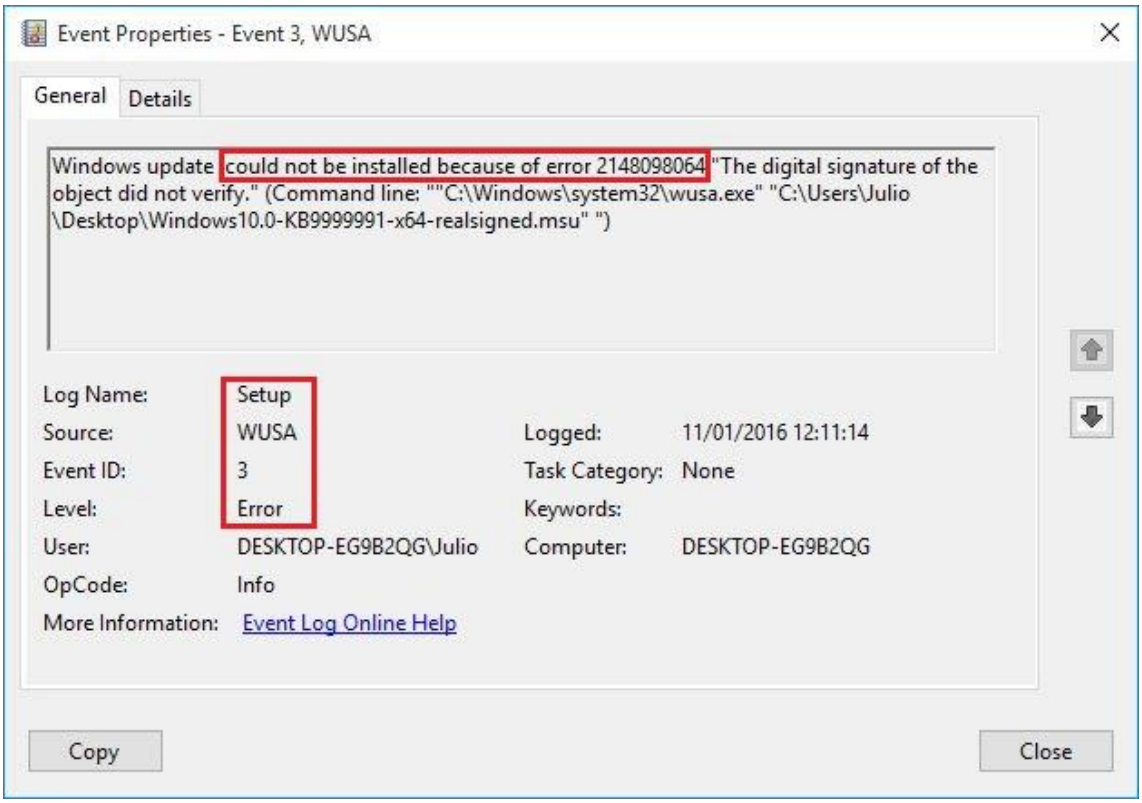
6. Open Event Viewer and go to the “Windows Logs -> Setup” section. A new event shall be generated with ID 2 and Source WUSA.

3.3.5.3. Results

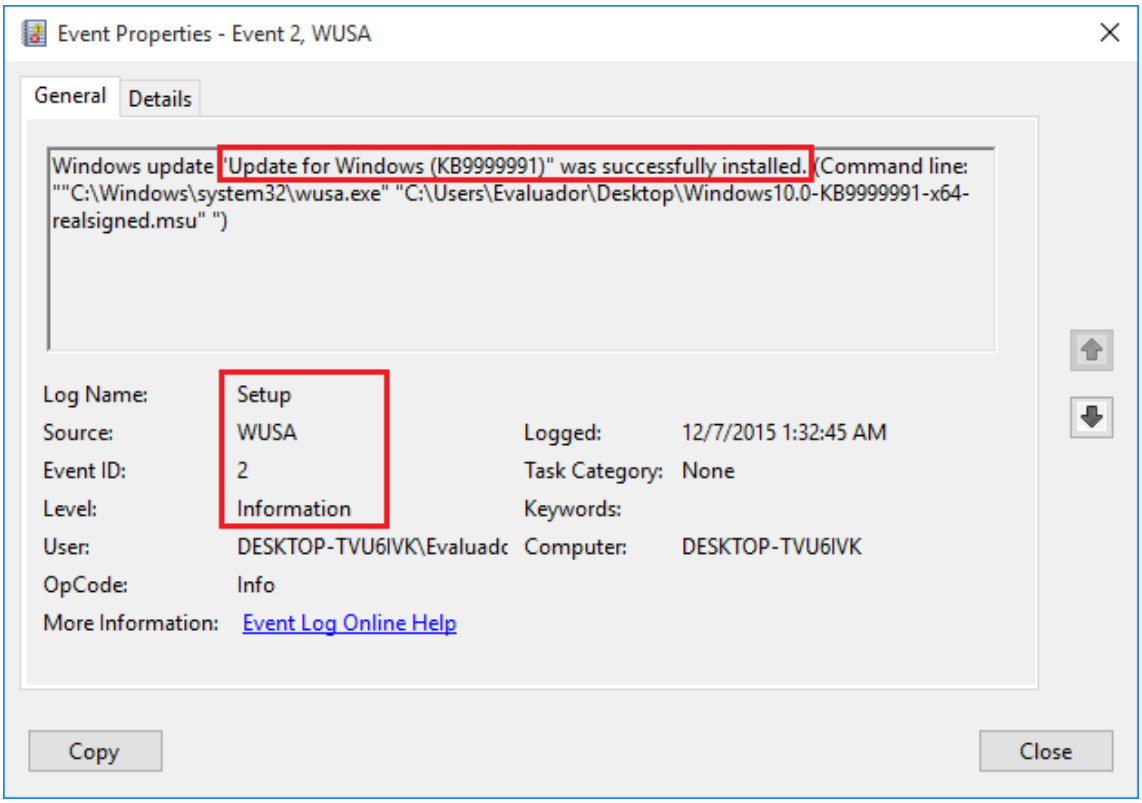
The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same result for all the tested platforms. The evaluator has attempted to install an update after modifying it and the operating system has rejected the operation. The operating system has generated the following event audit:



After that, the evaluator has attempted to install the original update, and it has been installed successfully. The following audit event has been generated:



3.3.5.4. Verdict

As the result above stated, the audit events related to the cryptographic verification events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 5 - Cryptographic verification events**.

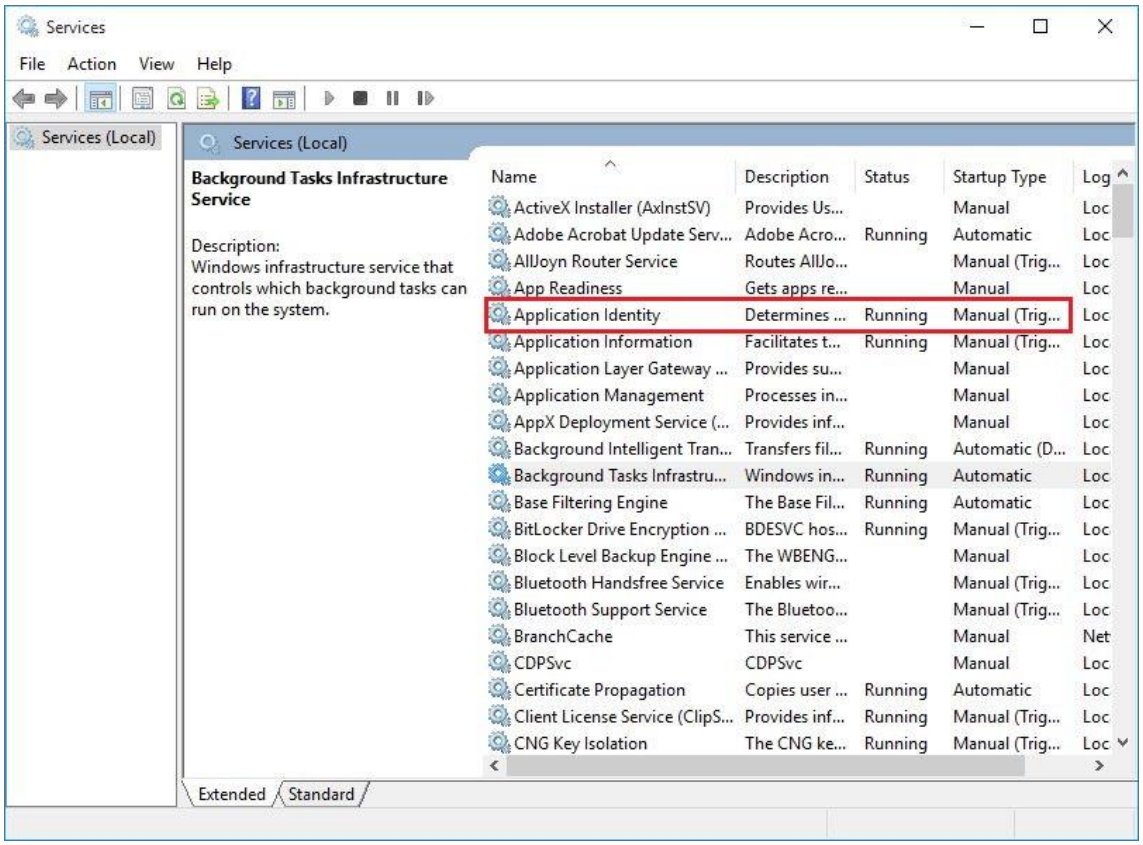
3.3.6. Test 6 - Program initiation events

3.3.6.1. Setup

This test is divided in two different parts, depending on the operating system edition installed in the platforms. Firstly, the evaluator shall use AppLocker, which is only applicable for Windows 10 Enterprise and Windows Server 2012 R2, in order to restrict the application execution. The following conditions must be fulfilled in order to perform this part of the test.

- Application Identity service must be running. The evaluator shall carry out the next steps in order to ensure that the service is already running or start it in case of the service is stopped:

- Right-click in Start button and select Run option. Then type "Services".
- Check that the Application Identity service is running. In case of the service is stopped, right-click over it and then select Start option



- AppLocker shall not have any defined rule in section "Packaged app Rules".

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

- cipolicy.bin file shall be available in the tested platform. This file, which has been provided by the vendor, contains the program restriction polices that it is going to be applied.
- TestConsolev11.exe shall be available in the tested platform. This file, which has been provided by the vendor, is an executable file signed with a non trusted certification authority.

3.3.6.2. Procedure

AppLocker policies are not applied in all TOE versions (it is only applicable for Windows 10 Enterprise and Windows Server 2012 R2), the evaluator has therefore divided the test in two

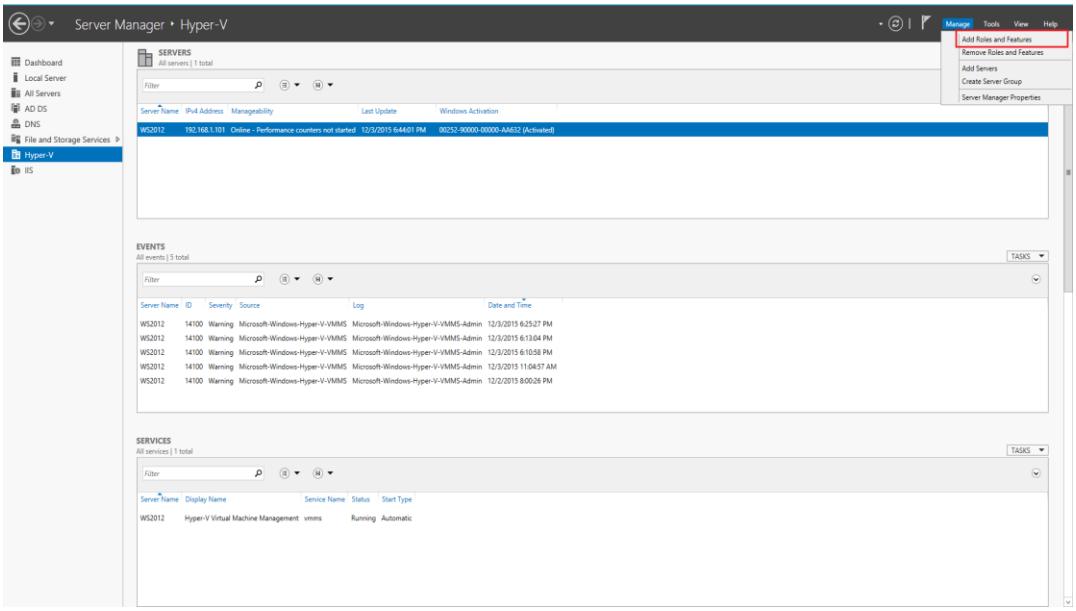
different parts. In the first part, an AppLocker policy is configured in order to restrict the application (*.appx* files) execution. This method only applies to Windows Server 2012 R2 and Windows 10 Enterprise edition. In the second part, the evaluator shall copy a pre-configured policy in order to restrict the execution of executable files which has 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 depends on the tested platform.

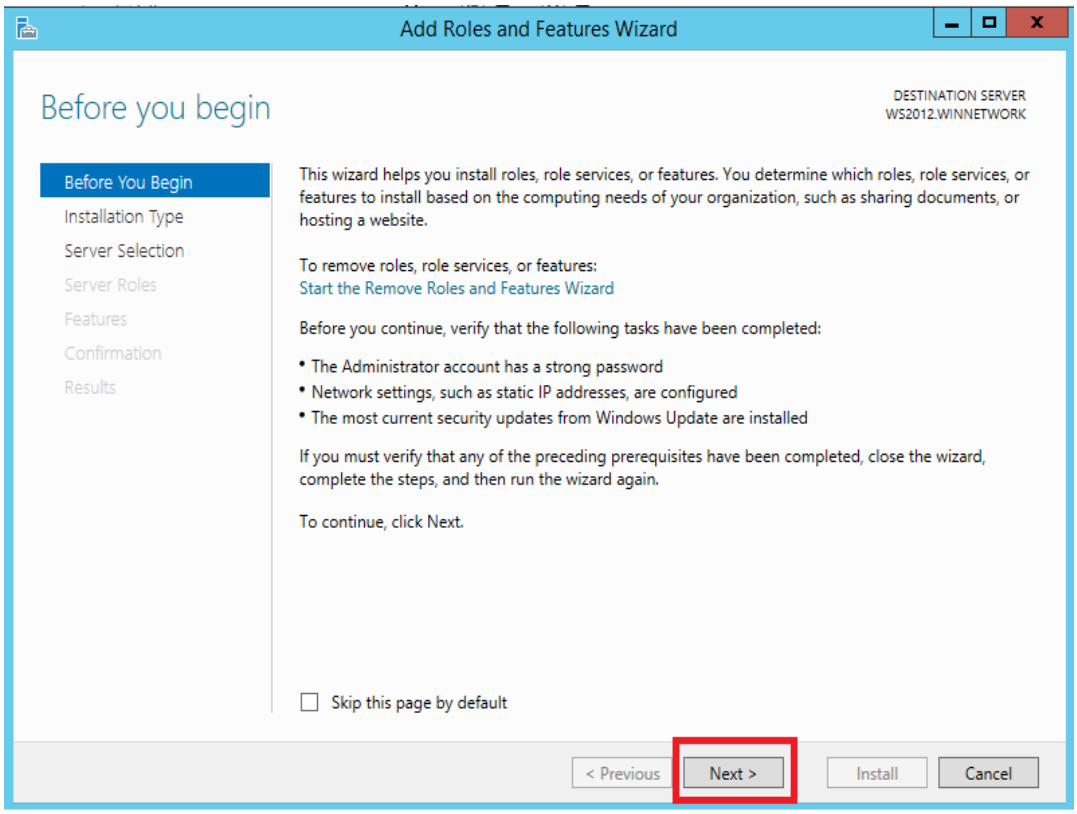
AppLocker - Windows Server 2012 R2

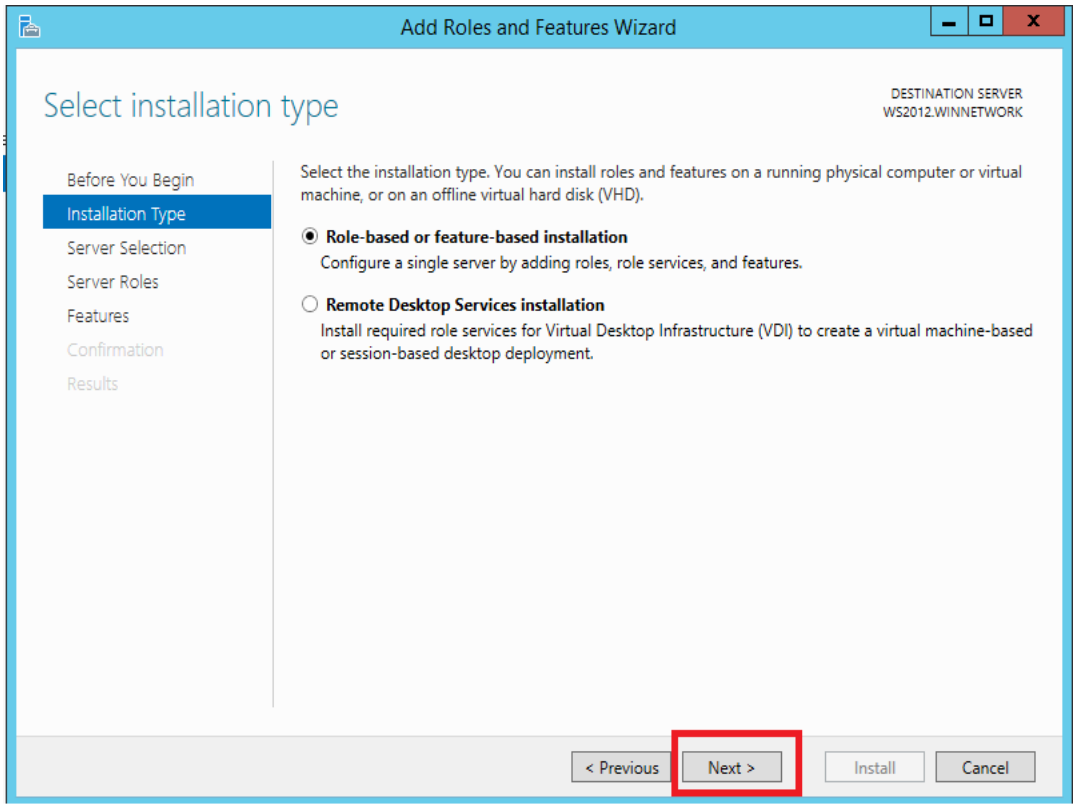
1. Log in using the administrator account.

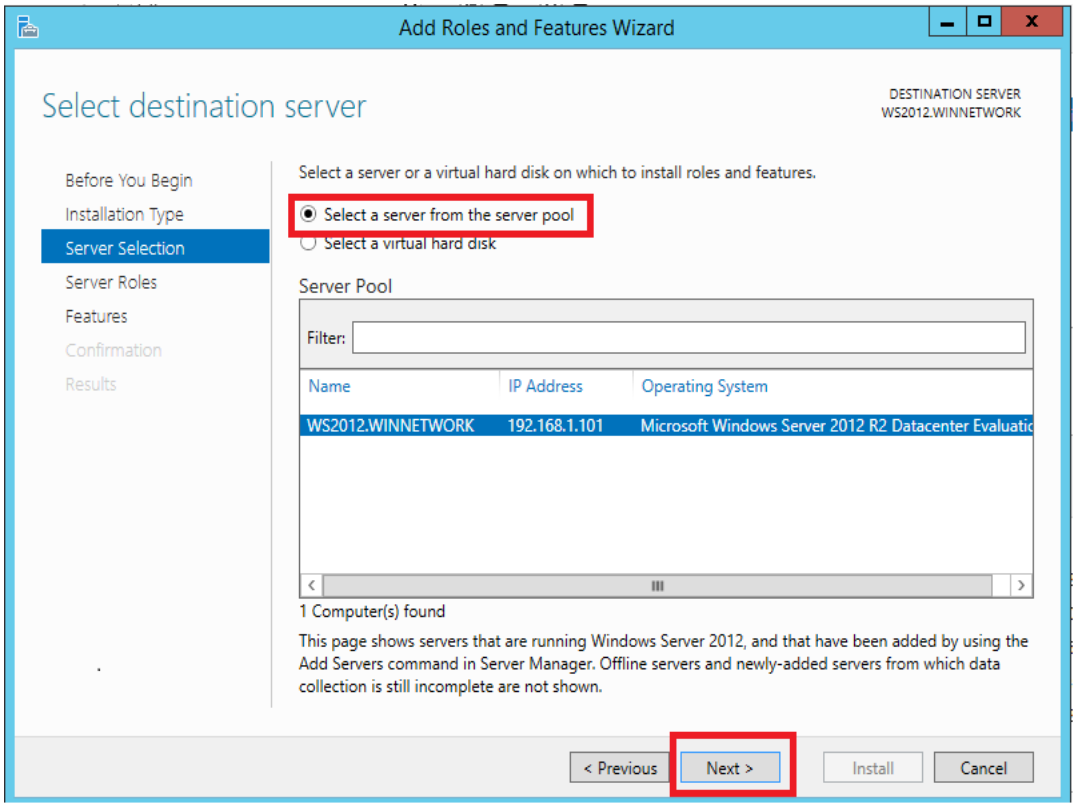
2. In Server Manager, click over Manage and click-in Add Roles and Features.

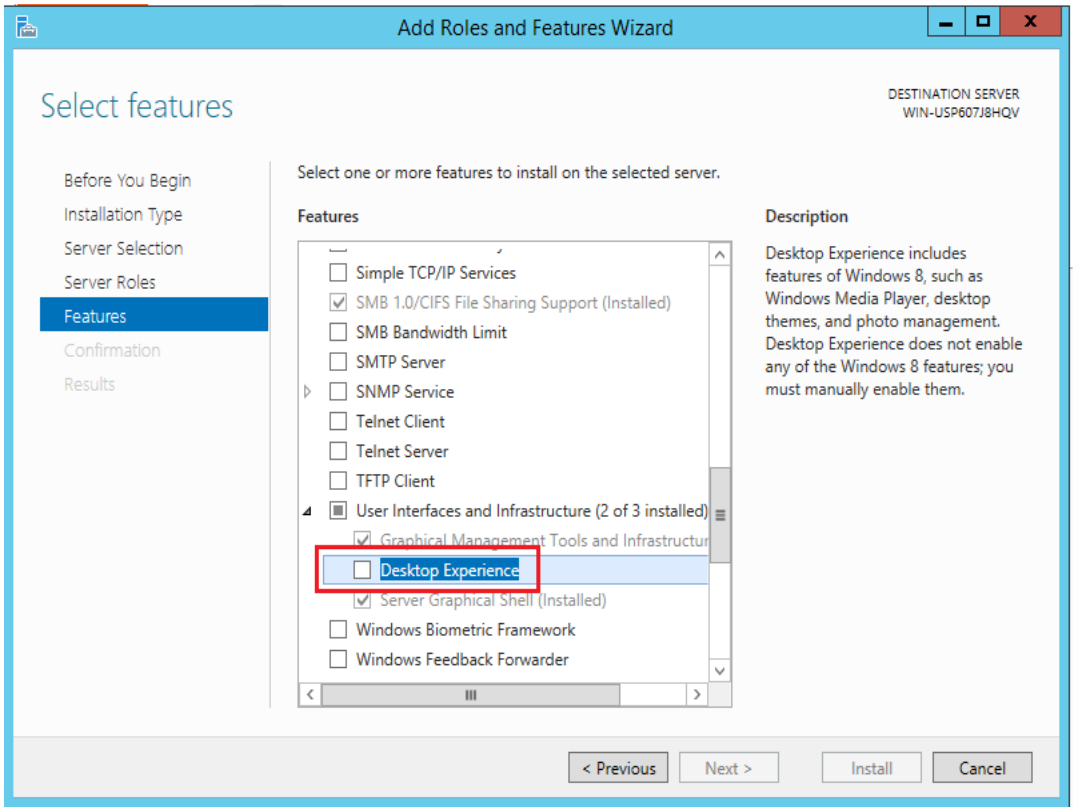


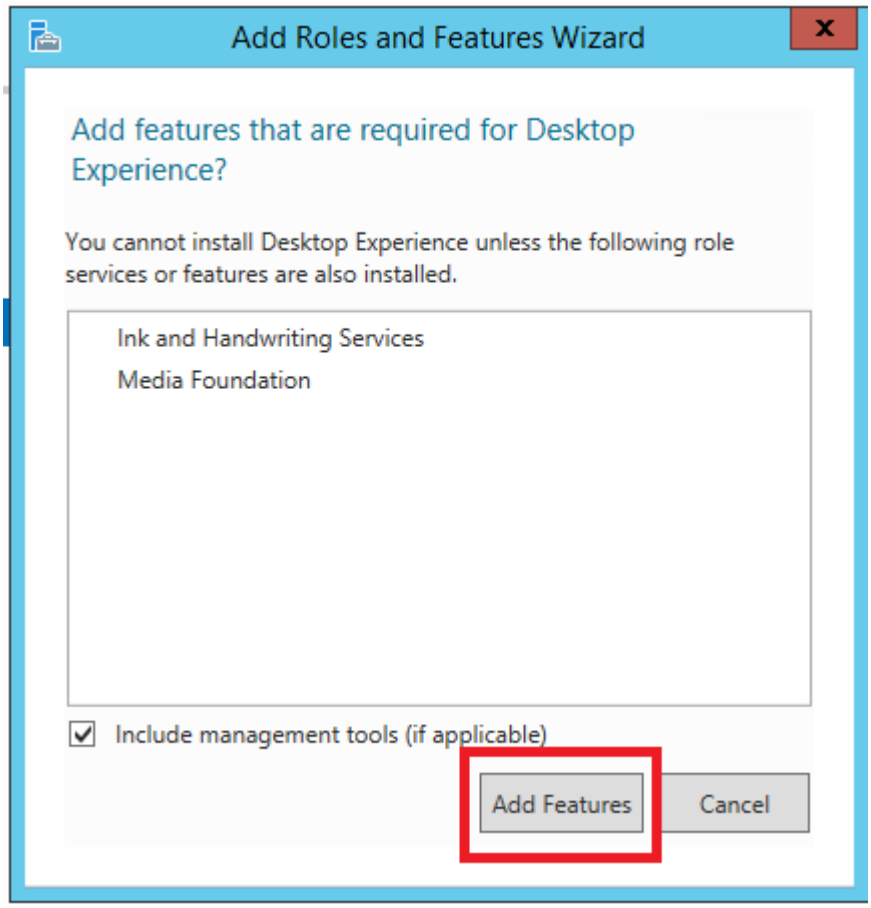
3. The new opened window, shows a wizard, which allow the evaluator add new features and roles. The evaluator shall configure the wizard as shown the next images in order to install the Desktop Experience in the tested platform.



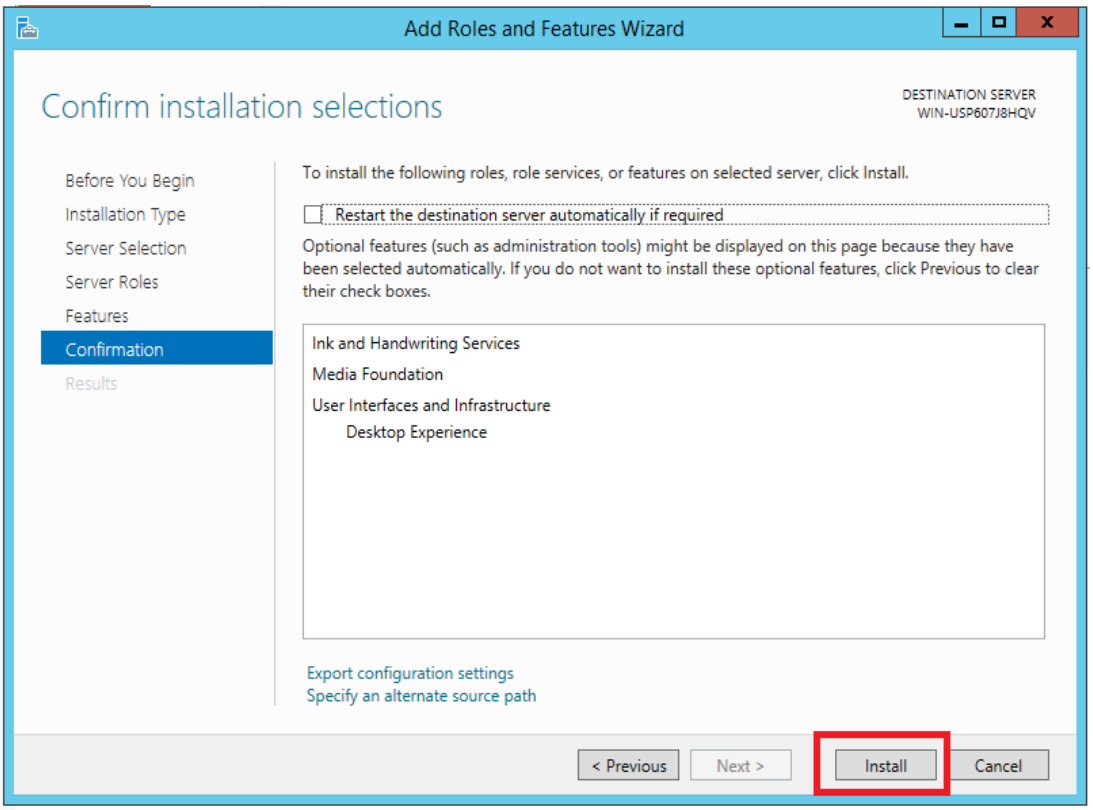




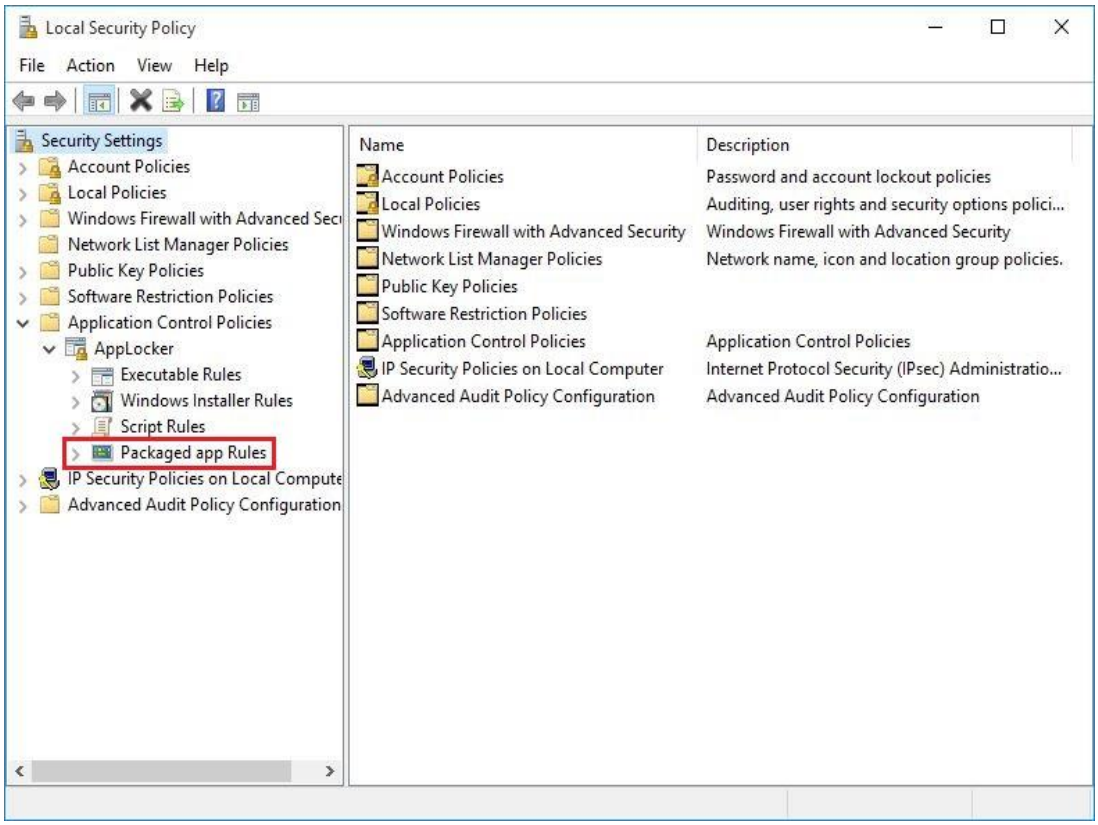




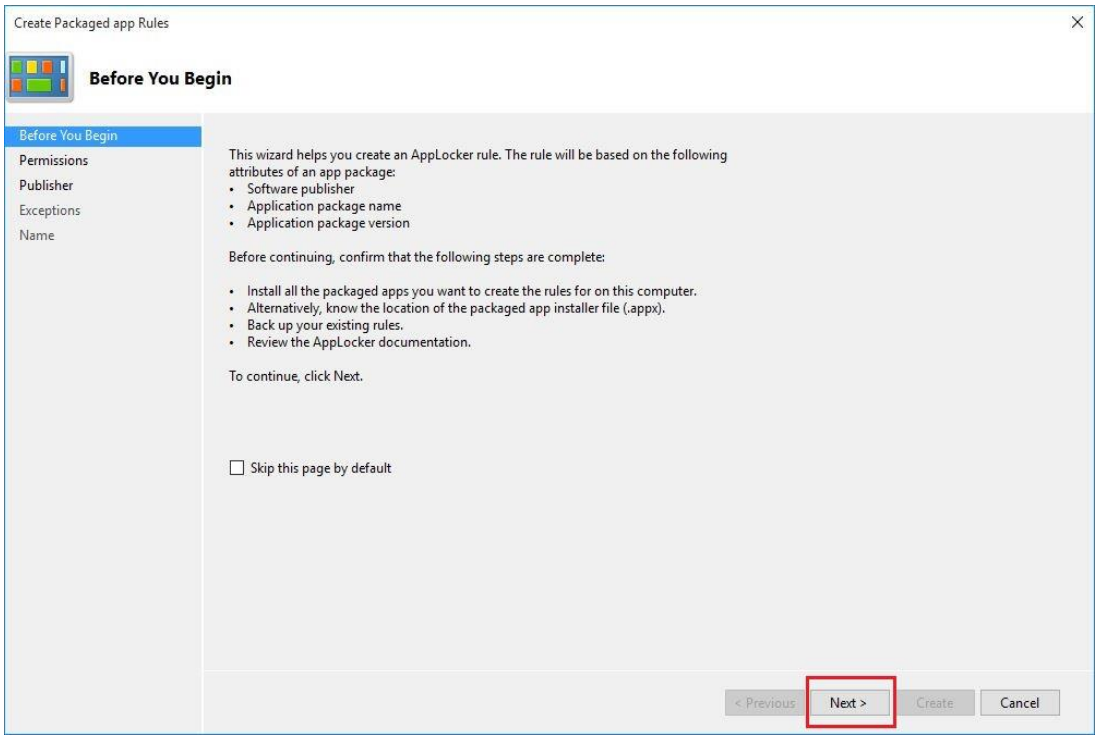
4. Finally, install the new configured features clicking on the Install button. When the installation has finished the tested platform must be restarted in order to apply the changes.

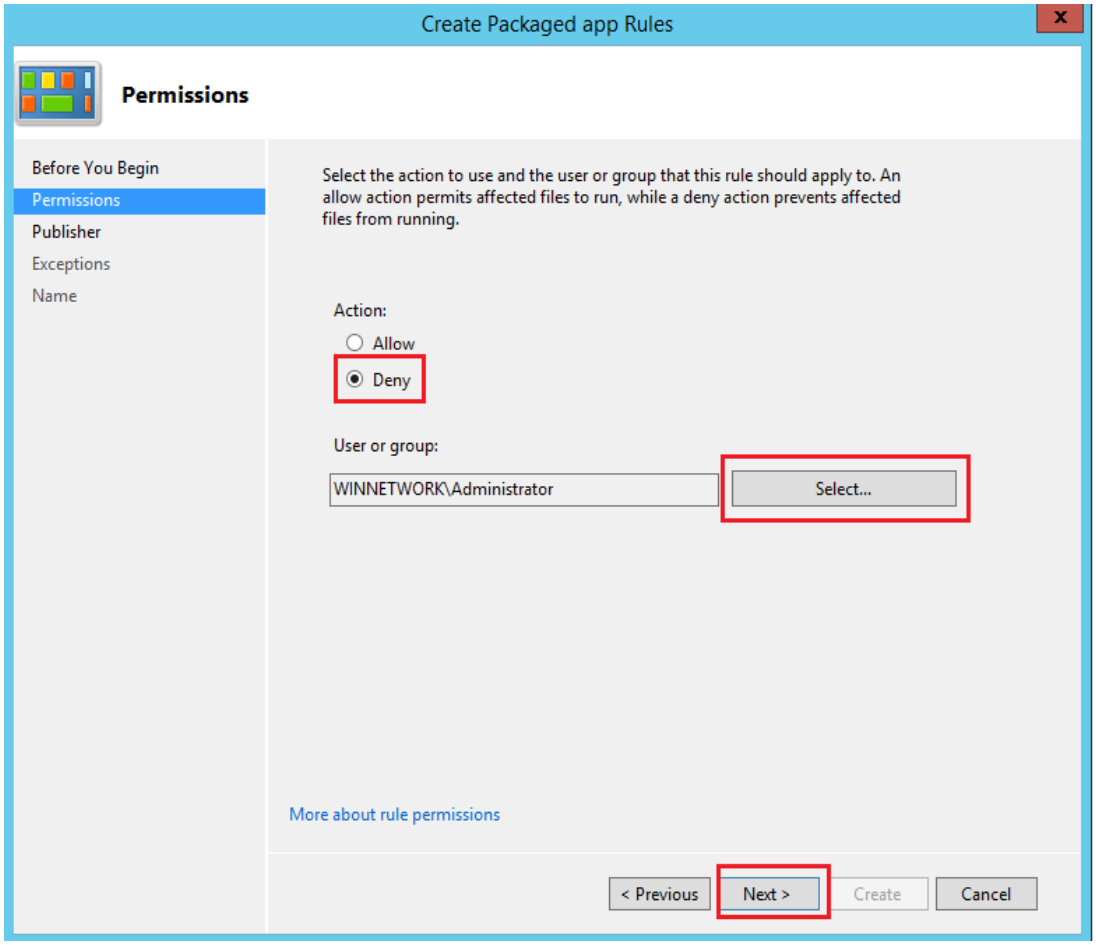


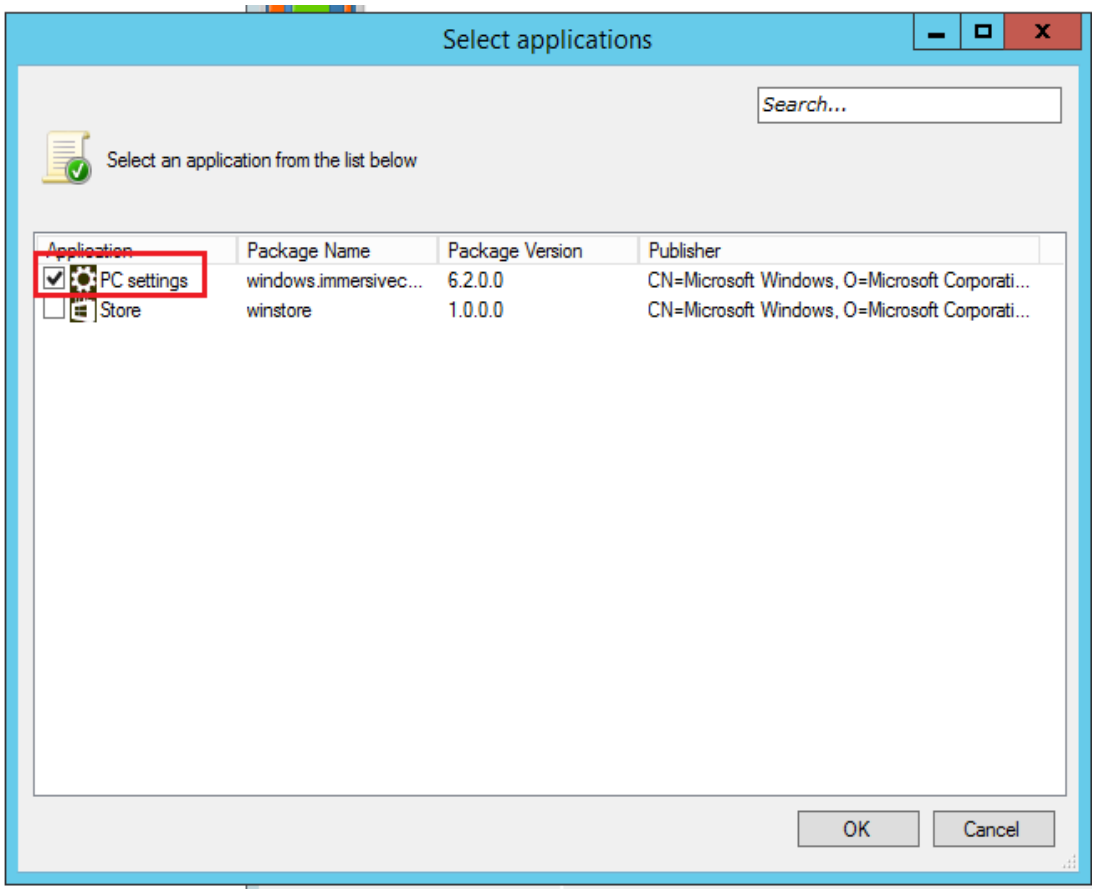
5. After restarting the platform, the evaluator shall configure AppLocker. To do this the evaluator shall open the Local Security Policy tool and go to *Application Control Policies -> AppLocker -> Packaged app Rules*.

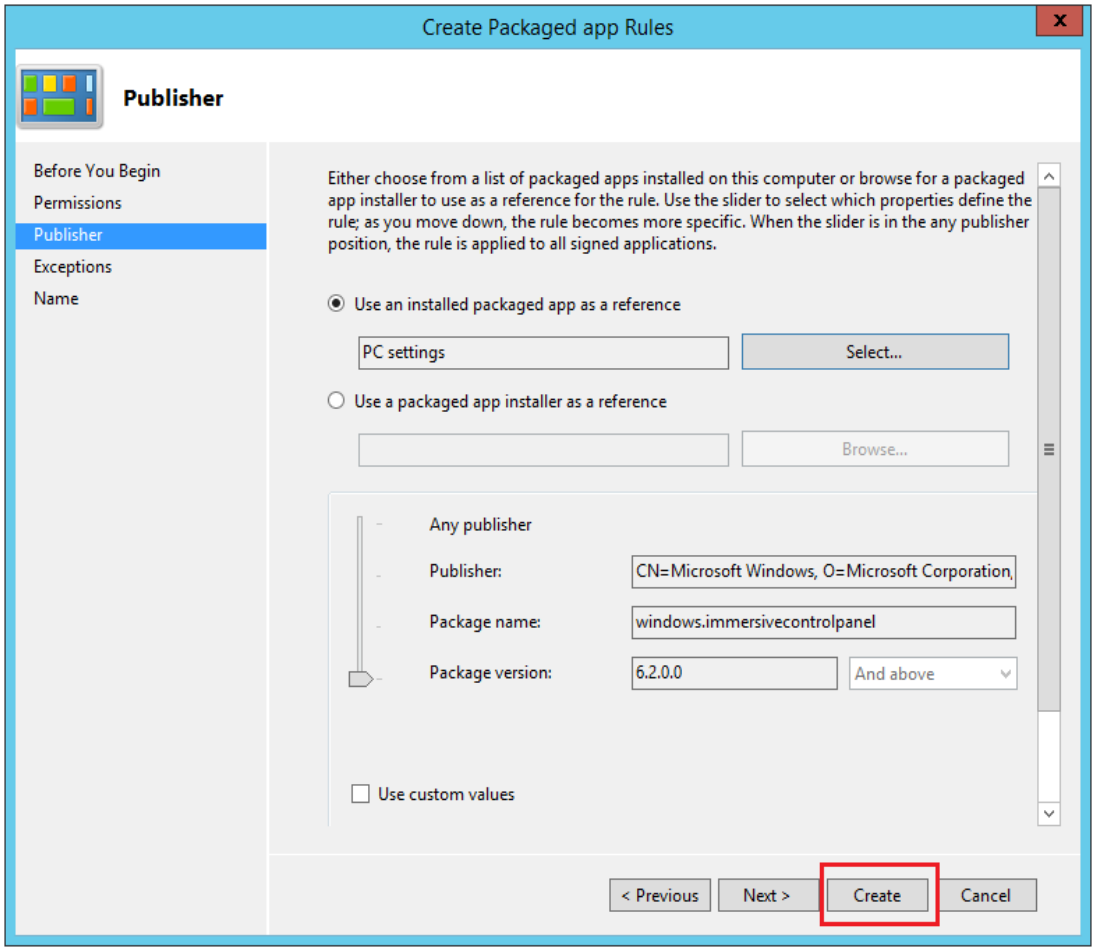


6. Right-click in the *Packaged app Rules* node and select *Create New Rule…* menu item. A new wizard shall be shown, in which the evaluator can configure the software restriction policy which is going to be applied. The evaluator shall configure the wizard as shown the next images:

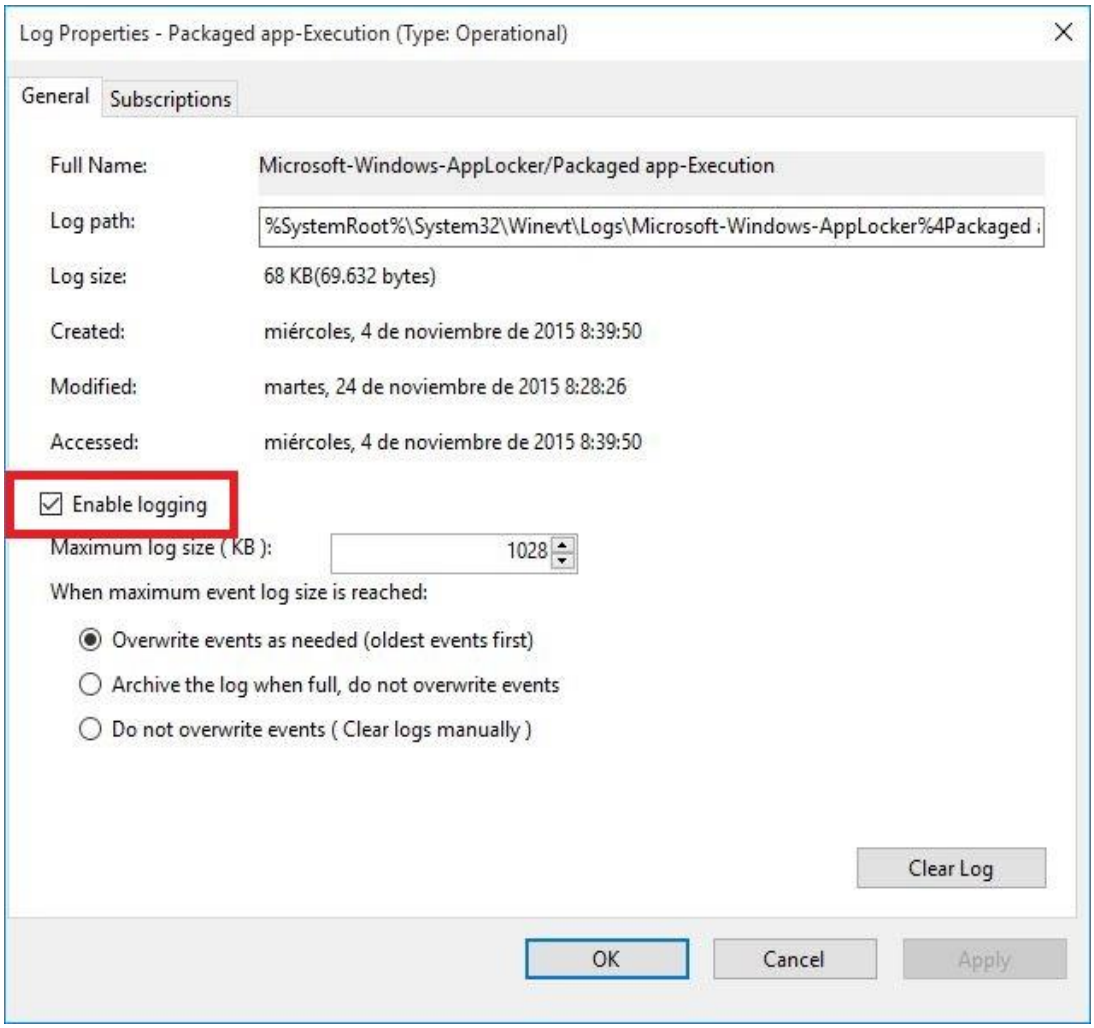






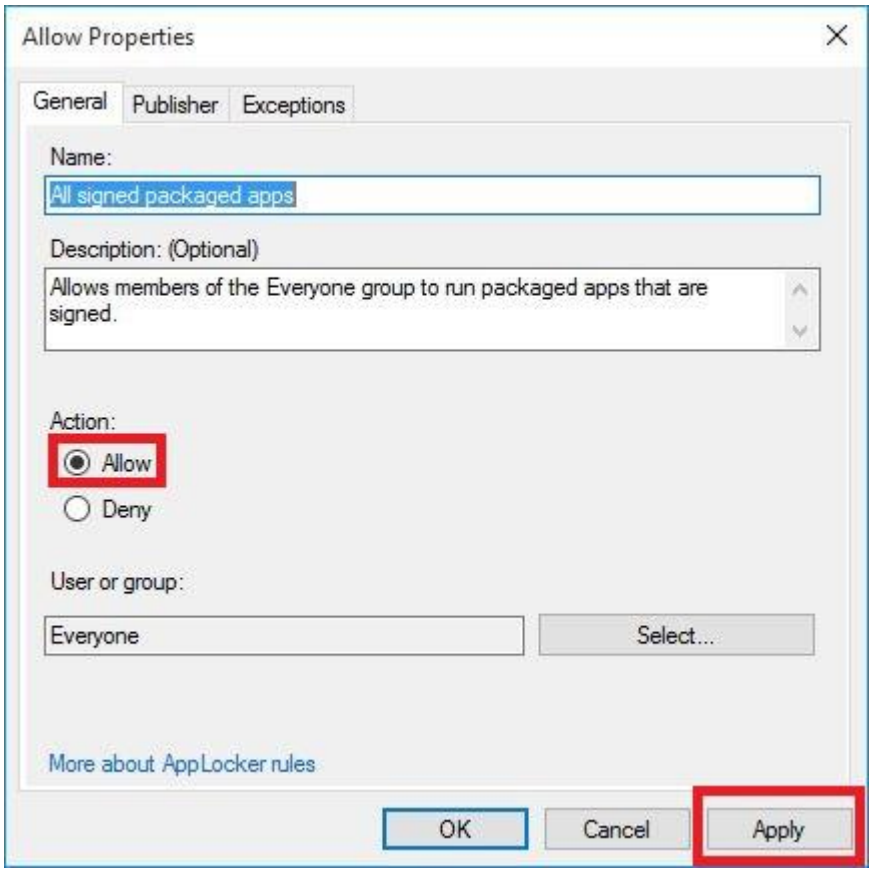


7. Once the software restriction policy has been configured, the audit for the created rule shall be enabled. Open the *Event Viewer* and go to *Applications and Services Logs -> Microsoft -> Windows -> AppLocker -> Packaged app-Execution*. Right-click over *Packaged app-Execution* and open *Properties*. Verify that the *Enable logging* check box is activated.



8. Run *PC settings* application. 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.

9. After that, modify the created rule in order to allow the application execution. To do that run the *Local Security Policy* tool 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*.

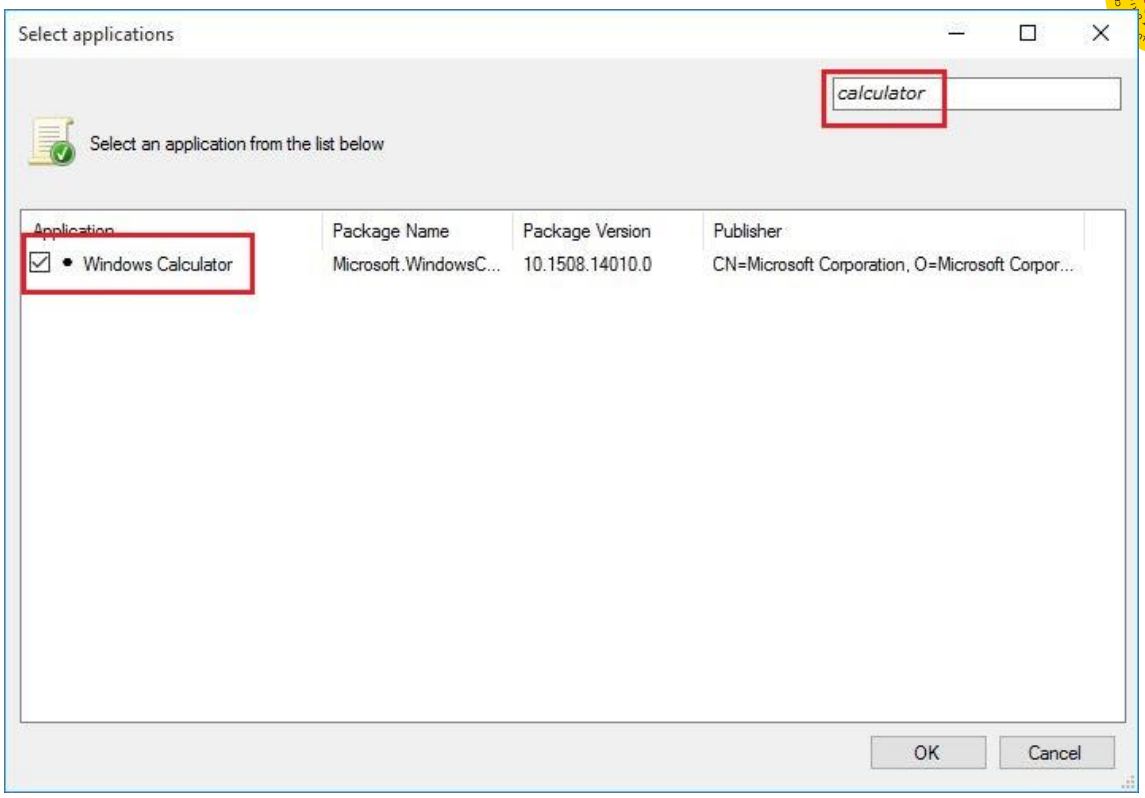


10. Attempt to execute *PC Settings* application. This time the application shall be executed successfully and a new audit event shall be generated. To see the generated audit event, open the Event Viewer and go to *Applications and Services Logs -> Microsoft -> Windows -> AppLocker -> Packaged app-Execution*.

AppLocker - Microsoft Windows 10 Enterprise

1. Log in using the administrator account.

2. Configure a new package app rules in the AppLocker. To do this, the evaluator can repeat the steps 5-7 defined in Windows Server 2012 R2 procedure, but this time using the *Windows Calculator* application.



3. Once the rule is configured, the evaluator shall attempt to execute the *Windows Calculator* application, and the operating system shall reject the execution. Additionally, a new audit event shall be generated indicating that the application has not been executed successfully.

4. After that, modify the created rule in order to allow the application execution. To do that run the Local Security Policy tool 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*.

5. Attempt to execute *Windows Calculator* application. This time the application shall be executed successfully and a new audit event shall be generated. To see the generated audit event, open the Event Viewer and go to *Applications and Services Logs -> Microsoft -> Windows -> AppLocker -> Packaged app-Execution*.

Device Guard - Windows 10 Pro & Home Edition

1. Open a command line terminal and type the following command in order to copy the pre-configured policy file to the tested platform

   copy cipolicy.bin C:\windows\system32\codeintegrity\sipolicy.p7b

2. Restart the TOE to apply the new policy. After that, attempt to execute the TestConsolev11.exe, which has not been signed by a trusted certification authority.

The file shall not execute properly and an error message shall be shown. Additionally, a new audit event shall be generated indicating that there was an error during the program execution.

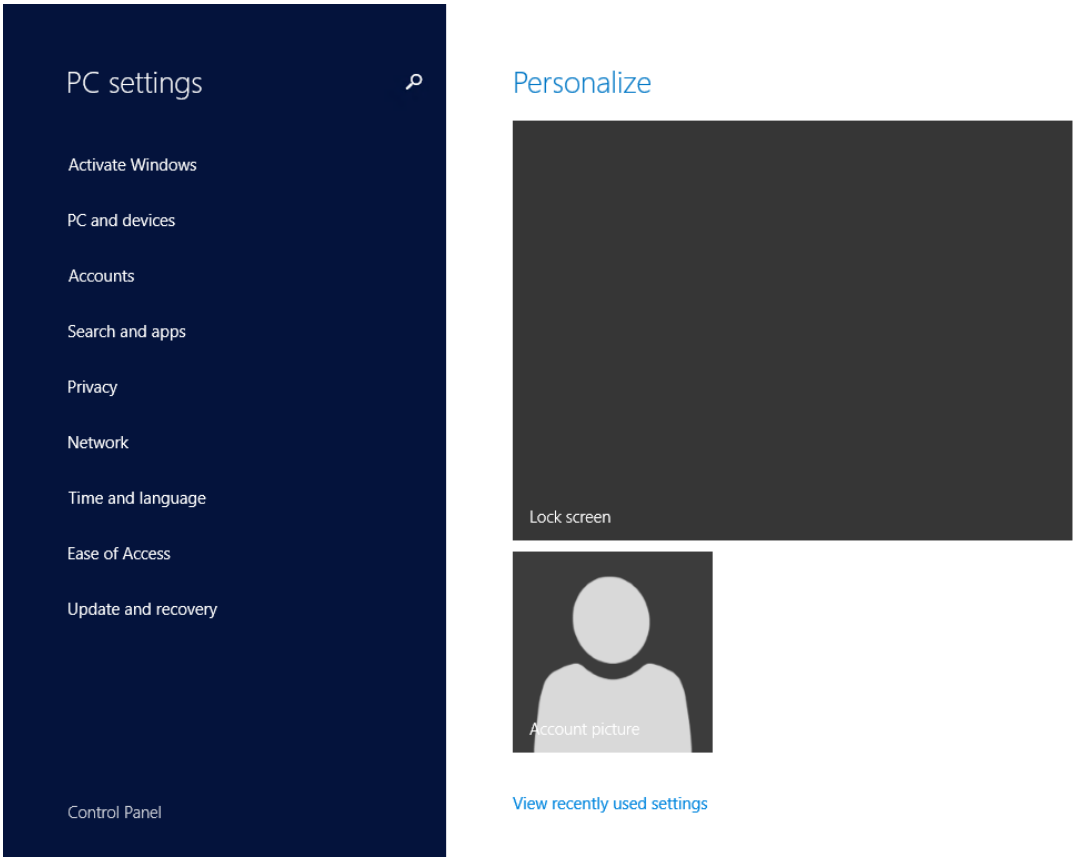
3.3.6.3. Results

The evaluator has performed this test on the following evaluated platforms:

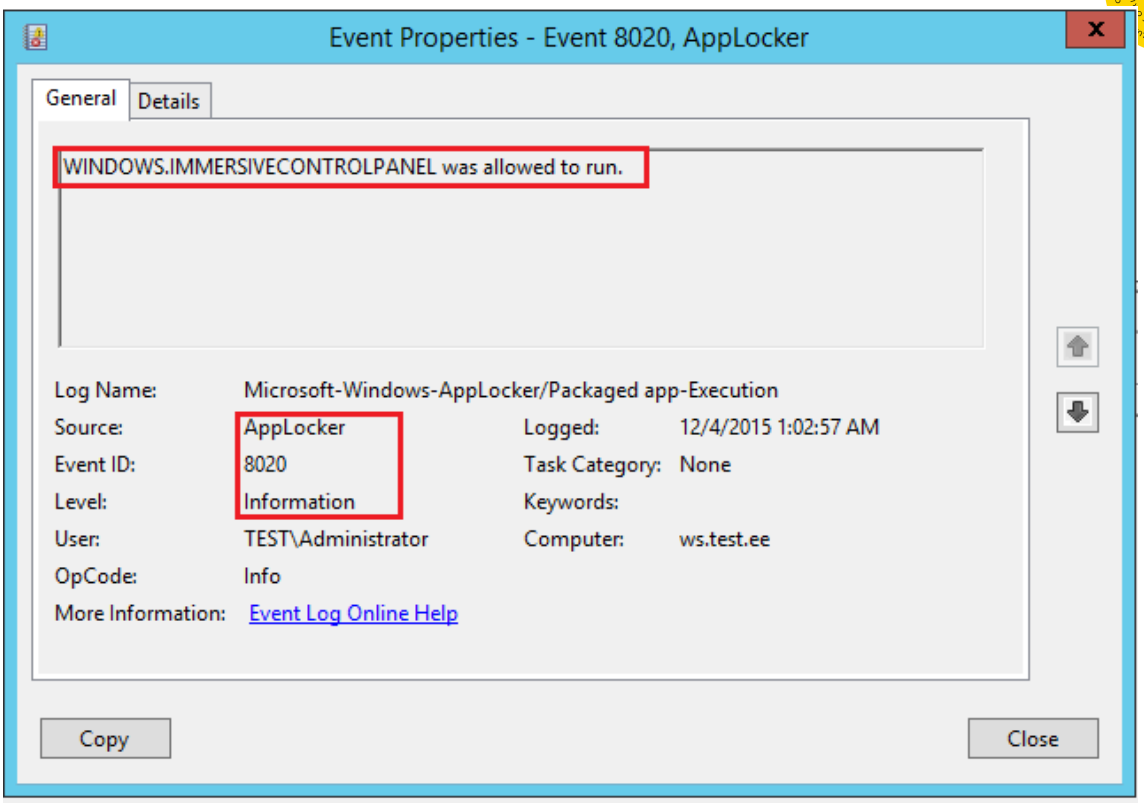
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- HP Pro x2 612 with Windows 10 x64 Pro Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition

The evaluator has obtained the following results for each tested platforms:

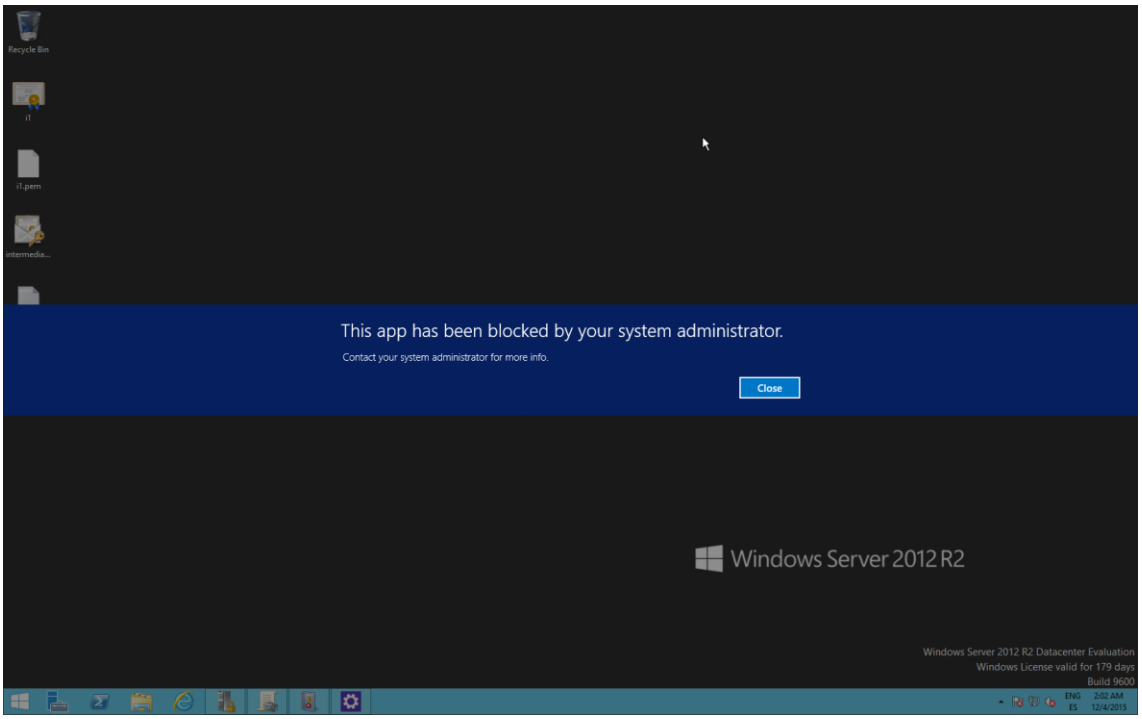
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
    - The evaluator has executed the PC Setting application successfully when the AppLocker rule allows the application execution.



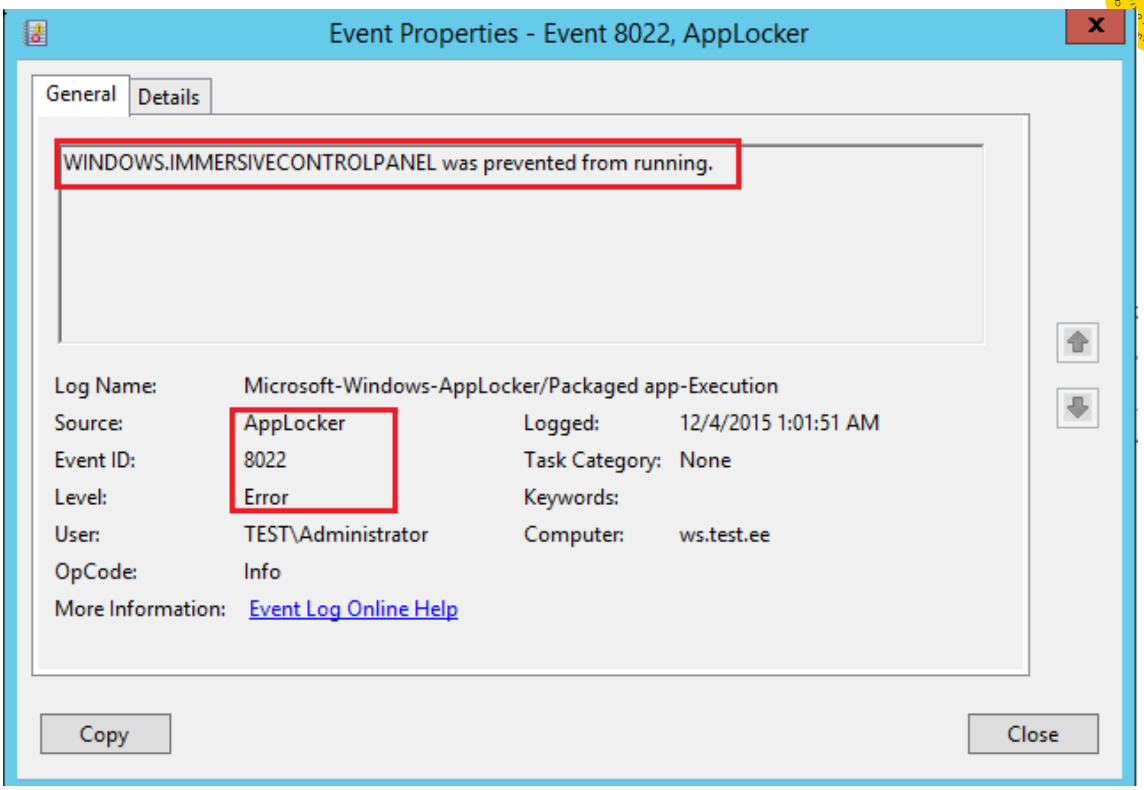
Additionally, the following audit event has been generated, indicating that the application execution has been allowed.



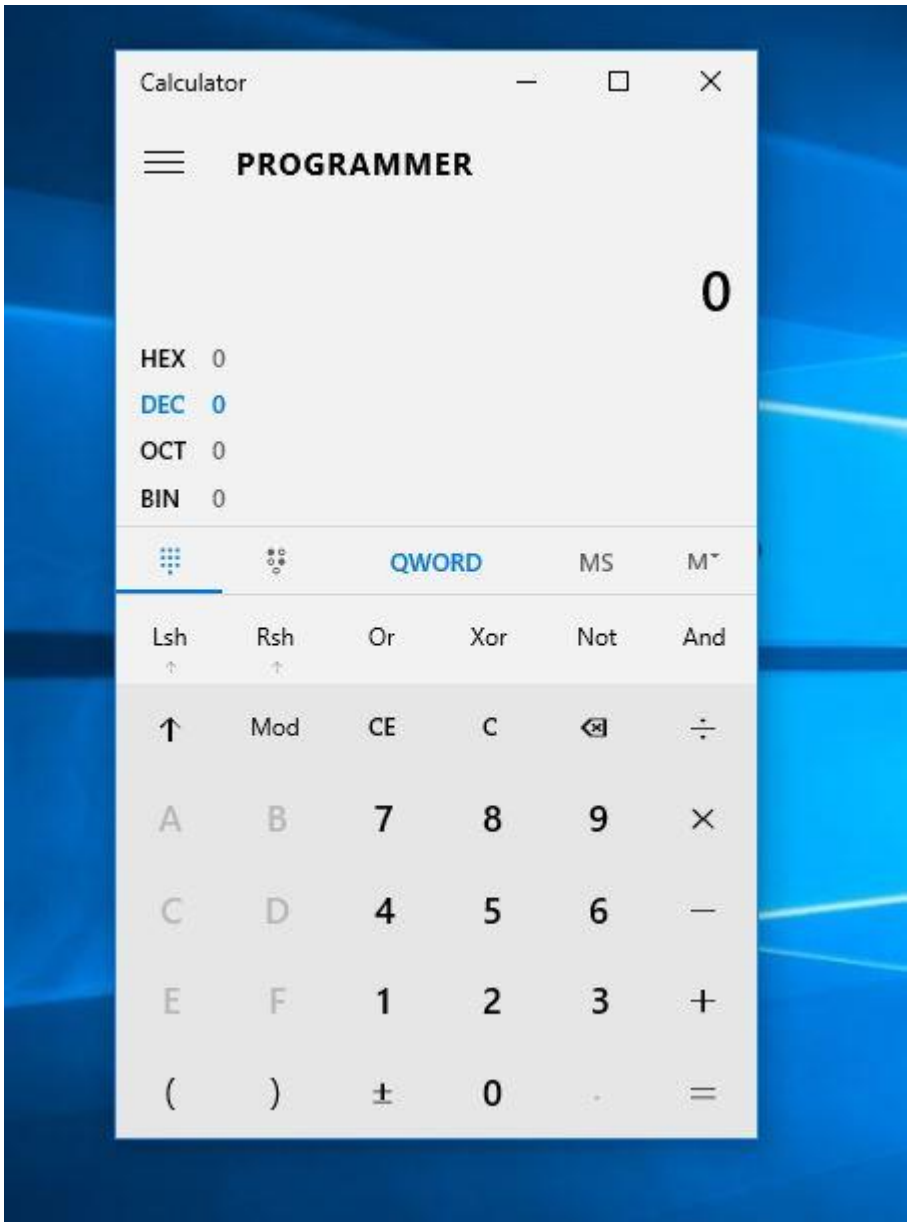
- The evaluator has executed the PC setting application when the AppLocker rule denies the application execution. The obtained results is as follows:



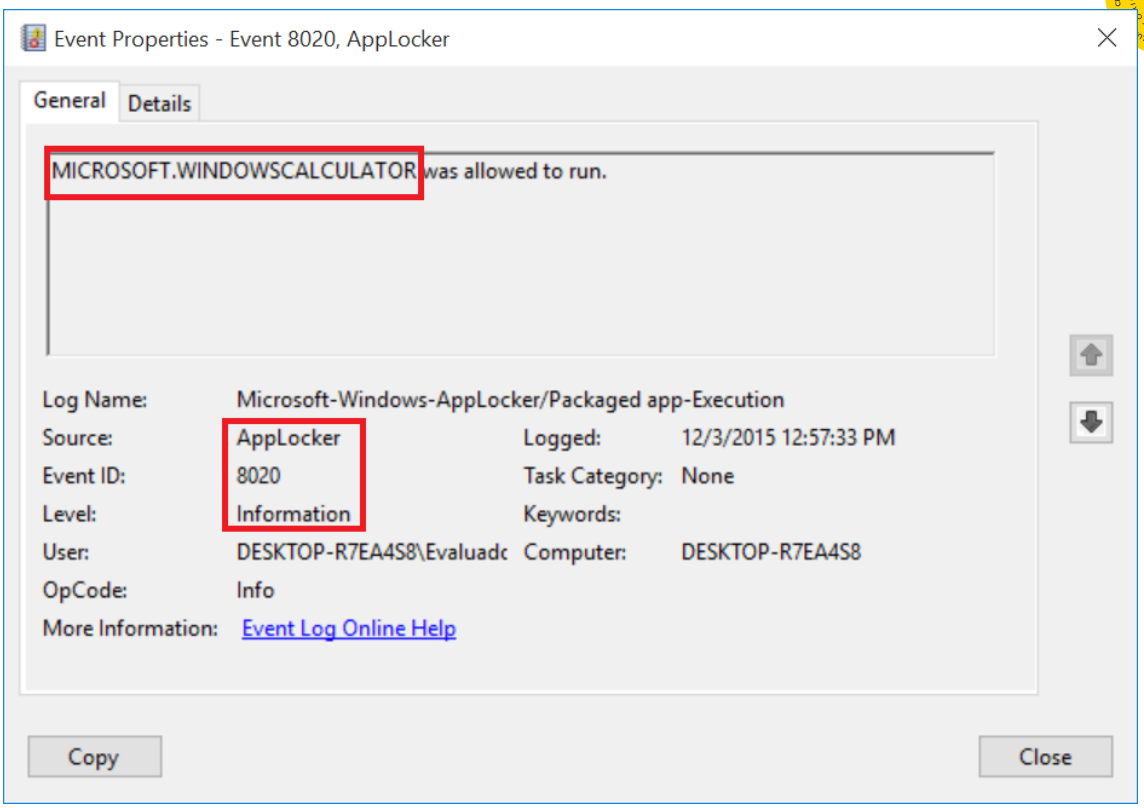
And the following audit event has been generated, indicating that the application execution has not been allowed:



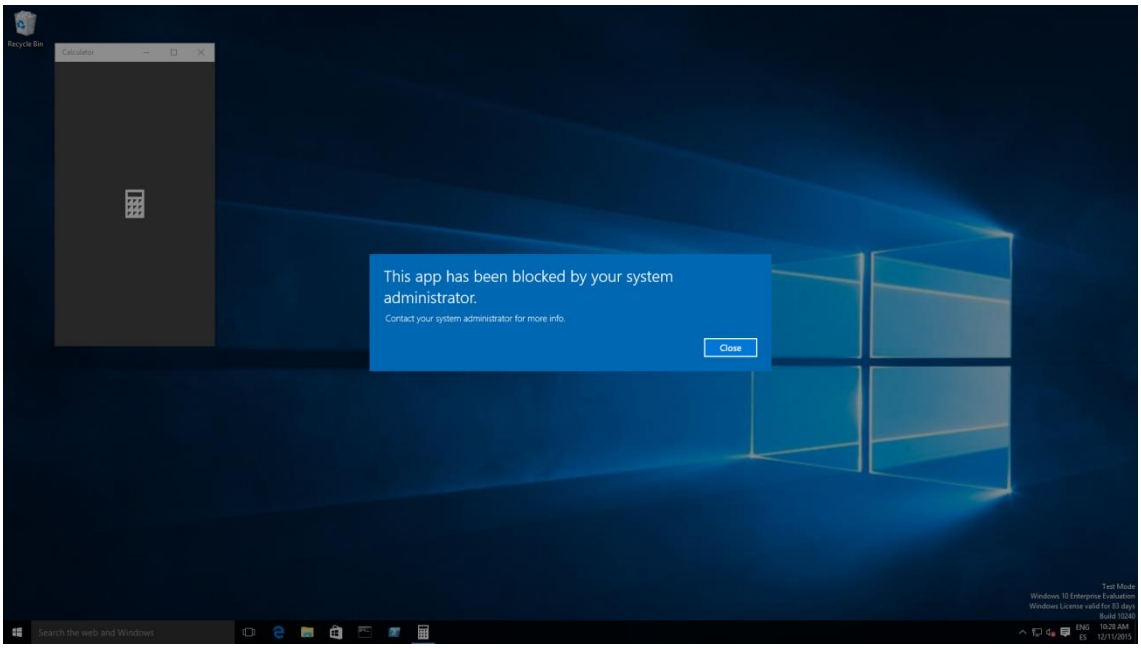
- Surface 3 with Windows 10 x64 Enterprise Edition
    - The evaluator has executed the Windows Calculator application successfully when the AppLocker rule allows the application execution.



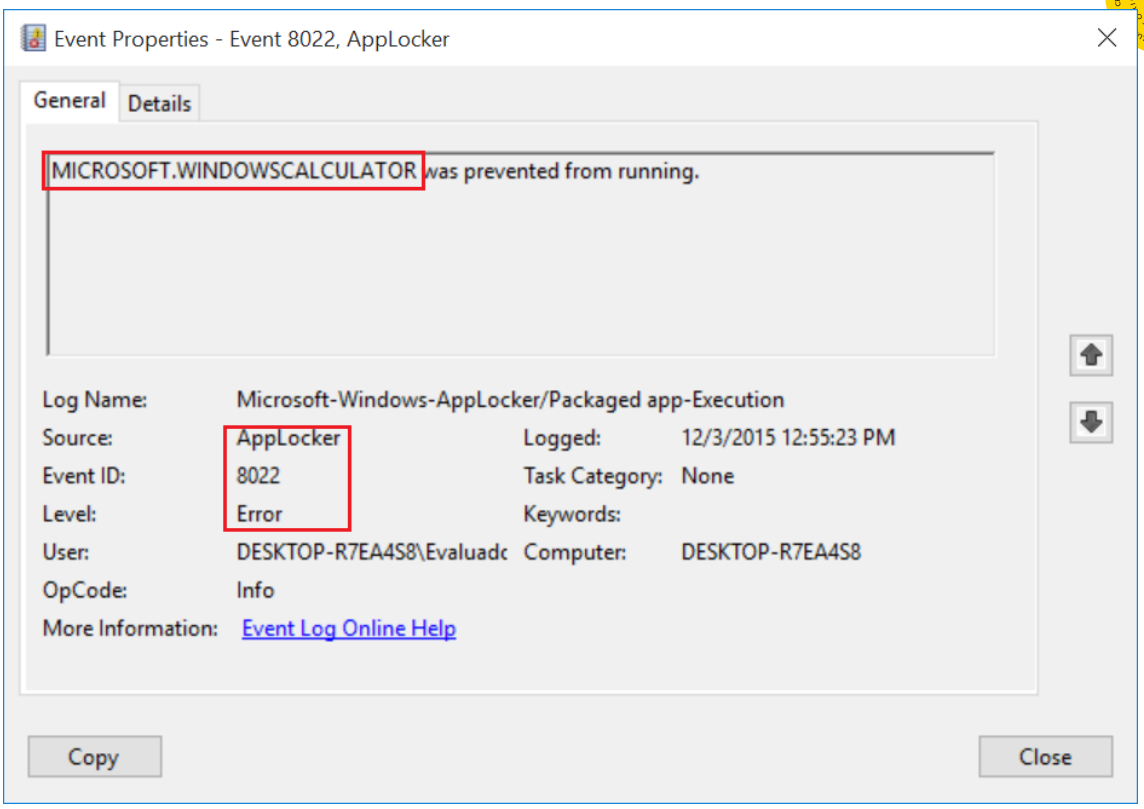
Additionally the following audit event has been generated, indicating that the application execution has been allowed.



- The evaluator has executed the PC setting application when the AppLocker rule denies the application execution. The obtained results is as follows:

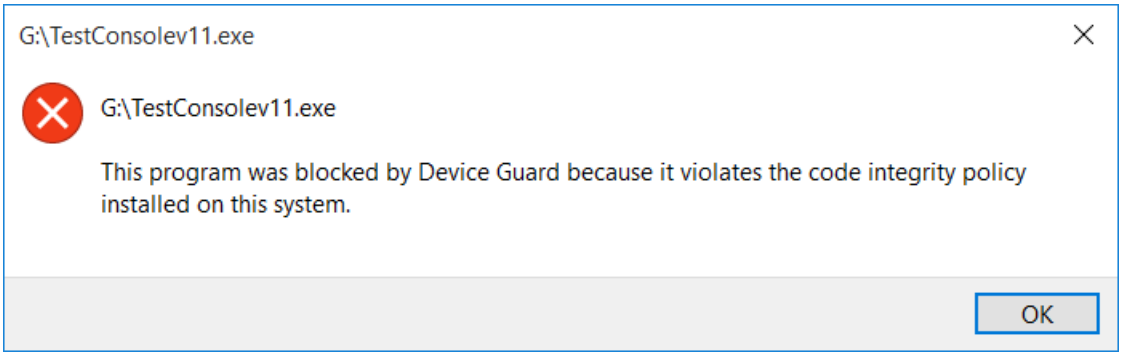


And the following audit event has been generated, indicating that the application execution has not been allowed:

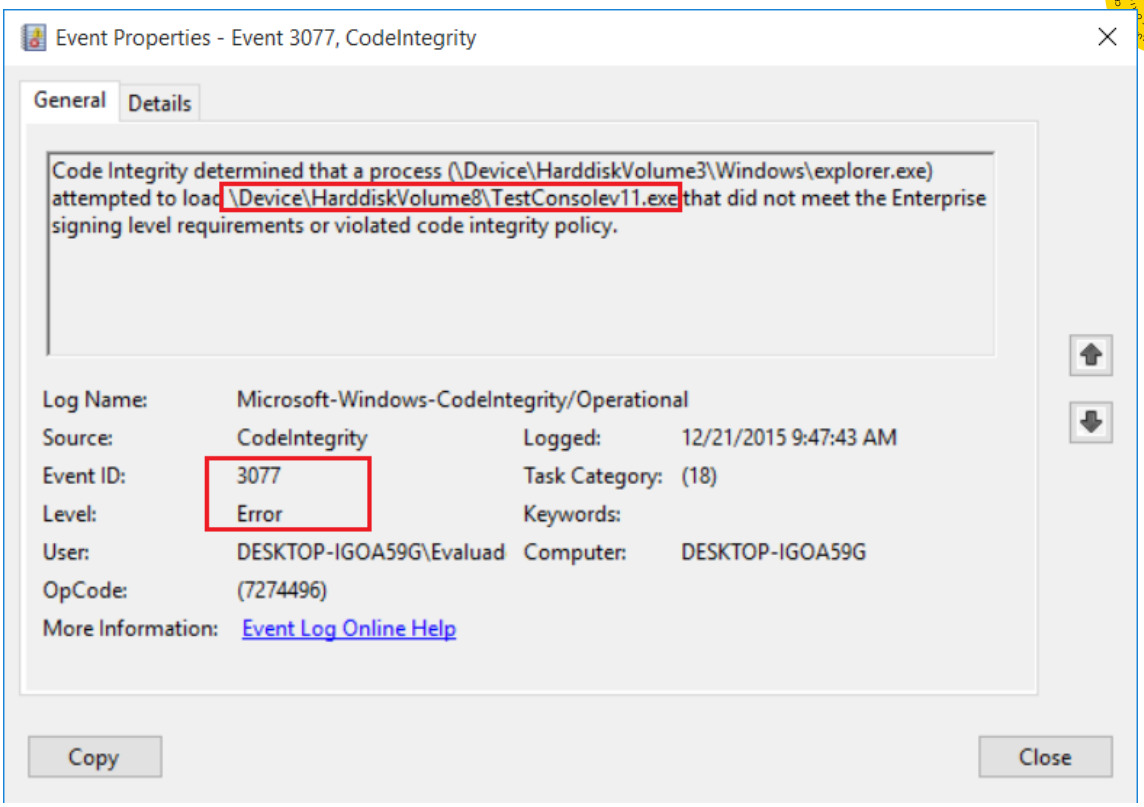


- HP Pro x2 612 with Windows 10 x64 Pro Edition

  - After copy the pre-configure policy file, the evaluator has attempted to execute a executable file and the operating system has rejected the execution showing the following message:



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



- Generation of event with ID 3038 is covered by Test 7 - Kernel module loading events.

3.3.6.4. Verdict

As the result above stated, the audit events related to the program initiation events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 6 - Program initiation events**.

3.3.7. Test 7 - Kernel module loading events

3.3.7.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

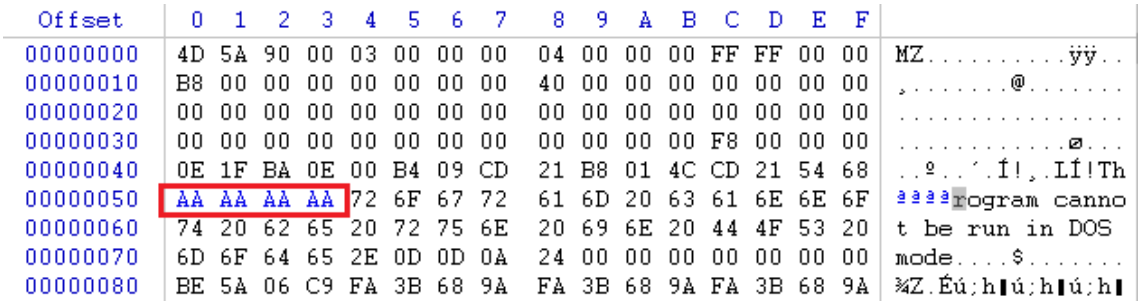
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

- A hexadecimal editor, e.g. WinHex, must be installed in the tested platform.
- 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: *enableAudit.bat*, which enable the CodeIntegrity audit log, and *getLog.ps1*, which generates two text files with the CodeIntegrity audit logs.

3.3.7.2. Procedure

The evaluator shall perform the steps described below in order to perform this test:

1. First of all, 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.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
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	AA	AA	AA	AA	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	ªªªªrogram 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	BE	5A	06	C9	FA	3B	68	9A	FA	3B	68	9A	FA	3B	68	9A	¾Z.Éú;hšú;hšú;hš

2. 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 *enableAudit.bat*. The content of this script is as follows:

```
@echo off
auditpol /set /category:* /success:disable /failure:disable
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
pause
```

3. Boot the TOE into the Windows Pre Environmet using the WinPE USB.
4. Create a copy of the original file, which shall be used in the steps below to restore the corrupt file. Replace the original file with the one modified in step 1 and restart the TOE.
5. Observe that the TOE does not boot properly.

6. Boot the TOE into the Windows Pre Environment again, and restore the original file using the backup created in the step 4.

7. Restart the TOE and observe that this time the TOE boots successfully.

8. Login using the administrator account and open a PowerShell terminal. After that, execute the PowerShell script *getLog.ps1* in order to obtain the Windows-CodeIntegrity/Verbose and Windows-CodeIntegrity/Operational audit logs. The content of this script is as follows:

```
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
$events | fl * > "$path\outputVerbose.txt"
$events = Get-WinEvent -logname Microsoft-Windows-CodeIntegrity/Operational -oldest
$events | fl * > "$path\outputOperational.txt"
```

9. Analyze the audit log obtained in the above step.

3.3.7.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.

The evaluator has obtained the same result for all the tested platforms. Analyzing the audit log generated during the both failed and successful boot process, the evaluator has obtained the following results.

In case of the failed boot process, an event log is generated in Windows-CodeIntegrity/Operational, indicating the name of the corrupt file:

```
Message              : Windows is unable to verify the image integrity of the file
                       \Device\HarddiskVolume2\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              : 0
Qualifiers           :
Level                : 2
Task                 : 1
Opcode               : 104
Keywords             : -9223372036854775808
RecordId             : 1
ProviderName         : Microsoft-Windows-CodeIntegrity
ProviderId           : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName              : Microsoft-Windows-CodeIntegrity/Operational
ProcessId            : 388
ThreadId             : 392
MachineName          : WS2012.WINNETWORK
UserId               : S-1-5-18
TimeCreated          : 12/7/2015 3:50:10 PM
ActivityId           :
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}
```

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

```
Message              : Code Integrity started validating image header of \Windows\System32\sppobjs.dll file.
Id                   : 3038
Version              : 0
Qualifiers           :
Level                : 5
Task                 : 13
Opcode               : 1
Keywords             : 4611686018427387904
RecordId             : 568
ProviderName         : Microsoft-Windows-CodeIntegrity
ProviderId           : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName              : Microsoft-Windows-CodeIntegrity/Verbose
ProcessId            : 3780
ThreadId             : 3788
MachineName          : WS2012.WINNETWORK
UserId               : S-1-5-20
TimeCreated          : 12/7/2015 3:57:42 PM
ActivityId           :
RelatedActivityId    :
ContainerLog         : microsoft-windows-codeintegrity/verbose
MatchedQueryIds      : {}
Bookmark             : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName     : Verbose
OpcodeDisplayName    : Start
TaskDisplayName      :
KeywordsDisplayNames : {}
Properties           : {System.Diagnostics.Eventing.Reader.EventProperty,
                       System.Diagnostics.Eventing.Reader.EventProperty}
```

3.3.7.4. Verdict

As the result above stated, the audit events related to the kernel module loading events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 7 - Kernel module loading events**.

3.3.8. Test 8 - Lock and unlock user account

3.3.8.1. Setup

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

3.3.8.2. Procedure

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

3.3.8.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition

- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the following results during the test execution of FIA_AFL.1, which state that the events related to lock and unlock the user account are generated successfully.

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

```
EventID             : 4740
MachineName         : DESKTOP-IGOA59G
Data                : {}
Index               : 22435
Category            : (13824)
CategoryNumber      : 13824
EntryType           : SuccessAudit
Message             : A user account was locked out.

                      Subject:
                          Security ID:        S-1-5-18
                          Account Name:        DESKTOP-IGOA59G$
                          Account Domain:        WORKGROUP
                          Logon ID:        0x3e7

                      Account That Was Locked Out:
                          Security ID:        S-1-5-21-718890231-1235543865-2686741715-1009
                          Account Name:        evaluadorLocal

                      Additional Information:
                          Caller Computer Name:    DESKTOP-IGOA59G
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {evaluadorLocal, DESKTOP-IGOA59G, S-1-5-21-718890231-1235543865-2686741715-1009, S-1-5-18...}
InstanceId          : 4740
TimeGenerated       : 12/15/2015 12:26:11 PM
TimeWritten         : 12/15/2015 12:26:11 PM
UserName            :
Site                :
Container           :
```

Once the user account has been locked, it has to be unlocked using an administrator account, and the following audit event has been generated:

```
EventID             : 4767
MachineName         : DESKTOP-IGOA59G
Data                : {}
Index               : 22488
Category            : (13824)
CategoryNumber      : 13824
EntryType           : SuccessAudit
Message             : A user account was unlocked.

                      Subject:
                          Security ID:        S-1-5-21-718890231-1235543865-2686741715-1001
                          Account Name:        Evaluador
                          Account Domain:        DESKTOP-IGOA59G
                          Logon ID:        0x206eb9e

                      Target Account:
                          Security ID:        S-1-5-21-718890231-1235543865-2686741715-1009
                          Account Name:        evaluadorLocal
                          Account Domain:        DESKTOP-IGOA59G
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {evaluadorLocal, DESKTOP-IGOA59G, S-1-5-21-718890231-1235543865-2686741715-1009,
                      S-1-5-21-718890231-1235543865-2686741715-1001...}
InstanceId          : 4767
TimeGenerated       : 12/15/2015 12:27:00 PM
TimeWritten         : 12/15/2015 12:27:00 PM
UserName            :
Site                :
Container           :
```

Finally, the audit log entries include the required fields defined in the security target (Date and time of the event, type of the event, subject identity and outcome).

3.3.8.4. Verdict

As the result above stated, the audit events related to the lock and unlock user account events have been generated correctly and they include the information defined in the security target.

Due to this, the evaluator considers that, the results obtained during this subtest activity demonstrate that the analyzed audit events are generated by the TOE when a specific action is performed. Therefore, the **PASS** verdict is assigned to **Test 8 - Lock and unlock user account**.

3.4. 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 and FAU_GEN.1.2.

4. FCS_CKM.1.1

4.1. Assurance activity

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(p1,e) = 1,*
- *GCD(q1,e) = 1,*
- *2 ≤ e ≤ 2 and e is an odd integer,*

- *|p-q| > 2 ^nlen/2 – 100,*
- *p ≥ 2 ^nlen/2 – 1/2 ,*
- *q ≥ 2 ^nlen/2 – 1/2,*
- *2^(nlen/2) < d < LCM(p1,q1),*
- *e□d = 1 mod LCM(p1,q1).*

Key Generation for ANSI X9.311998 RSA Schemes

If the TSF implements the ANSI X9.311998 scheme, the evaluator will check to ensure that the TSS describes how the keypairs are generated. In order to show that the TSF implementation complies with ANSI X9.311998, the evaluator will ensure that the TSS contains the following information:

- *The TSS shall list all sections of the standard to which the OS complies;*
- *For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not") , if the OS implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the OS;*
- *For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described.*

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P256, P384 and P521, 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., P256, P384 and P521, 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 ≤ x ≤ q-1*
  - *len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1 ≤ x ≤ 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 mod p = 1*
- *g mod p = y*

for each FFC parameter set and key pair.

4.2. Documentation review activity

4.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868

Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

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

983	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 10, Microsoft Surface Pro 3 with Windows 10, Microsoft Surface 3 with Windows 10, Microsoft Surface Pro 2 with Windows 10, Microsoft Surface Pro with Windows 10 MsBignum Cryptographic Implementations** Version 10.0.10240	Intel Core i7 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64);AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64); Intel Core i3 without AES-NI or PCLMULQDQ or SSSE 3 w/ Windows 10 (x86); AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 Enterprise (x64); Intel x64 Processor with AES-NI w/ Microsoft Surface Pro w/ Windows 10 Enterprise (x64); Intel Core i5 with AES-NI w/ Microsoft Surface Pro 2	7/31/2015	**FIPS186-4:** **PQG(gen)PARMS TESTED:** [(2048,256)SHA(256); (3072,256) SHA(256)] **PQG(ver)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256)] **Key Pair:** [(2048,256) ; (3072,256)] **SIG(gen)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256);] **SIG(ver)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256)] SHS: Val# 2886 DRBG: Val# 868

505	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 8.1, Microsoft Windows Server 2012 R2, Microsoft Windows Storage Server 2012 R2, Microsoft Windows RT 8.1, Microsoft Surface with Windows RT 8.1, Microsoft Surface Pro with Windows 8.1, Microsoft Surface 2, Microsoft Surface Pro 2, Microsoft Surface Pro 3, Microsoft Windows Phone 8.1, Microsoft Windows Embedded 8.1 Industry and Microsoft StorSimple 8100 MsBignum Cryptographic Implementations** Version 6.3.9600	NVIDIA Tegra 3 Quad-Core w/ Windows RT 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 400 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 800 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows RT 8.1 (ARMv7 Thumb-2); NVIDIA Tegra 3 Quad-Core w/ Microsoft Surface w/ Windows RT 8.1 (ARMv7 Thumb-2); Intel Core i3 without AES-NI and with PCLMULQDQ and SSSE3 w/ Windows 8.1 Enterprise (x64); Intel Core i7 with AES-NI and PCLMULQDQ and SSSE3 w/ Windows 8.1 Enterprise (x64); AMD A4 without AES-NI or PCLMULQDQ or SSSE3 w/ Windows 8.1 Enterprise (x86); AMD Athlon 64 X2 without AES-NI w/ Windows 8.1 Enterprise (x86); Intel Core i7 without AES-NI or PCLMULQDQ or	6/6/2014	**FIPS186-4:** **PKG: CURVES**(P-256 P-384 P-521 ExtraRandomBits) **SigGen: CURVES**(P-256: (SHA-256) P-384: (SHA-384) P-521: (SHA-512) **SigVer: CURVES**(P-256: (SHA-256) P-384: (SHA-384) P-521: (SHA-512)) **SHS: Val#2373** **DRBG: Val# 489** "The Microsoft Windows MSBignum Library algorithm implementation provides DSA, ECDSA, and RSA support to other Microsoft libraries and cryptographic modules." *07/10/2014: Added new tested information;* *12/11/14: Added new tested information;* *03/13/15: Added new tested information;;* *03/18/15: Updated implementation information;*
855	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 8.1, Microsoft Windows Server 2012 R2, Microsoft Windows Storage Server 2012 R2, Microsoft Windows RT 8.1, Microsoft Surface with Windows RT 8.1, Microsoft Surface Pro with Windows 8.1, Microsoft Surface 2, Microsoft Surface Pro 2, Microsoft Surface Pro 3, Microsoft Windows Phone 8.1, Microsoft Windows Embedded 8.1 Industry and Microsoft StorSimple 8100 MsBignum Cryptographic Implementations** Version 6.3.9600	NVIDIA Tegra 3 Quad-Core w/ Windows RT 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 400 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 800 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows RT 8.1 (ARMv7 Thumb-2); NVIDIA Tegra 3 Quad-Core w/ Microsoft Surface w/ Windows RT 8.1 (ARMv7 Thumb-2); Intel Core i3 without AES-NI and with	6/6/2014	**FIPS186-4:** **PQG(gen)PARMS TESTED:** [(2048,256)SHA(256); (3072,256) SHA(256)] **PQG(ver)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256)] **Key Pair:** [(2048,256) ; (3072,256)] **SIG(gen)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256);] **SIG(ver)PARMS TESTED:** [(2048,256) SHA(256); (3072,256) SHA(256)] SHS: Val# 2373 DRBG: Val# 489

On the other hand, the AGD guidance (Windows 10 and Server 2012 R2 GP OS Operational Guidance.docx) states that for the evaluated version the next security policy needs to be applied (section 1.2.1):

- 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 algorithms can be used, including the key generation algorithms defined in the tables above. In addition the vendor has included the following wordings:

- By default Windows 10 and Windows Server 2012 R2 generate asymmetric RSA keys using methods that meet FIPS-PUB 186-4 Appendix B.3. No configuration is necessary to generate keys in this way.
- By default Windows 10 and Windows Server 2012 R2 generate asymmetric ECC keys using methods that meet FIPS-PUB 186-4 Appendix B.4. No configuration is necessary to generate keys in this way.

4.2.2. Verdict

The evaluator considers that the TSS identifies the key sizes supported by the OS for every algorithm through its NIST certificates. On the other hand, 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 AGD guidance 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 more configuration is needed in order 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

4.3. 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.

4.4. Final Verdict

Due to all activities 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 this FCS_CKM.1.1 requirement.

5. FCS_CKM.2.1

5.1. Assurance activity

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).
Assurance 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.

SP80056A 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 shall 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 rolekey 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.

If the OS does not use a KDF defined in SP 80056A, 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 SP80056A 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.

SP80056B Key Establishment Schemes

The evaluator will verify that the TSS describes whether the OS acts as a sender, a recipient, or both for RSAbased 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 RSAbased 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 KTSOAEP is supported), the tester shall generate 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 RSAbased 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 KTSOAEP 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 80056B, 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 KTSOAEP is supported, the evaluator will create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 80056B 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 KTSKEMKWS is supported, the evaluator will create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 80056B 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.

5.2. Documentation review activity

5.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476

Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

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

64	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 10, Microsoft Surface Pro 3 with Windows 10, Microsoft Surface 3 with Windows 10, Microsoft Surface Pro 2 with Windows 10, Microsoft Surface Pro with Windows 10 Cryptography Next Generation (CNG) Implementations** Version 10.0.10240	Intel Core i7 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64); AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64); Intel Core i3 without AES-NI or PCLMULQDQ or SSSE 3 w/ Windows 10 (x86); AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 Enterprise (x64); Intel x64 Processor with AES-NI w/ Microsoft Surface Pro w/ Windows 10 Enterprise (x64); Intel Core i5 with AES-NI w/ Microsoft Surface Pro 2 w/ Windows 10 Enterprise (x64); Intel Core i7 with AES-NI w/ Microsoft Surface Pro 3 w/ Windows 10 Enterprise (x64); Intel Core i3 without AES-NI or PCLMULQDQ or SSSE 3 w/ Windows 10 Enterprise (x86);	8/29/2015	**FFC:** (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Partial Validation) **SCHEMES [dhEphem** (KARole(s): Initiator / Responder) (**FB:** SHA256) (**FC:** SHA256)] [**dhOneFlow** (KARole(s): Initiator / Responder) (**FB:** SHA256) (**FC:** SHA256)] [**dhStatic** (**No_KC** < KARole(s): Initiator / Responder >) (**FB:** SHA256 HMAC) (**FC:** SHA256 HMAC)] SHS Val#2886 DSA Val#983 DRBG Val#868 **ECC:** (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Partial Validation Key Regeneration) **SCHEMES [EphemeralUnified** (**No_KC** < KARole(s): Initiator / Responder >) (**EC:** P-256 SHA256 HMAC) (**ED:** P-384 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512)))] [**OnePassDH** (**No_KC** < KCRole(s): Initiator Responder >) (**EB:**) (**EC:** P-256 SHA256 HMAC) (**ED:** P-521 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512))] [**StaticUnified** (**No_KC** < KARole(s): Initiator / Responder >) (**EC:** P-256 SHA256 HMAC) (**ED:** P-384 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512))] SHS Val#2886 ECDSA Val#706 DRBG Val#868
47	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Windows Storage Server 2012 R2, Microsoft Windows RT 8.1, Microsoft Surface with Windows RT 8.1, Microsoft Surface Pro with Windows 8.1, Microsoft Surface 2, Microsoft Surface Pro 2, Microsoft Surface Pro 3, Microsoft Windows Phone 8.1, Microsoft Windows Embedded 8.1 Industry and Microsoft StorSimple 8100 Cryptography Next Generation Cryptographic Implementations** Version 6.3.9600	NVIDIA Tegra 4 Quad-Core w/ Microsoft Surface 2 w/ Windows RT 8.1 (ARMv7 Thumb-2); NVIDIA Tegra 3 Quad-Core w/ Windows RT 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 400 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Windows Phone 8.1 (ARMv7 Thumb-2) w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows RT 8.1 (ARMv7 Thumb-2); NVIDIA Tegra 3 Quad-Core w/ Microsoft Surface w/ Windows RT 8.1 (ARMv7 Thumb-2); AMD A4 with AES-NI and PCLMULQDQ and SSSE3 w/ Microsoft Windows 8.1 Enterprise (x64); Intel Core i3 without AES-NI and with PCLMULQDQ and SSSE3 w/	7/10/2014	**FFC:** (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Partial Validation) **SCHEMES [dhEphem** (KARole(s): Initiator / Responder) (**FB:** SHA256) (**FC:** SHA256)] [**dhOneFlow** (KARole(s): Initiator / Responder) (**FB:** SHA256) (**FC:** SHA256)] [**dhStatic** (**No_KC** < KARole(s): Initiator / Responder >) (**FB:** SHA256 HMAC) (**FC:** SHA256 HMAC)] SHS Val#2373 DSA Val#855 DRBG Val#489 **ECC:** (FUNCTIONS INCLUDED IN IMPLEMENTATION: DPG DPV KPG Partial Validation Key Regeneration) **SCHEMES [EphemeralUnified** (**No_KC** < KARole(s): Initiator / Responder >) (**EC:** P-256 SHA256 HMAC) (**ED:** P-384 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512)))] [**OnePassDH** (**No_KC** < KCRole(s): Initiator Responder >) (**EB:**) (**EC:** P-256 SHA256 HMAC) (**ED:** P-521 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512))] [**StaticUnified** (**No_KC** < KARole(s): Initiator / Responder >) (**EC:** P-256 SHA256 HMAC) (**ED:** P-384 SHA384 HMAC) (**EE:** P-521 HMAC (SHA512, HMAC_SHA512))] SHS Val#2373 ECDSA Val#505 DRBG Val#489

As it is appreciated in the images above both KAS (key agreement schemes) FFC and ECC are supported in W10 and WS12R2.

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.

On the other hand, the AGD guidance (Windows 10 and Server 2012 R2 GP OS Operational Guidance.docx) states that for the evaluated version the next security policy needs to be applied (section 1.2.1):

- 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 algorithms can be used, including the key agreement algorithms defined in the tables above. In addition the vendor has included the following wordings:

- By default Windows 10 and Windows Server 2012 R2 implement RSA-based key establishment schemes that meet SP-800-56B. No configuration is necessary to perform RSA-based key establishment in this way.

- By default Windows 10 and Windows Server 2012 R2 implement elliptic curve-based key establishment schemes that meet SP-800-56A. No configuration is necessary to perform elliptic curve-based key establishment in this way

5.2.2. 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 configurations are needed for using the supported key agreement schemes.

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

5.3. Test Activity

The evaluator has reviewed the NIST certificates (#47 and #64) 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.

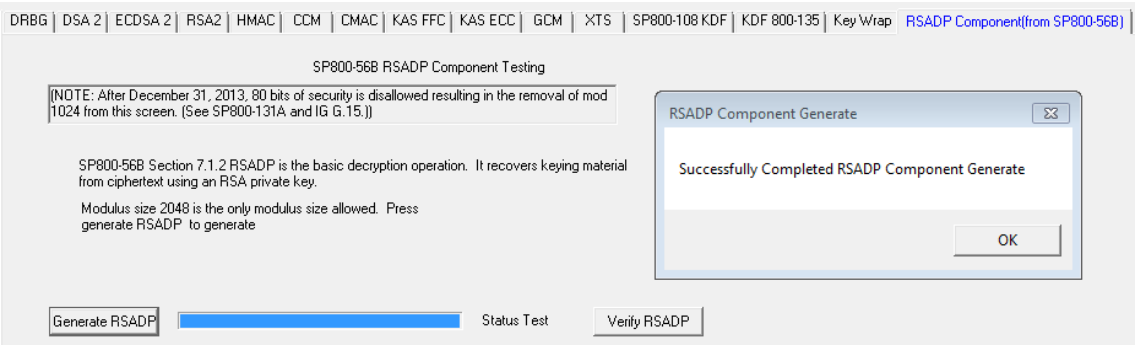
On the other hand, given that the TOE can act as sender and recipient for the RSA-based key establishment scheme that meets SP-800-56B, the Lab considers that this test cannot be covered by a CAVP certificate, due to the CAVS tool (19.2) only supports RSADP (recipient). So, in order to test this requirement the Lab has carried out the following test:

5.3.1. Test 1

5.3.1.1. 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.

The plaintexts used have been obtained from the NIST CAVS tool (19.2) with the following configuration:



Based-on this configuration the following input vectors are generated (30 vectors of 2048 bits length):

```
# CAVS 19.2
# "RSADP Component (from SP800-56B)" information for "testing"
# Combination tested: Mod Size 2048

[mod = 2048]

COUNT = 0
c = f5f687c71a18acf8eeac04f13a625006544e2111fb3fdf67824c100b31afd722e8d7bba705b8aa60

COUNT = 1
c = 8f099cd09d6915d2890b415ef6d5f83b9d67eca1e65bde9db9cb12b6911400d1d3da4c525f4df884

COUNT = 2
c = 89bee0ed59890ef3da8daa6104f603e129d7ab4620c1c3c2a3937859f693ce95c9be9d7c605dd8bf

COUNT = 3
c = fc4bc5d324f4adb9dec38986948bb4bc11faf39ae66ffb920c0b33fdcc0bbc0edc55af04cb50a5a4

COUNT = 4
c = 8e94e6b3346759d1aeb965e7004a5956113a65ddb9cc52751b696f7935e8cf0ab50cb1b20e4e4047
```

The OpenSSL FIPS Object Module 2.0.10 (*fipscanister.o* file) and OpenSSL 1.0.1 version are installed in the following operational environment:

- OS: Ubuntu 10.04 (32 bits)
- Processor: Intel Core i5 with AES-NI

```
eval@openssl-fips:~$ openssl-fips version
OpenSSL 1.0.1r-fips  28 Jan 2016
```

For the installation of the OpenSSL FIPS Object Module 2.0.10, hereinafter called OpenSSL FIPS, all the steps defined in the "User Guide for the OpenSSL FIPS Object module v2.0" document have been followed.

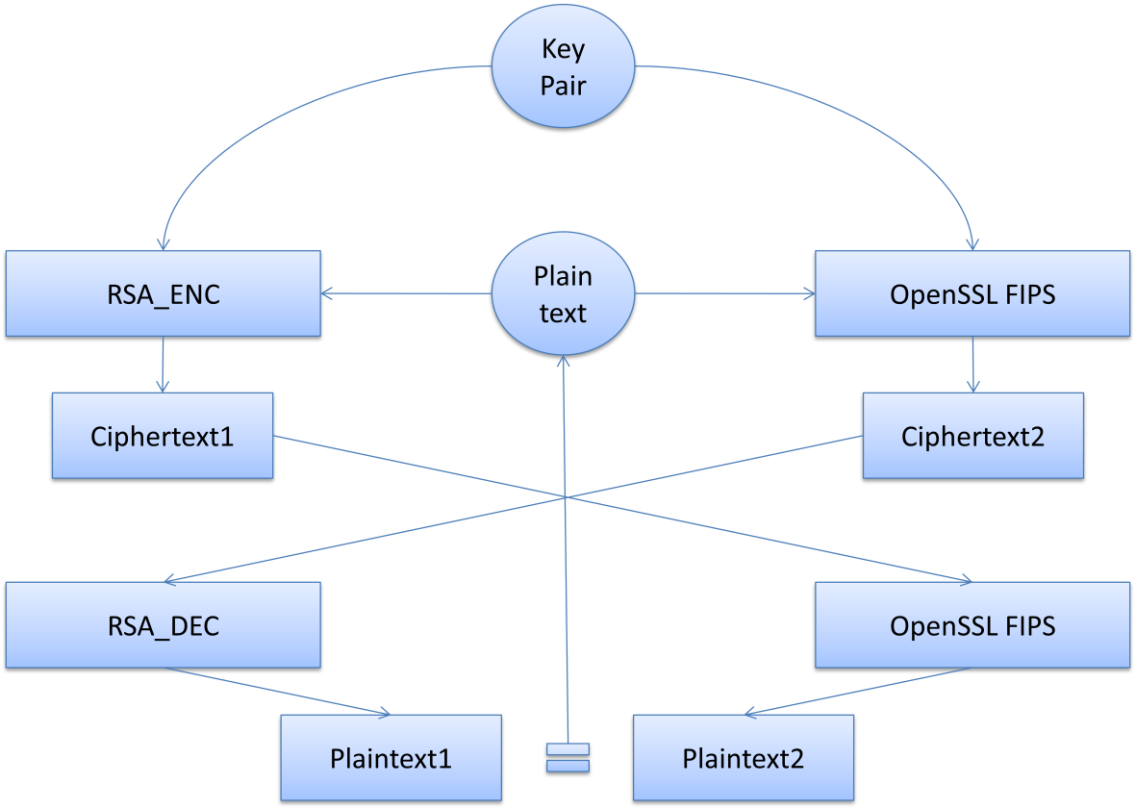
The Lab has developed two different SWs: RSA_ENC.cpp, which receives a plaintext as input and generates a ciphertext; and RSA_DEC.cpp, which receives a ciphertext as input and decrypts it to obtain the plaintext. These SWs have been developed based on the public APIs

for W10 and WS2012R2 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 (plaintexts) and encrypts them using a valid key pair (public and private key). As result of this execution RSA_ENC.cpp generates a set of ciphertexts. On the other hand, the evaluator executes OpenSSL FIPS to generate a new set of ciphertexts using the same plaintexts and key pair.

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



The following screen shots show the RSA_DEC.cpp source code, where the main function is *BCryptDecrypt*:

```
BCRYPT_ALG_HANDLE       hRSAAlg = NULL;
BCRYPT_KEY_HANDLE       hPriv = NULL;
NTSTATUS                status = STATUS_UNSUCCESSFUL;
DWORD
    c_length = 0,
    k_length = 0;
PBYTE
    pC = NULL,
    pK = NULL;

BYTE pubExp[] = {...};

BYTE mod[] = {...};

BYTE p1[] = {...};

BYTE p2[] = {...};

struct {
    BCRYPT_RSAKEY_BLOB header;
    UCHAR publicExponent[3];
    UCHAR modulus[256];
    UCHAR prime1[128];
    UCHAR prime2[128];
}pPriv;

pPriv.header.Magic = BCRYPT_RSAPRIVATE_MAGIC;
pPriv.header.BitLength = 2048;
pPriv.header.cbPublicExp = 3;
pPriv.header.cbModulus = 256;
pPriv.header.cbPrime1 = 128;
pPriv.header.cbPrime2 = 128;

CopyMemory(pPriv.publicExponent, pubExp, 3);
CopyMemory(pPriv.modulus, mod, 256);
CopyMemory(pPriv.prime1, p1, 128);
CopyMemory(pPriv.prime2, p2, 128);



// Open an algorithm handle.
if (!NT_SUCCESS(status = BCryptOpenAlgorithmProvider(
    &hRSAAlg,
    BCRYPT_RSA_ALGORITHM,
    MS_PRIMITIVE_PROVIDER,
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptOpenAlgorithmProvider\n", status);
    goto Cleanup;
}

//Import private key
if (!NT_SUCCESS(status = BCryptImportKeyPair(
    hRSAAlg,
    NULL,
    BCRYPT_RSAPRIVATE_BLOB,
    &hPriv,
    (PUCHAR)&pPriv,
    sizeof(pPriv),
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptImportKey\n", status);
    goto Cleanup;
}
```

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

pK = (PBYTE)HeapAlloc(GetProcessHeap(), 0, k_length);
if (NULL == pK)
{
    wprintf(L"**** memory allocation failed\n");
    goto Cleanup;
}

// 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;
}

cout << "Plaintext: " << endl;
PrintBytes((BYTE *)pK, k_length);
```

On the other hand, the following screen shot shows the main function in RSA_ENC.cpp source code, which it is called *BCryptEncrypt*:

```
BCRYPT_ALG_HANDLE       hRSAAlg = NULL;
BCRYPT_KEY_HANDLE       hPub = NULL;
NTSTATUS                status = STATUS_UNSUCCESSFUL;
DWORD
    c_length = 0,
    k_length = 0;
PBYTE
    pC = NULL,
    pK = NULL;

BYTE pubExp[] = {...};

BYTE mod[] = {...};

BYTE p1[] = {...};

BYTE p2[] = {...};


struct {
    BCRYPT_RSAKEY_BLOB header;
    UCHAR publicExponent[3];
    UCHAR modulus[256];
}pPub;

pPub.header.Magic = BCRYPT_RSAPUBLIC_MAGIC;
pPub.header.BitLength = 2048;
pPub.header.cbPublicExp = 3;
pPub.header.cbModulus = 256;
pPub.header.cbPrime1 = 0;
pPub.header.cbPrime2 = 0;

CopyMemory(pPub.publicExponent, pubExp, 3);
CopyMemory(pPub.modulus, mod, 256);
```

```
// Open an algorithm handle.
if (!NT_SUCCESS(status = BCryptOpenAlgorithmProvider(
    &hRSAAlg,
    BCRYPT_RSA_ALGORITHM,
    MS_PRIMITIVE_PROVIDER,
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptOpenAlgorithmProvider\n", status);
    goto Cleanup;
}

//Import public key
if (!NT_SUCCESS(status = BCryptImportKeyPair(
    hRSAAlg,
    NULL,
    BCRYPT_RSAPUBLIC_BLOB,
    &hPub,
    (PUCHAR)&pPub,
    (sizeof(pPub)-1),
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptpublicKey\n", status);
    goto Cleanup;
}

  // Get ciphertext length
  if (!NT_SUCCESS(status = BCryptEncrypt(
      hPub,
      pK,
      k_length,
      NULL,
      NULL,
      0,
      NULL,
      0,
      &c_length,
      BCRYPT_PAD_PKCS1)))
  {
      wprintf(L"**** Error 0x%x returned by BCryptEncrypt\n", status);
      goto Cleanup;
  }

  pC = (PBYTE)HeapAlloc(GetProcessHeap(), 0, c_length);
  if (NULL == pC)
  {
      wprintf(L"**** memory allocation failed\n");
      goto Cleanup;
  }

  // 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;
  }

  cout << "Ciphertext: " << endl;
  PrintBytes((BYTE *)pC, c_length);
```

5.3.1.2. Procedure

1. Generate the input vectors.
2. Check that the input vectors are correctly generated.
3. Put the input vectors in the same path of the RSA_ENC.exe.
4. Execute RSA_ENC.exe to obtain the set of ciphertexts.
5. Put the input vectors in the same path where OpenSSL FIPS is stored.

6. Execute the following command to obtain one ciphertext using one plaintext.

   openssl rsautl -encrypt -pkcs -pubin -inkey pub.pem -in plaintext.bin -out ciphertext.bin

7. Copy the set of ciphertexts generated by RSA_ENC.exe in the same path where OpenSSL FIPS is stored.

8. Execute the following command to obtain the plaintext using one ciphertext.

   openssl rsautl -decrypt -inkey priv.pem -in ciphertext.bin -out plaintext.bin

9. Copy the set of ciphertexts generated by OpenSSL FIPS in the same path of the RSA_DEC.exe.

10. Execute RSA_DEC.exe to obtain the set of plaintexts.

11. Compare the set of plaintexts generated by RSA_DEC.exe with the one generated by OpenSSL FIPS.

5.3.1.3. Results

The test has been performed in the following platforms:

- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.

Following an example of the obtained results is shown. Given the following plaintext,

k=f5f687c71a18acf8eeac04f13a625006544e2111fb3fdf67824c100b31afd722e8d7bba705b8aa6088663
e1175e23fef7a336857d2b96ec069bf3eef414f866a144a782e3bb8cba4ce5c02d20d8bf0cefe3f7c795868a
269a082ba27926cfd9bc27845c483bf14ad08b7d89cc9076d8c2899e4dd55f1834b6fbe0c1decaf439e3a88
aad1a6758f617e9e48d772fbbe0f406a28ed91ff194c1d9bbd3716c05f5cc4a32fbd6d8343a9773adbebd01
2e9418ed9d110d3386d55f44054dfda465cbca8f15df19d91386e3db21840abf2f8095a0f66ade444864f3a
d6597cfeea3fa52d99bbd301467796162f883ecb43f24eeb346f2e8dcc6c20388455754b531201

the evaluator has obtained the following ciphertext using OpenSSL FIPS:

```
eval@openssl fips:~$ env OPENSSL_FIPS=1 openssl-fips rsautl -encrypt -pkcs -pubin -inkey pub.pem -in plaintext0.bin | hexdump -C
00000000  36 0c 94 93 ef 4c c1 52  54 18 8f 17 b8 56 5d 6b  |6....L.RT....V]k|
00000010  d0 82 b9 5d 1e c8 1b ce  6d 4a d8 10 7a 37 3b 4b  |...]....mJ..z7;K|
00000020  0e 76 d0 ad ee e3 6d 8a  72 26 99 fa 3f ab 46 a9  |.v....m.r&..?.F.|
00000030  65 a1 23 c7 fa b3 a7 20  4e ad 1b 24 04 88 95 3c  |e.#.... N..$...<|
00000040  7c 1d a5 f6 b2 25 db 1c  4e 63 1c 57 bf b4 10 ac  ||....%..Nc.W....|
00000050  71 25 d3 fe da 5a 95 61  56 b2 b5 c0 ff 4a 44 31  |q%...Z.aV....JD1|
00000060  40 62 43 7f 82 39 1d 4f  a8 e5 29 b6 3f 81 f1 42  |@bC..9.O..).?..B|
00000070  a1 ca bd 3a d5 93 ea 38  e3 4d 6c ad 23 8a 10 fc  |...:...8.Ml.#...|
00000080  1e 49 9e 9f 0c d6 e7 4d  8e 7c 64 5a 2c 7d d3 d7  |.I.....M.|dZ,}..|
00000090  a6 e4 e8 e2 d8 7c fa 29  2b e0 ec 9b 07 09 b6 fc  |.....|.)+.......|
000000a0  0c 67 19 8d f4 07 40 b3  4d 9b 8b e2 0b 41 48 3a  |.g....@.M....AH:|
000000b0  b3 cf d4 07 a1 4e d2 73  3e 03 d7 05 f9 3a c3 2e  |.....N.s>....:..|
000000c0  71 03 da 0e 0e 17 4e 26  c9 f8 93 41 f8 47 16 e5  |q.....N&...A.G..|
000000d0  cf 86 1b cd 15 14 d2 6b  2c da 3d 37 91 44 ed 9b  |.......k,.=7.D..|
000000e0  e3 72 05 38 99 33 fe cb  1f 9a b5 7c dc ba 9a 3d  |.r.8.3.....|...=|
000000f0  90 a4 a1 63 c7 2d 5e a1  60 5c e3 52 dc 2e 4e 6f  |...c.-^.`\.R..No|
00000100
```

and using the SW RSA_ENC.exe:

```
ciphertext0.bin
0x1b, 0x0a, 0xf6, 0x46, 0x54, 0xb4, 0x83, 0x2a, 0x25, 0x24,
0x82, 0x6f, 0xdf, 0x0b, 0x7c, 0x14, 0x9f, 0xab, 0xa0, 0xee,
0x18, 0x5c, 0xd2, 0x80, 0x5b, 0x33, 0x2e, 0x81, 0x19, 0x32,
0xba, 0x3f, 0x1d, 0xa7, 0xa9, 0x88, 0xd2, 0xcb, 0xe9, 0xe6,
0xc2, 0xb5, 0xa0, 0xd5, 0x31, 0x6d, 0x82, 0x3b, 0xb0, 0xb9,
0xfb, 0x1b, 0x9e, 0x04, 0x76, 0xa8, 0x36, 0x21, 0x02, 0x22,
0x29, 0xe3, 0x4f, 0x1c, 0xcd, 0xb1, 0x5f, 0x2d, 0x07, 0x58,
0x04, 0x5d, 0x01, 0x90, 0x35, 0xb7, 0xf7, 0x07, 0xdd, 0x9a,
0xc0, 0xc1, 0x41, 0x26, 0x9f, 0x92, 0xfc, 0xda, 0xcd, 0x18,
0x87, 0x70, 0x7a, 0x34, 0x39, 0x25, 0x58, 0xd9, 0x0f, 0xe0,
0x22, 0x76, 0xa7, 0x33, 0xde, 0x25, 0xea, 0xca, 0xf6, 0x98,
0x27, 0xaa, 0x58, 0xe9, 0x8f, 0x5b, 0x0c, 0x55, 0xb5, 0x4f,
0x48, 0x57, 0xe8, 0x7f, 0x3c, 0xb1, 0x44, 0xf3, 0xbe, 0xf3,
0x8a, 0x7b, 0xaa, 0x62, 0x6b, 0x61, 0xeb, 0xd4, 0x0a, 0x3b,
0x30, 0x73, 0xbd, 0xd3, 0xb0, 0x60, 0xef, 0xce, 0x44, 0xad,
0x8f, 0x1f, 0xb3, 0x27, 0x4f, 0xa6, 0xe3, 0x63, 0x76, 0x65,
0x1c, 0xf7, 0xee, 0x8a, 0xc8, 0x6d, 0x4f, 0xba, 0x22, 0x0a,
0x5f, 0x80, 0x6a, 0xa8, 0x6a, 0x99, 0x30, 0xf6, 0xa0, 0x74,
0xcd, 0xe4, 0xa4, 0x03, 0x77, 0xb6, 0x9c, 0xde, 0x0b, 0xdb,
0xf3, 0xba, 0x25, 0xb1, 0x60, 0x0c, 0x1d, 0xfa, 0x41, 0x76,
0xd3, 0x7e, 0xcb, 0xf3, 0x72, 0xc4, 0x01, 0x62, 0x65, 0xd7,
0x9a, 0x47, 0x34, 0x20, 0xcb, 0xe4, 0x1d, 0xfa, 0x07, 0x4d,
0x74, 0x35, 0xdf, 0xff, 0x2f, 0xda, 0x59, 0x12, 0x8f, 0xee,
0x25, 0xd9, 0x49, 0xa6, 0x44, 0xc8, 0x58, 0x79, 0xee, 0xea,
0x47, 0x5f, 0x54, 0xff, 0x38, 0x3b, 0x85, 0xce, 0xd6, 0x14,
0xf2, 0x6f, 0x4f, 0xac, 0x66, 0x23,
```

After that, the ciphertexts are swapped, and the evaluator has obtained the following plaintexts using the OpenSSL FIPS:

```
eval@openssl1fips:~$ env OPENSSL_FIPS=1 openssl-fips rsautl -decrypt -inkey priv.pem -in ciphertextWin0.bin | hexdump -C
00000000  f5 f6 87 c7 1a 18 ac f8  ee ac 04 f1 3a 62 50 06  |............:bP.|
00000010  54 4e 21 11 fb 3f df 67  82 4c 10 0b 31 af d7 22  |TN!..?.g.L..1..".|
00000020  e8 d7 bb a7 05 b8 aa 60  88 66 3e 11 75 e2 3f ef  |.......`.f>.u.?.|
00000030  7a 33 68 57 d2 b9 6e c0  69 bf 3e ef 41 4f 86 6a  |z3hW..n.i.>.AO.j|
00000040  14 4a 78 2e 3b b8 cb a4  ce 5c 02 d2 0d 8b f0 ce  |.Jx.;....\......|
00000050  fe 3f 7c 79 58 68 a2 69  a0 82 ba 27 92 6c fd 9b  |.?|yXh.i...'.l..|
00000060  c2 78 45 c4 83 bf 14 ad  08 b7 d8 9c c9 07 6d 8c  |.xE...........m.|
00000070  28 99 e4 dd 55 f1 83 4b  6f be 0c 1d ec af 43 9e  |(...U..Ko.....C.|
00000080  3a 88 aa d1 a6 75 8f 61  7e 9e 48 d7 72 fb be 0f  |:....u.a~.H.r...|
00000090  40 6a 28 ed 91 ff 19 4c  1d 9b bd 37 16 c0 5f 5c  |@j(....L...7.._\|
000000a0  c4 a3 2f bd 6d 83 43 a9  77 3a db eb d0 12 e9 41  |../.m.C.w:.....A|
000000b0  8e d9 d1 10 d3 38 6d 55  f4 40 54 df da 46 5c bc  |.....8mU.@T..F\.|
```

and using the SW RSA_DEC.exe:

```
Plaintext 0:
0xf5, 0xf6, 0x87, 0xc7, 0x1a, 0x18, 0xac, 0xf8, 0xee, 0xac,
0x04, 0xf1, 0x3a, 0x62, 0x50, 0x06, 0x54, 0x4e, 0x21, 0x11,
0xfb, 0x3f, 0xdf, 0x67, 0x82, 0x4c, 0x10, 0x0b, 0x31, 0xaf,
0xd7, 0x22, 0xe8, 0xd7, 0xbb, 0xa7, 0x05, 0xb8, 0xaa, 0x60,
0x88, 0x66, 0x3e, 0x11, 0x75, 0xe2, 0x3f, 0xef, 0x7a, 0x33,
0x68, 0x57, 0xd2, 0xb9, 0x6e, 0xc0, 0x69, 0xbf, 0x3e, 0xef,
0x41, 0x4f, 0x86, 0x6a, 0x14, 0x4a, 0x78, 0x2e, 0x3b, 0xb8,
0xcb, 0xa4, 0xce, 0x5c, 0x02, 0xd2, 0x0d, 0x8b, 0xf0, 0xce,
0xfe, 0x3f, 0x7c, 0x79, 0x58, 0x68, 0xa2, 0x69, 0xa0, 0x82,
0xba, 0x27, 0x92, 0x6c, 0xfd, 0x9b, 0xc2, 0x78, 0x45, 0xc4,
0x83, 0xbf, 0x14, 0xad, 0x08, 0xb7, 0xd8, 0x9c, 0xc9, 0x07,
0x6d, 0x8c, 0x28, 0x99, 0xe4, 0xdd, 0x55, 0xf1, 0x83, 0x4b,
0x6f, 0xbe, 0x0c, 0x1d, 0xec, 0xaf, 0x43, 0x9e, 0x3a, 0x88,
0xaa, 0xd1, 0xa6, 0x75, 0x8f, 0x61, 0x7e, 0x9e, 0x48, 0xd7,
0x72, 0xfb, 0xbe, 0x0f, 0x40, 0x6a, 0x28, 0xed, 0x91, 0xff,
0x19, 0x4c, 0x1d, 0x9b, 0xbd, 0x37, 0x16, 0xc0, 0x5f, 0x5c,
0xc4, 0xa3, 0x2f, 0xbd, 0x6d, 0x83, 0x43, 0xa9, 0x77, 0x3a,
0xdb, 0xeb, 0xd0, 0x12, 0xe9, 0x41, 0x8e, 0xd9, 0xd1, 0x10,
0xd3, 0x38, 0x6d, 0x55, 0xf4, 0x40, 0x54, 0xdf, 0xda, 0x46
```

The obtained plaintexts are the same as the one provided by the CAVS tool.

5.3.1.4. Verdict

The evaluator considers that, the tests results obtained during the test activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the PASS verdict is assigned to the test activity.

5.4. Final Verdict

Due to all activities 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 this FCS_CKM.2.1 requirement.

6. FCS_CKM_EXT.3

6.1. Assurance activity

The evaluator will check to ensure the TSS lists each type of key material and its origin and storage location. The evaluator will verify that the TSS describes when each type of key material is cleared. For each software key clearing situation the evaluator will repeat the following test.

- ***Test 1:*** *The evaluator will utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key. Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator will perform the following steps for each key subject to clearing, including intermediate copies of keys that are persisted encrypted by the TOE:*

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

The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise. The evaluator will perform this test on all keys, including those persisted in encrypted form, to ensure intermediate copies are cleared.

6.2. Documentation review activity

6.2.1. 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.
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.
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 the SP 800-90 standards) as part of the generation process.

6.2.2. Verdict

The evaluator considers that the TSS identifies for the networking and DPAPI volatile keys when are generated, the purpose of its usage and how are zeroized when they are no longer needed using the RtlSecureZeroMemory internal function.

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

6.3. Test Activity

6.3.1. Test 1

6.3.1.1. Setup

A web server is available (IIS)

A web Client is available (iexplore)

A debugging environment is configured

6.3.1.2. Procedure

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 followed by a read-verify.

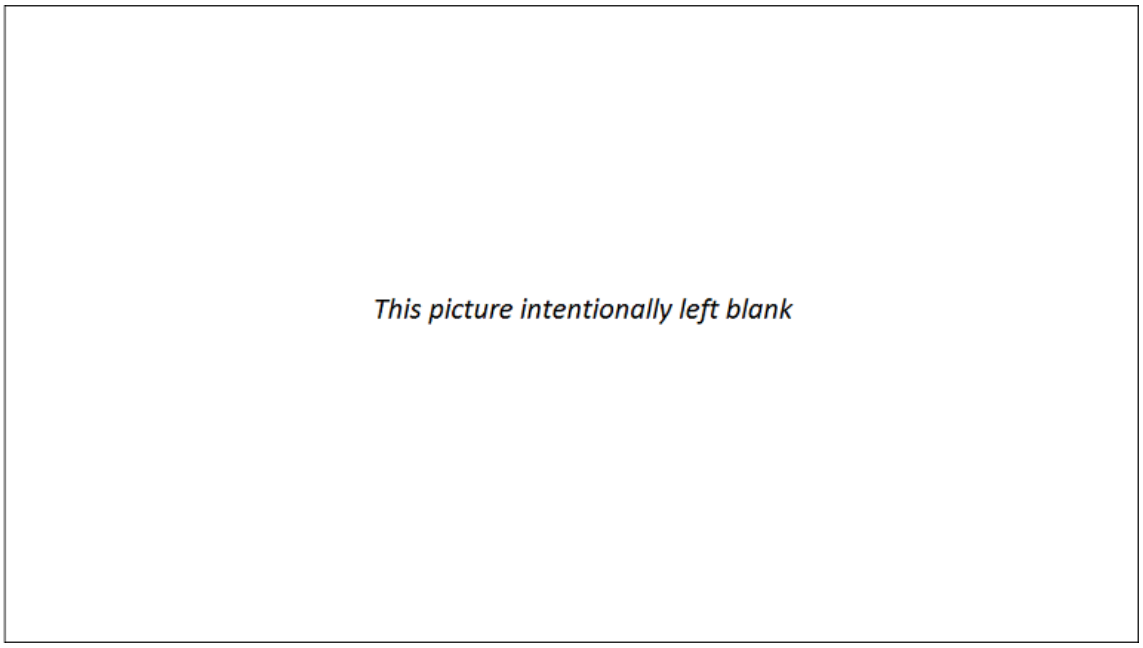
For volatile memory keys

1. Perform a cryptographic operation in debug mode.
2. Check how the key is generated.
3. Check the memory value of the key variable before performing cryptographic operation.
4. Perform the cryptographic operation and the corresponding zeroization mechanism
5. Check that the memory value of the key variable is overwritten with zeros
6. Dump the TOE memory
7. Search the memory value string (obtained in the step 2) within the dumped file.
8. Repeat the steps 1-6, using each hardware platform defined in section ***Evaluated Platforms.***

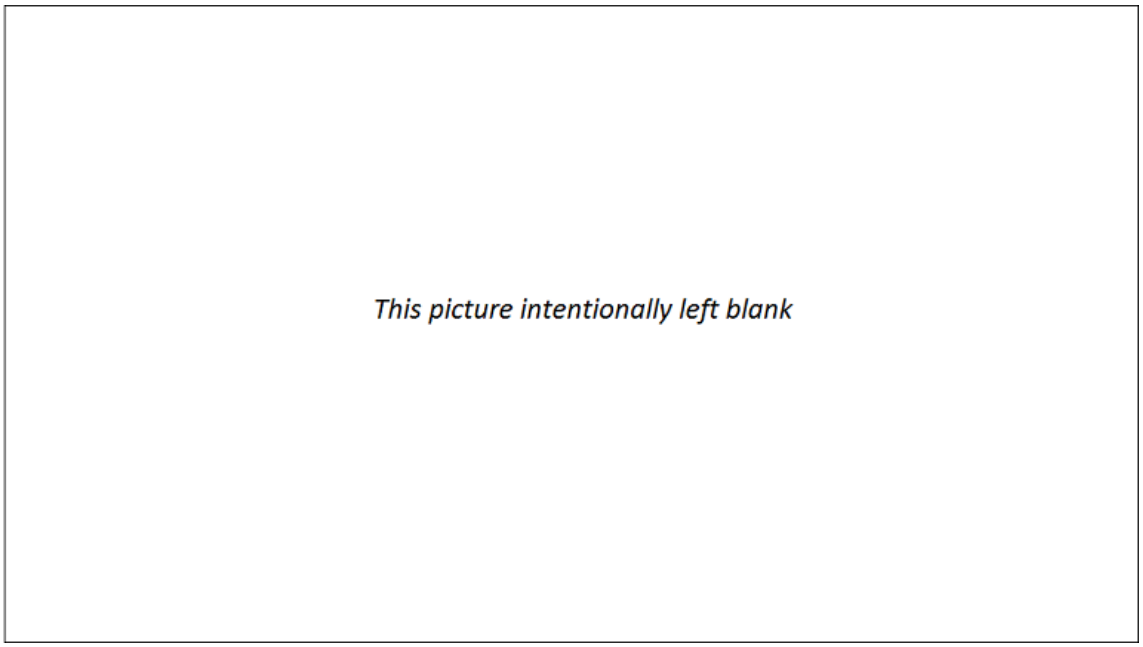
6.3.1.3. Results

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:

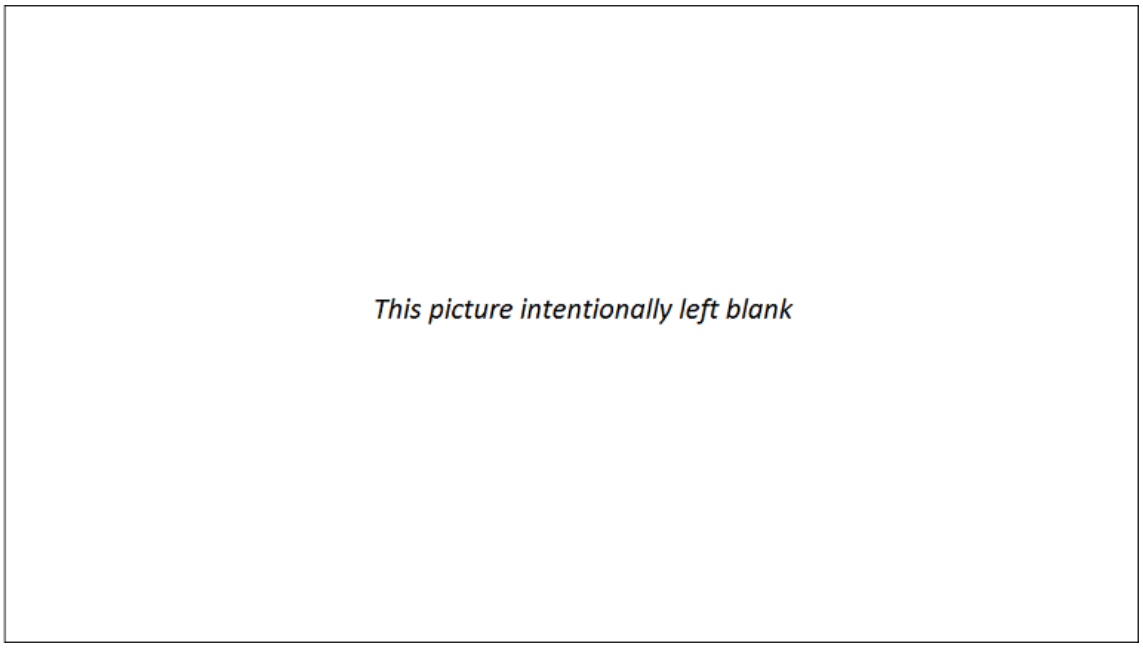
AES_KEY_ZEROIZE



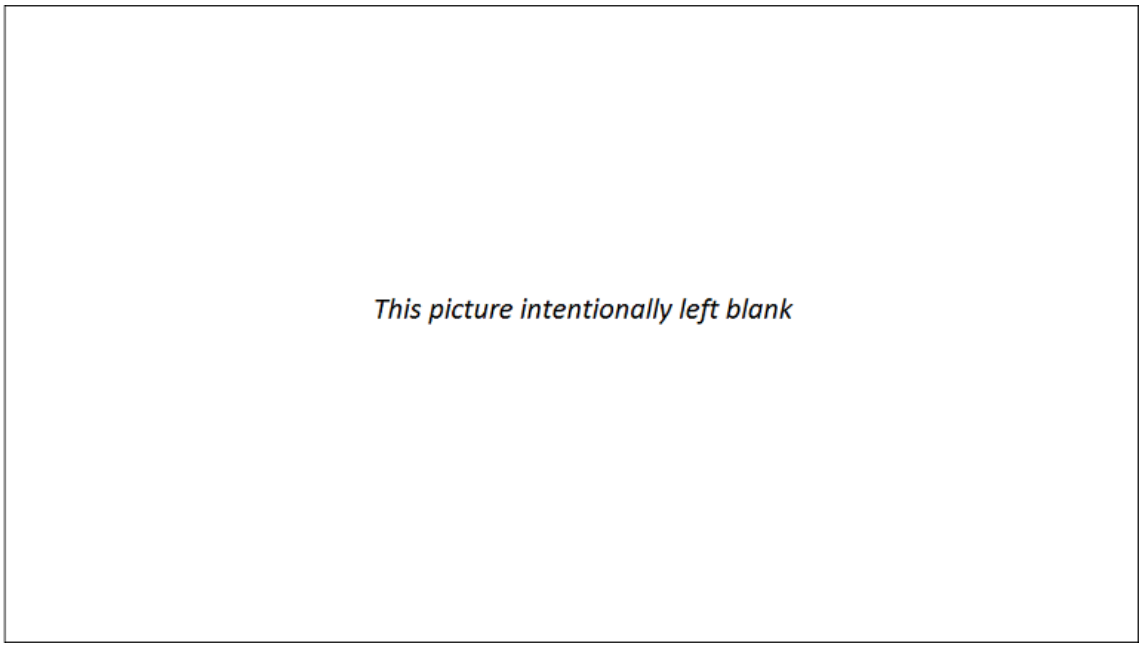
HMAC_ZEROIZE



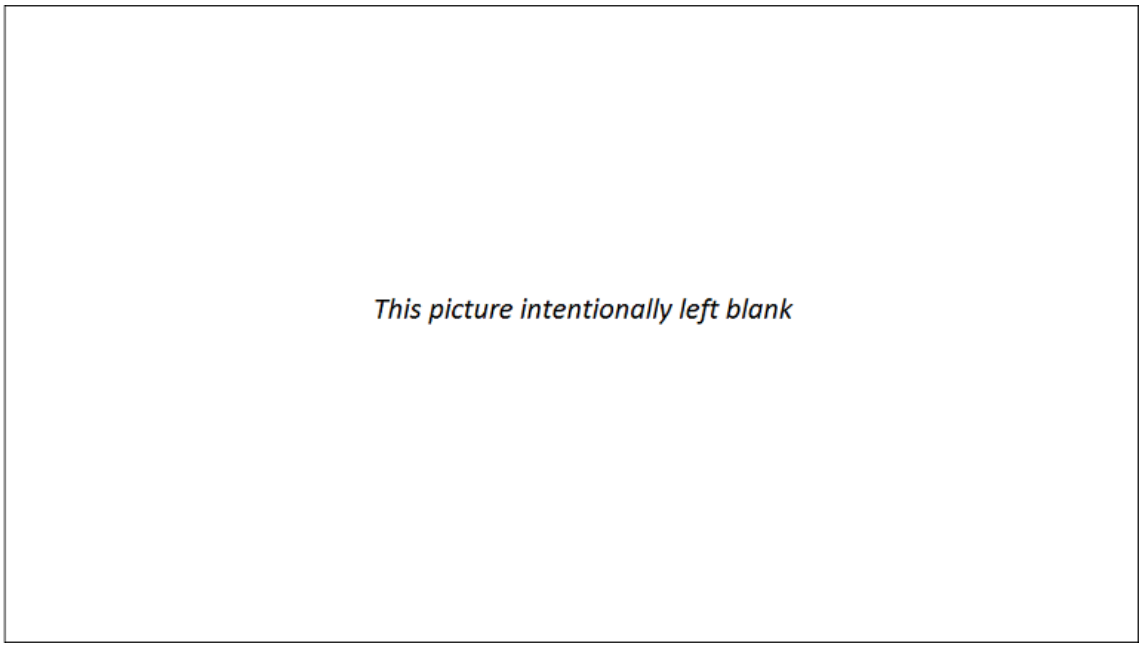
ECC_KEY_ZEROIZE



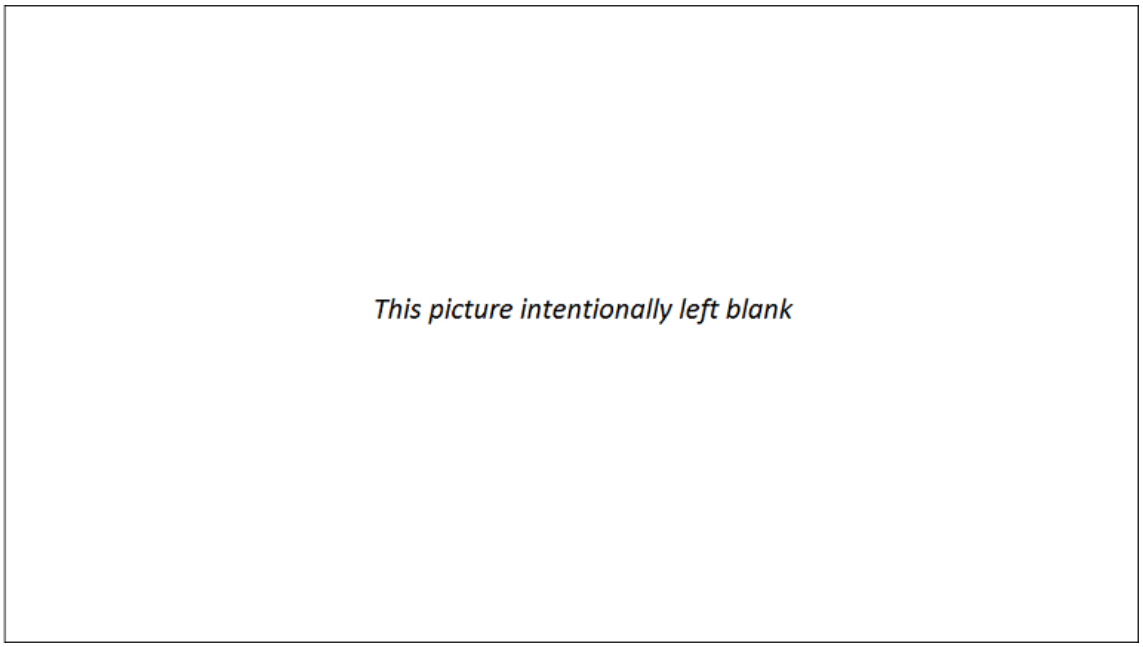
DH_KEY_ZEROIZE



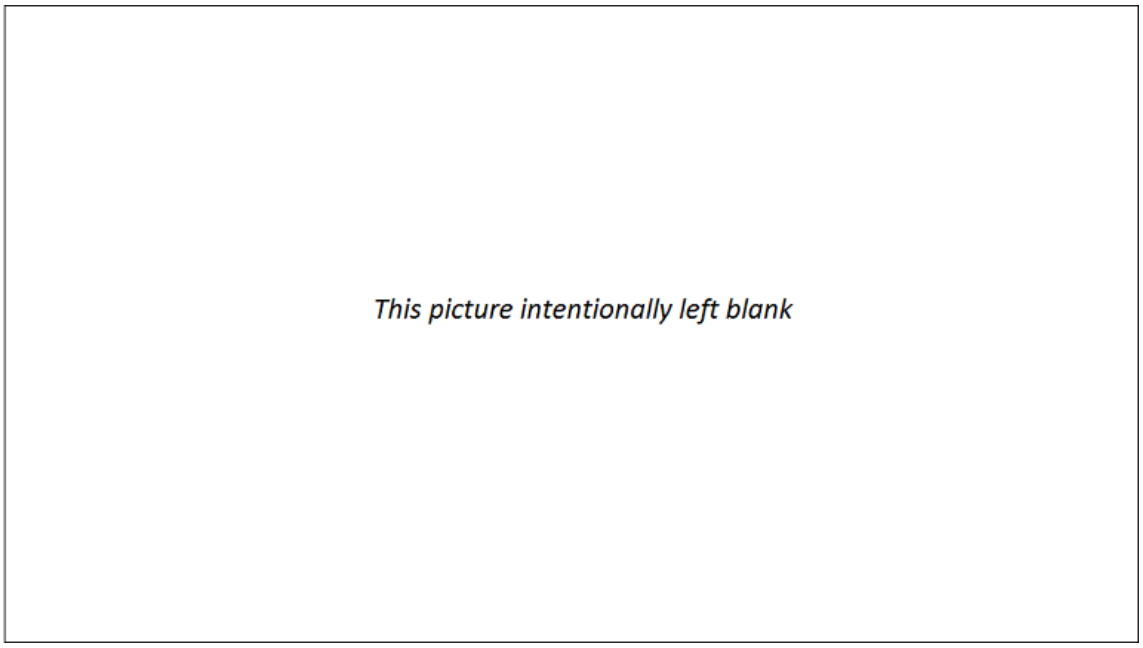
DSA_KEY_ZEROIZE



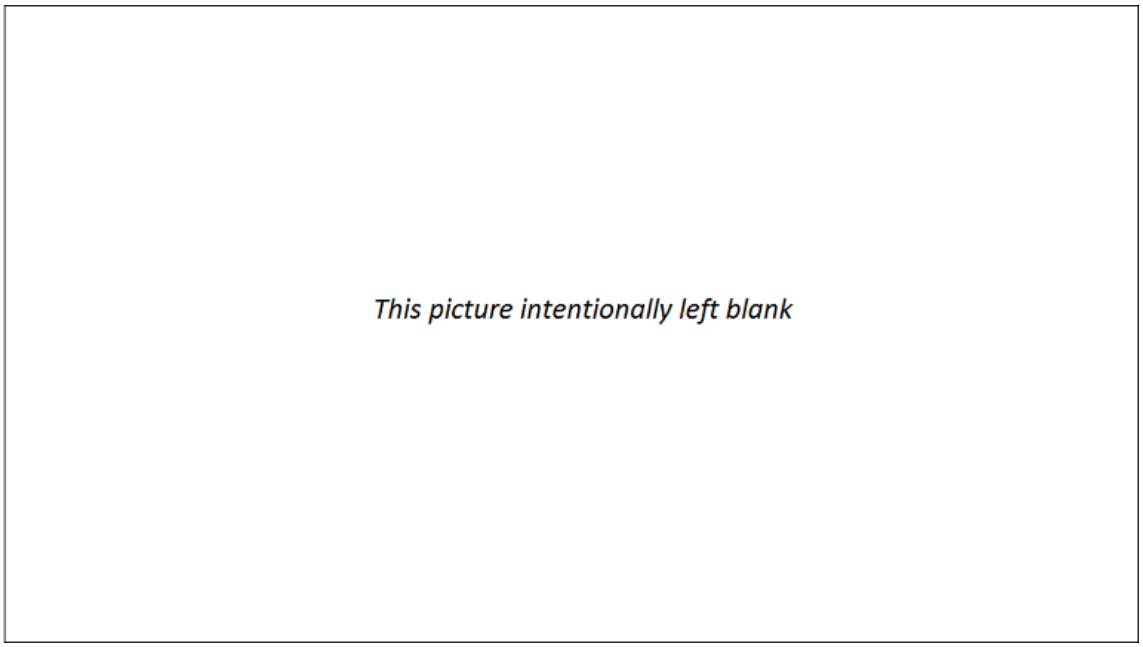
RSA_KEY_ZEROIZE

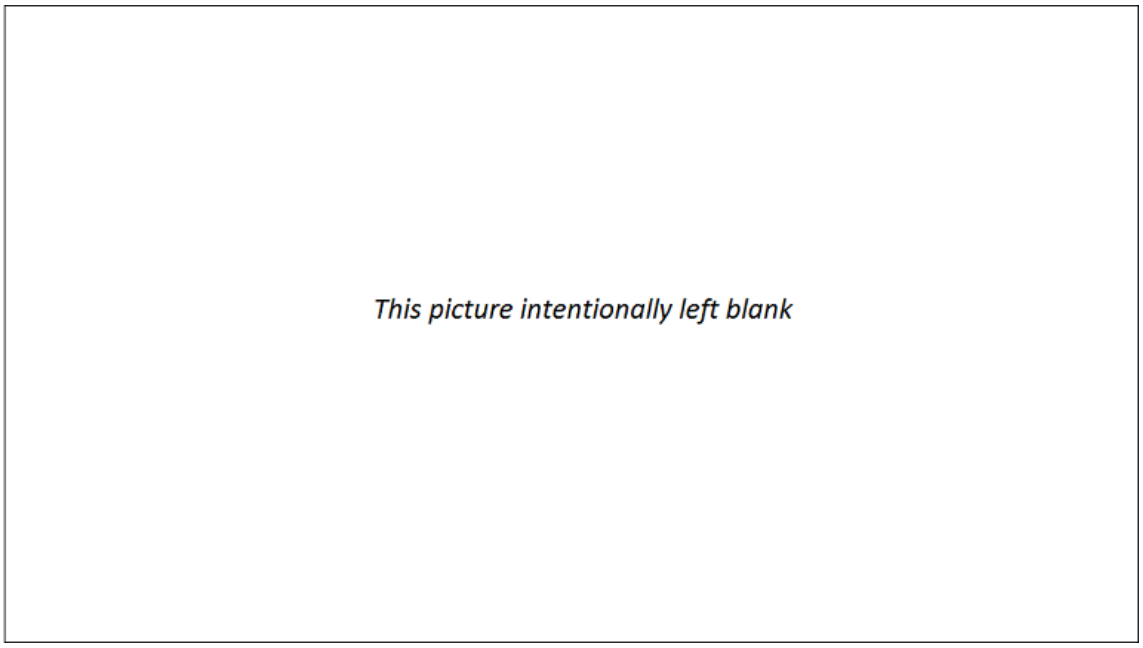


DBRG Seed zeroized



ZEROIZATION_FUNCTIONS:

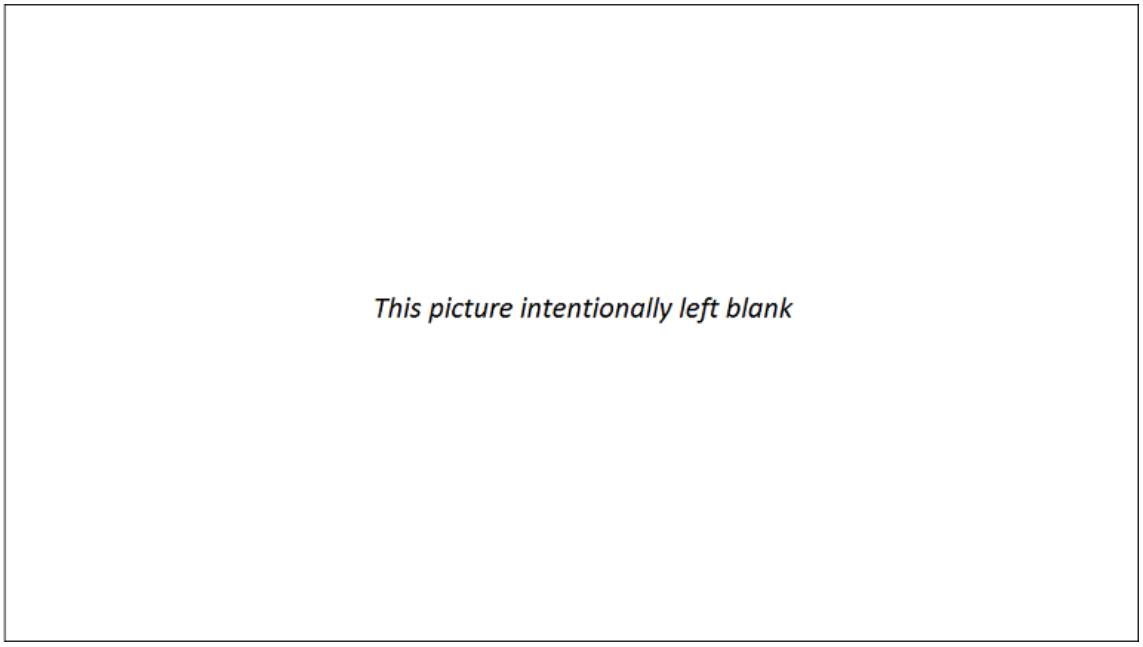




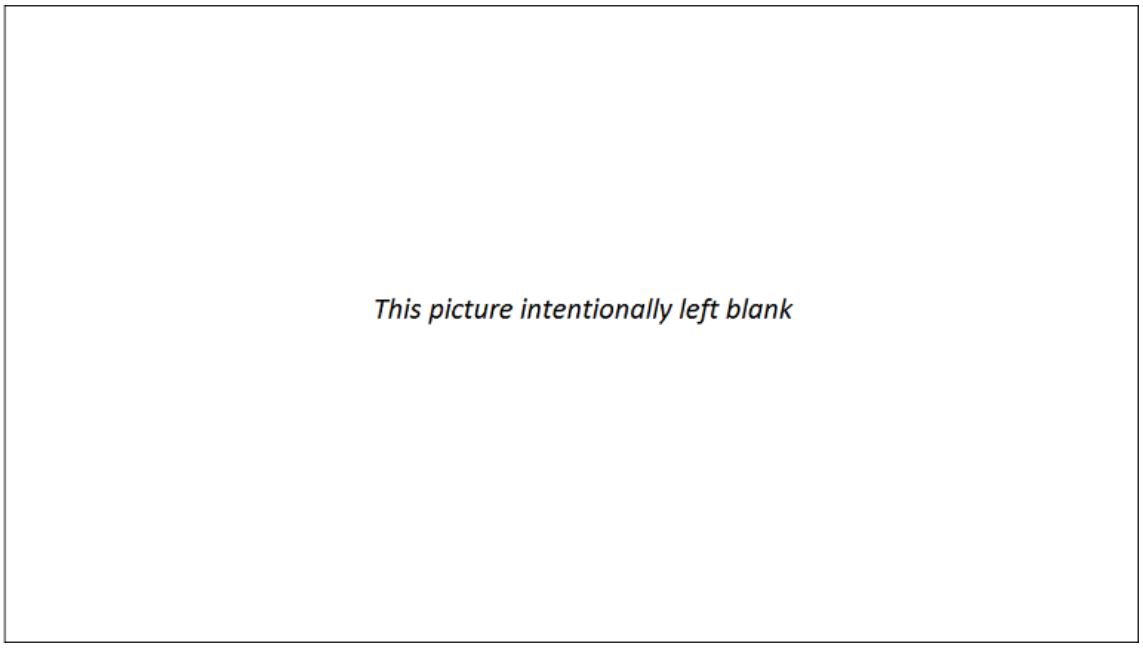
Source code review ("secure storage" keys)

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

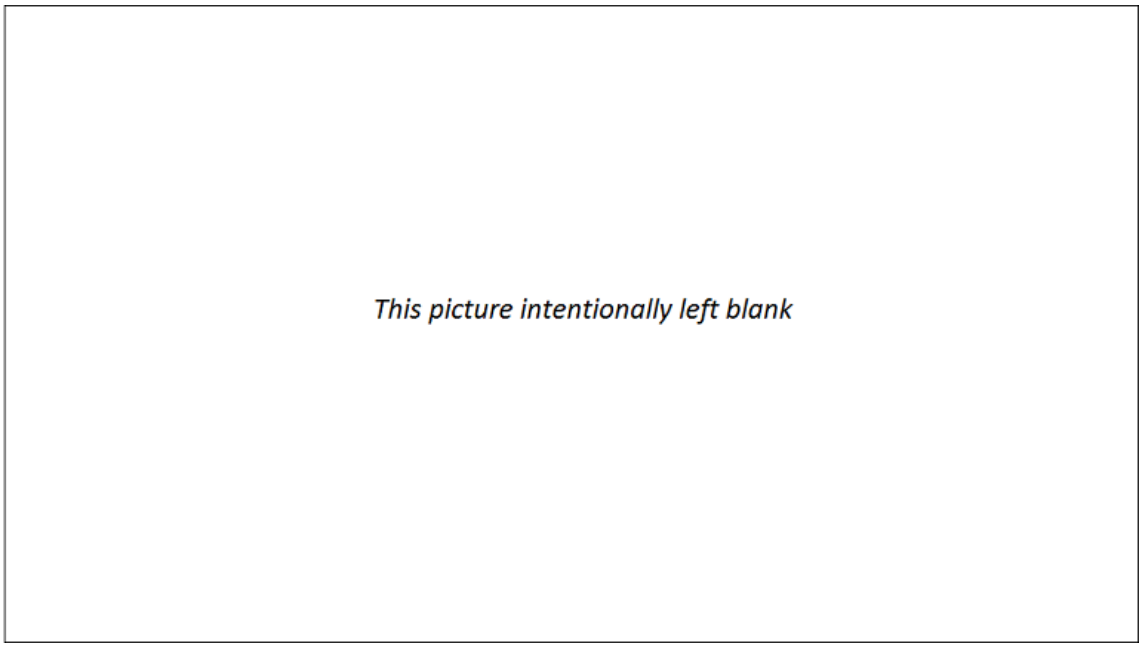
Zeroization of DPAPI AES key derived from User password



Zeroization of DPAPI Master Key (also called the Master Secret)



Zeroization of DPAPI AES key used to encrypt\decrypt DPAPI protected data

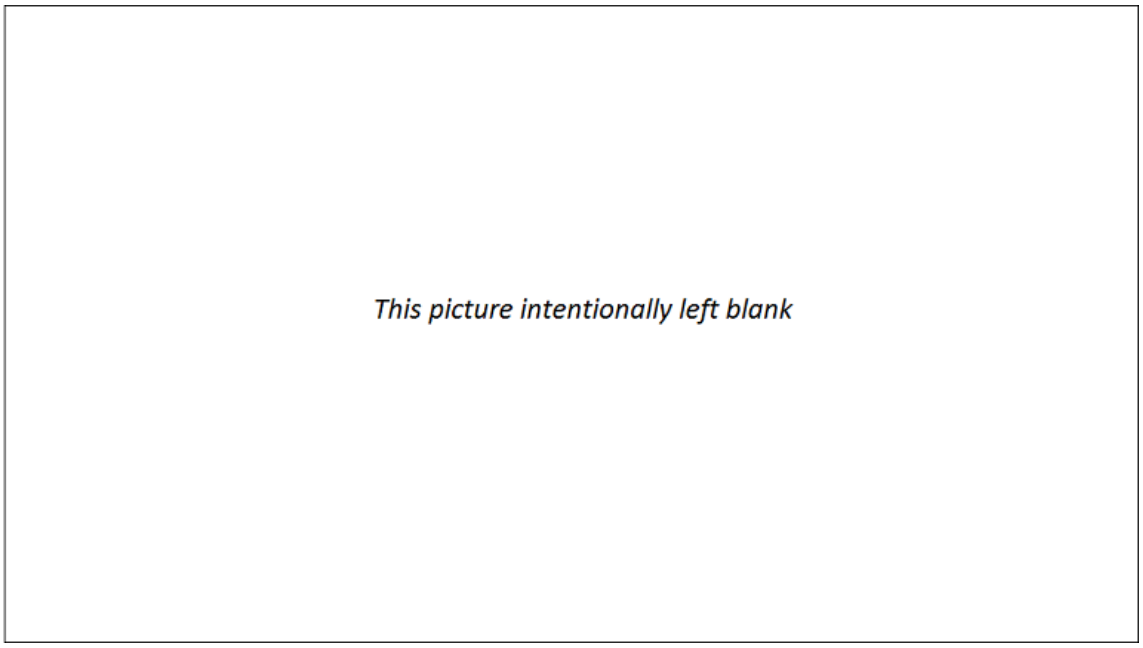


Source code review (DPAPI keys)

The evaluator has verified how the key zeroization functions are called in for each volatile key involved in the DPAPI, as it can be shown in the screenshots below:

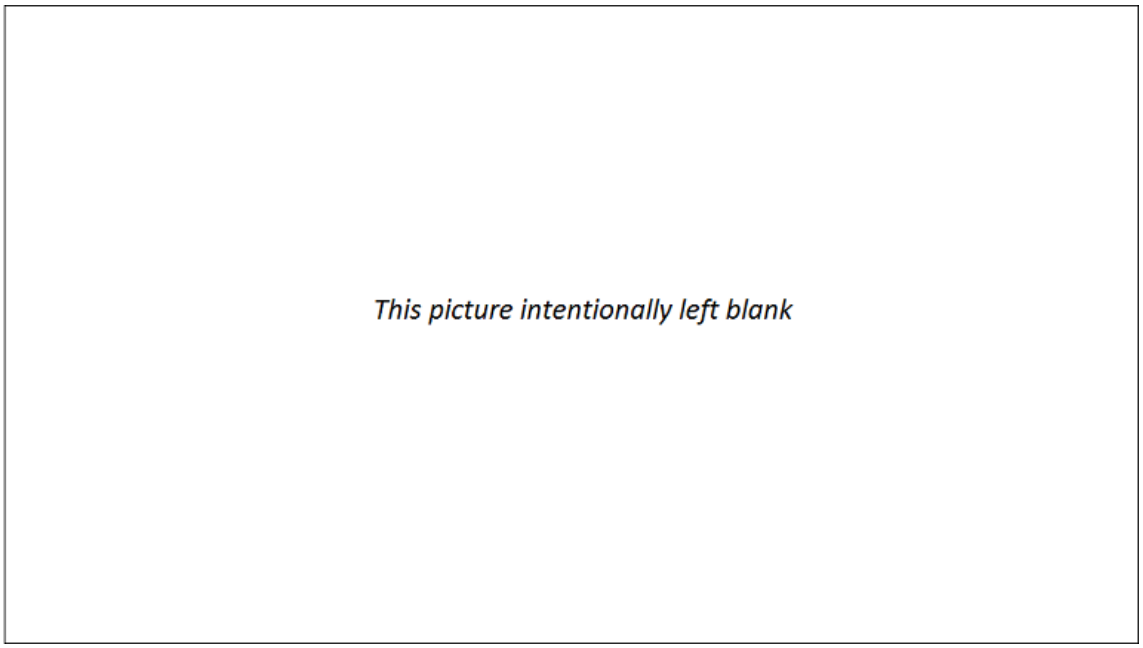
PBKDF2 HMAC key zeroization

The key used during the PBKDF2 HMAC process in the code is a buffer referred to as the rgbMKEncryptionKey. This keys is the length of an SHA1 hash (20 bytes). The following is a screen shot of when this key is zeroized:



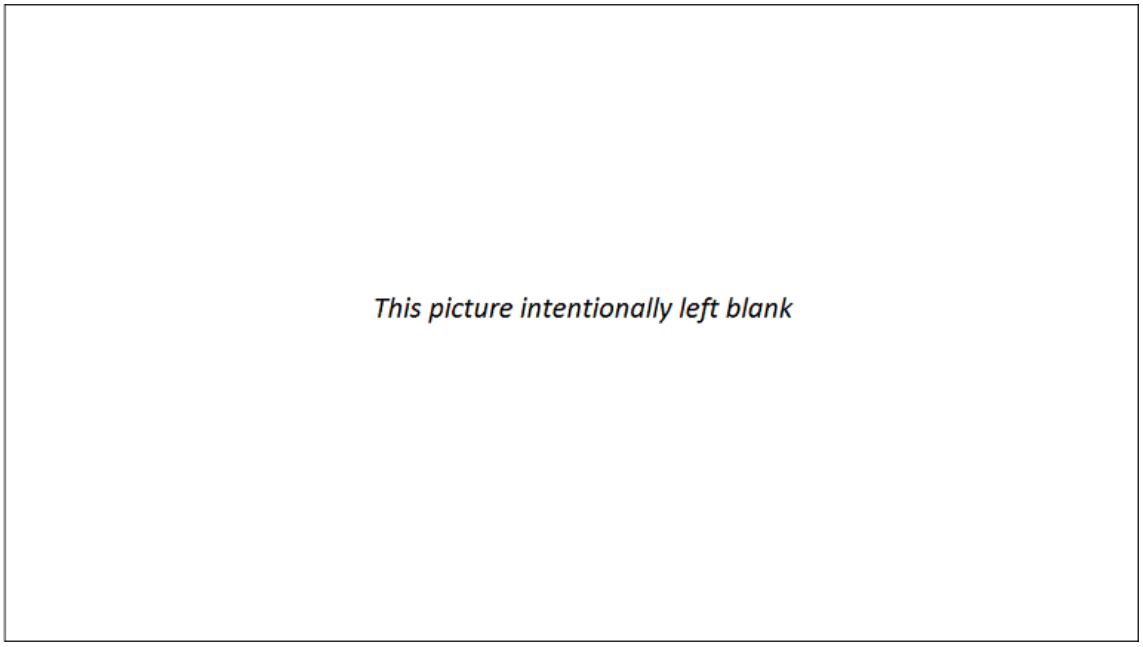
Zeroization of the symmetric key used to encrypt/decrypt the Master Key

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 is the screen shot of the decryption of the master key along with the destruction of the 256 bit AES key that is derived from the PBKDF2 process.



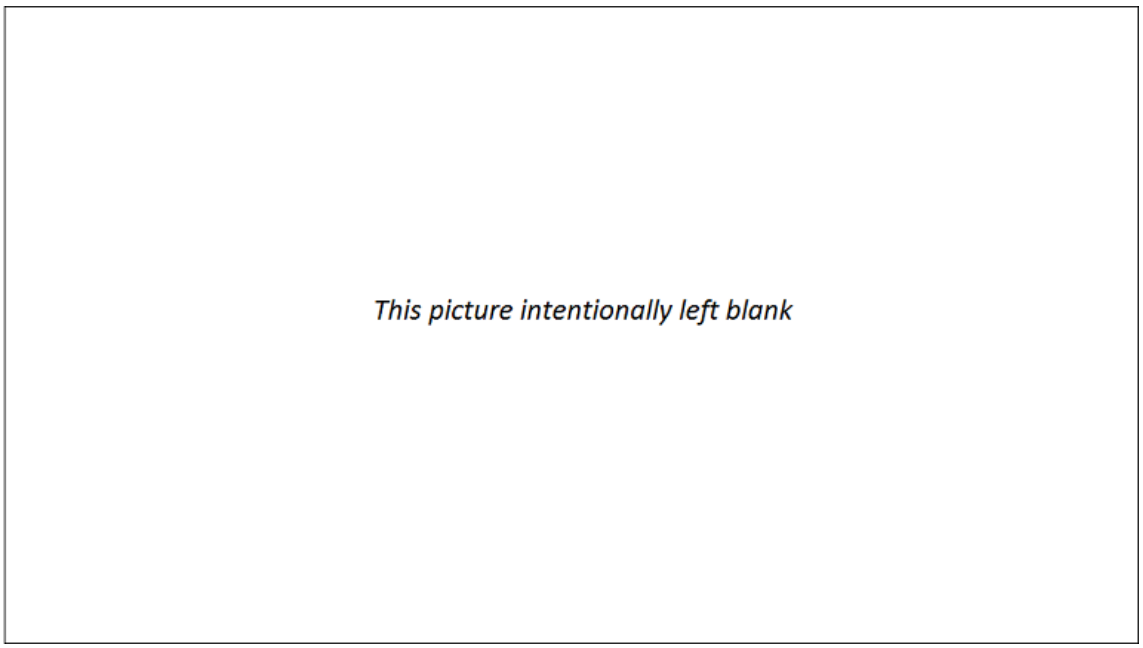
Zeroization of the symmetric key derived from the Master Key

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



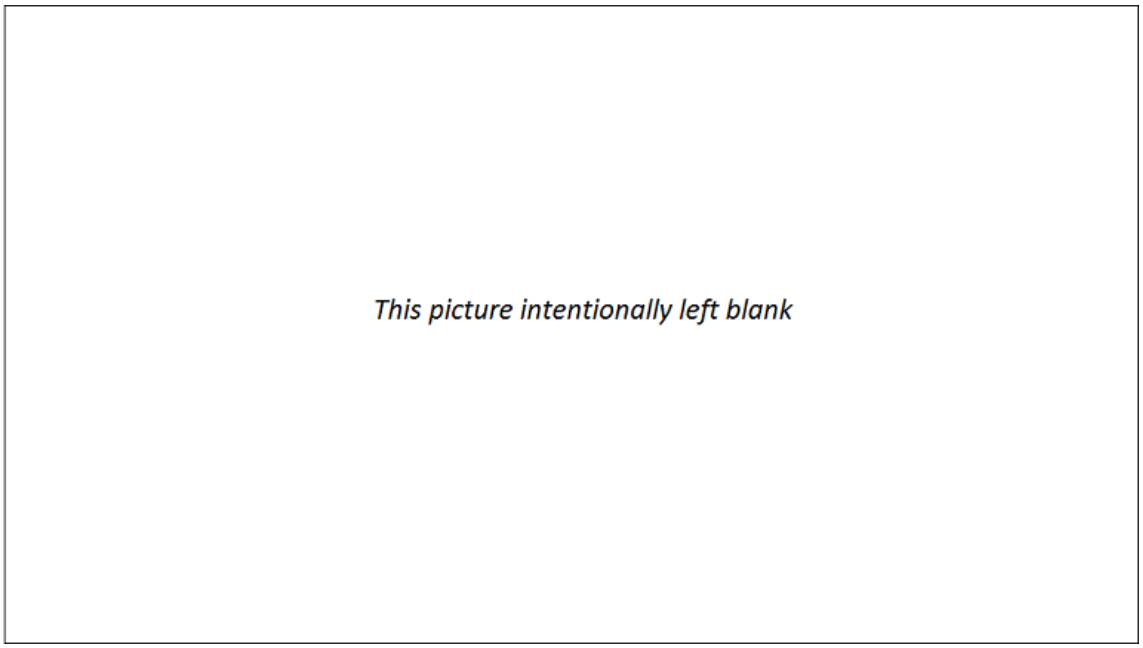
Zeroization of the Master Key

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



Explanation of BCryptDestroyKey and RtlSecureZeroMemory relationship

When BCryptDestroyKey is called the function determines the type of key that is to be destroyed and then makes a call to the provider for that particular key type. For example for an AES key the BCryptDestroyKey call will look at the key structure, determine that the key is an AES key and then a call to the MSCryptDestroyKey function is called. The MSCryptDestroyKey function is the function used for the destruction of symmetric keys such as AES. That function then makes a call to the inline function RtlSecureZeroMemory which performs the actual memory zeroization. Below is a call stack taken from the debugger of the MSCryptDestroyKey function (which holds the RtlSecureZeroMemory code) called by BCryptDestroyKey. Note the current function in the call stack is the MSCryptDestroyKey function and the next call in the stack is the BCryptDestroyKey function (which is calling MSCryptDestroyKey).



For volatile memory keys (networking keys)

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

Testing platforms:

- Surface 3 Pro Windows 10 x64 Enterprise Edition (Web Server)
- Surface 3 Pro Windows 10 x64 Enterprise Edition (Web Client)
- Surface 3 Windows 10 x86 Pro Edition (Web Client)

Test case:

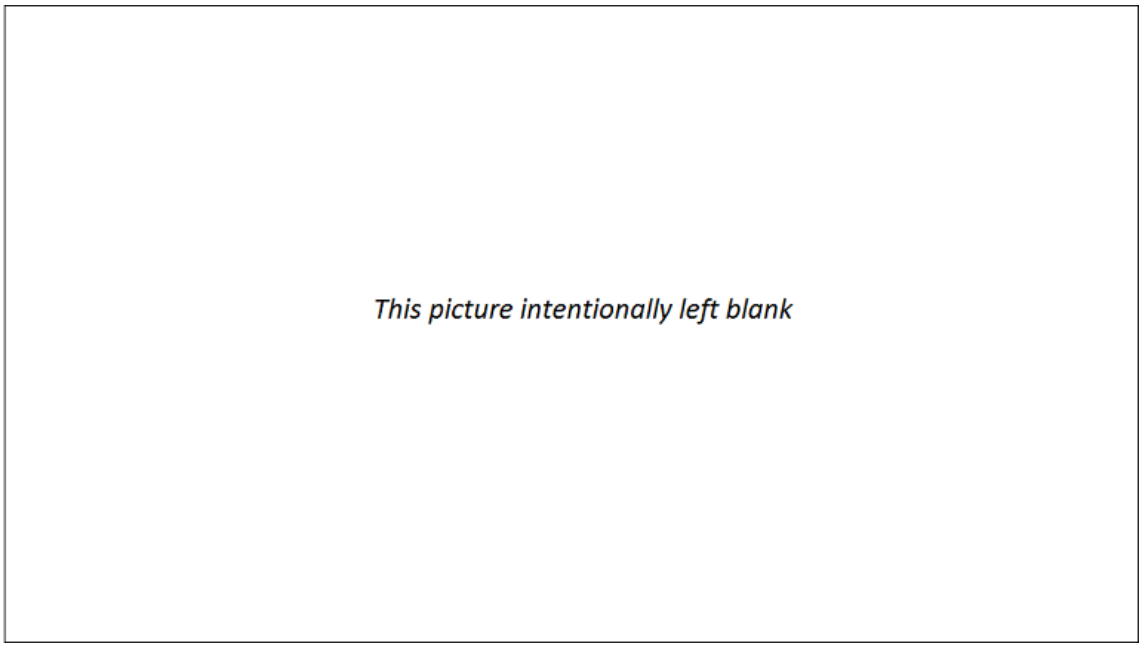
The server has configured a web page only available for TLS connections.

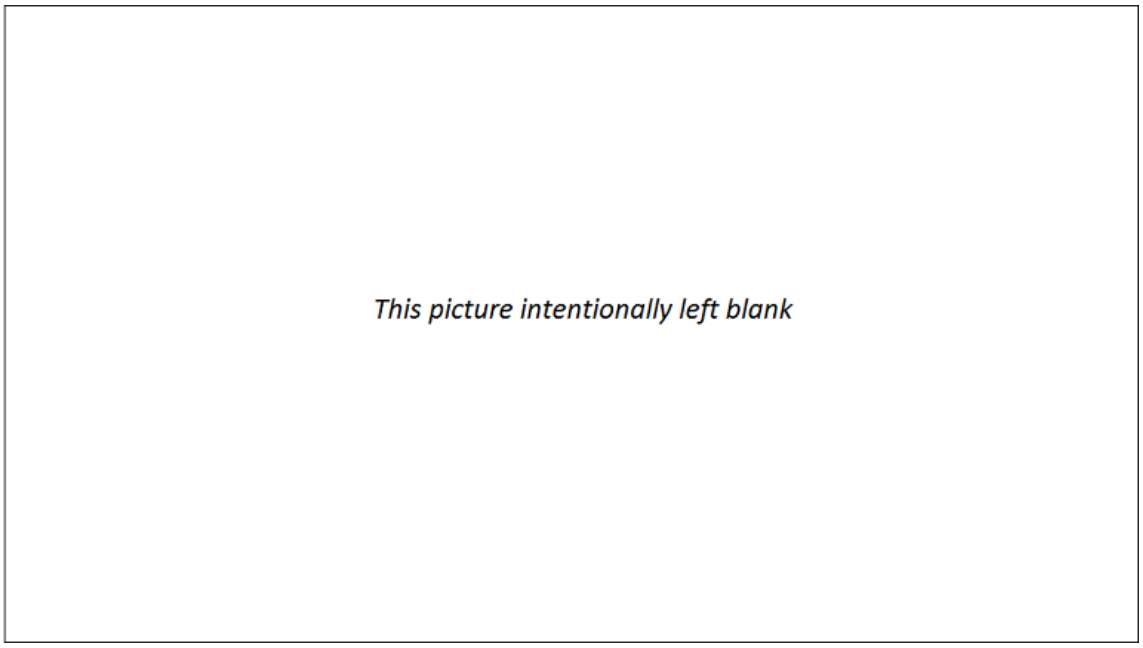
The client has accessed to the web page of the server, forcing a TLS connection using a symmetric cipher suite (AES).

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):



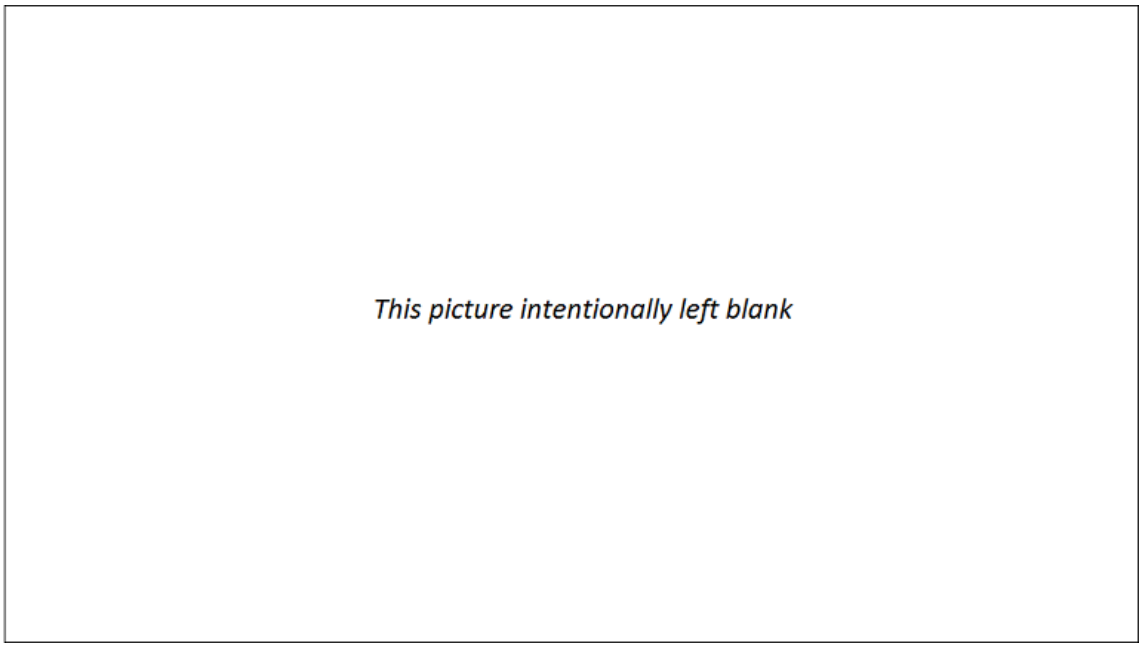


Once the keys are obtained, the client then closes the connection, forcing the zeroization of the keys (given that the keys are only available in memory while the client session is alive).

The next images show how when the session is closed, the "BCryptDestroyKey" function overwrites with zeros the structure which contains the keys:

```
bcrypt!BCryptDestroyKey+0x88:
00007ffd`435e65e8 8be8            mov     ebp,eax
0:002> db 000000c023c83350
000000c0`23c83350  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c83360  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c83370  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c83380  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c83390  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c833a0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c833b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000c0`23c833c0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................

0:012> db 000000b822dc2710
000000b8`22dc2710  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2720  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2730  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2740  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2750  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2760  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2770  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000b8`22dc2780  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
```

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



Finally the memory is dumped:

```
0:012> .dump c:\stuff\symmetrickeydump.dmp
Creating c:\stuff\symmetrickeydump.dmp - mini user dump
Dump successfully written
```

The generated file is opened with a Hex editor, the keys shown above have been searched without success.

Scenario 1-2: Windows 10 – HTTPS (TLS 1.2) communication with Asymmetric cipher suite (ECDSA)

Testing platforms:

- Surface 3 Pro Windows 10 x64 Enterprise Edition (Web Server)
- Surface 3 Pro Windows 10 x64 Enterprise Edition (Web Client)
- Surface 3 Windows 10 x86 Pro Edition (Web Client)

Test case:

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

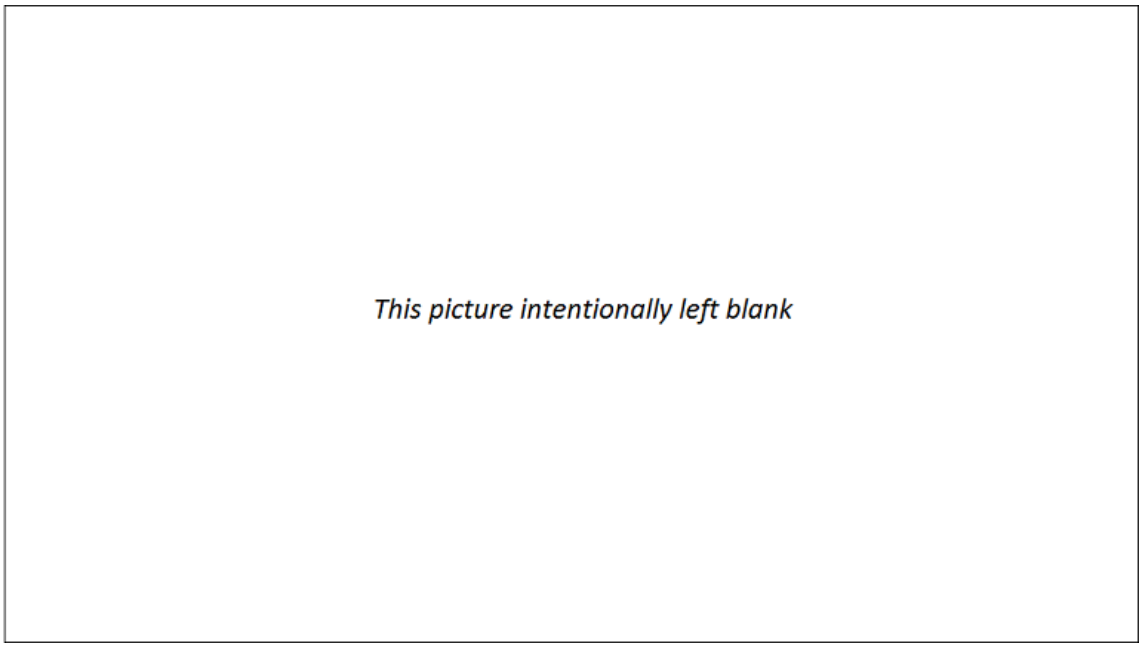
The client has accessed to the web page of the server, forcing a TLS connection with an asymmetric cipher suite (ECDSA).

The following server keys are used:

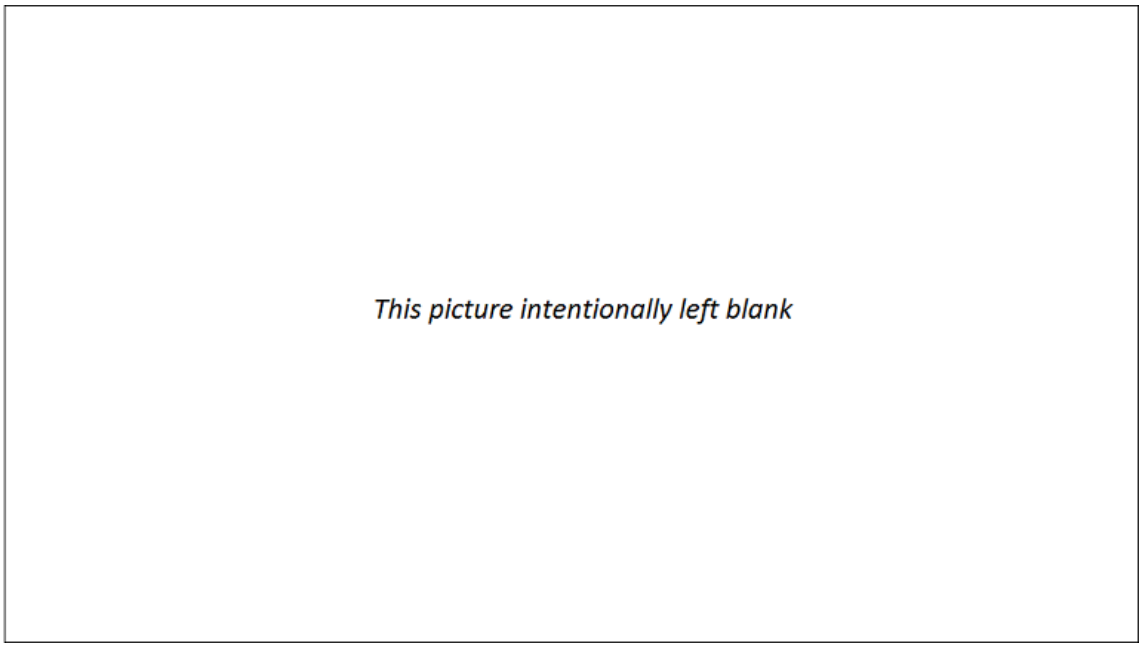
- Private Key

- Public Key

The next image shows the values of the keys structure:

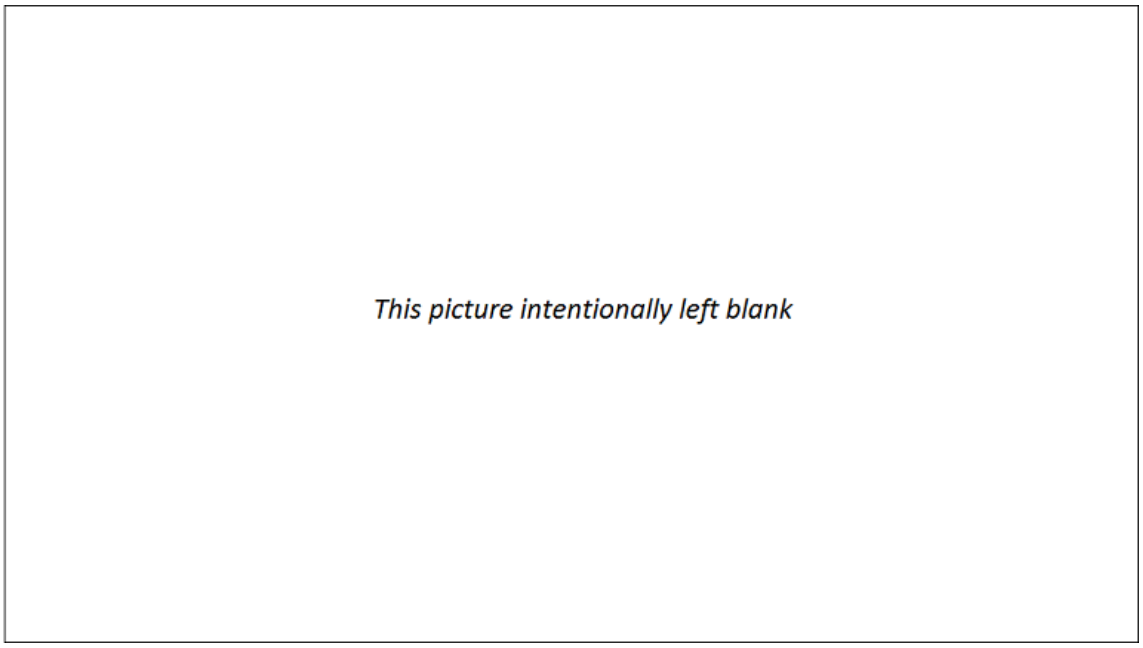


The below is a dump of the private key:

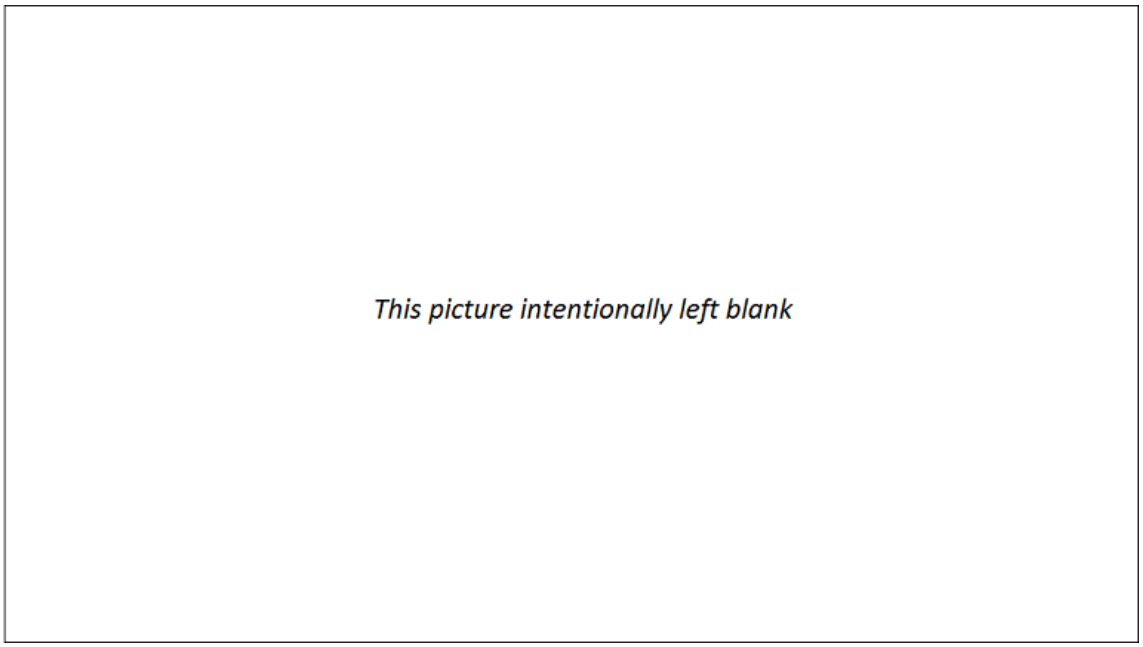


Once the keys are obtained, the client then closes the connection, forcing the zeroization of the keys. Given that the keys are only available in memory while the client session is alive.

The next images show how when the session is closed, the "BCryptDestroyKey" function overwrites with zeros the structure which contains the keys:



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



Finally the memory is dumped. The generated file is opened with a Hex editor, the keys shown above have been searched without success.

Scenario 2-1: Windows Server 2012 R2 – HTTPS (TLS 1.2) communication with Symmetric cipher suite (AES)

Testing platforms:

- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition Hyper-V
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition Hyper-V

Test case:

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

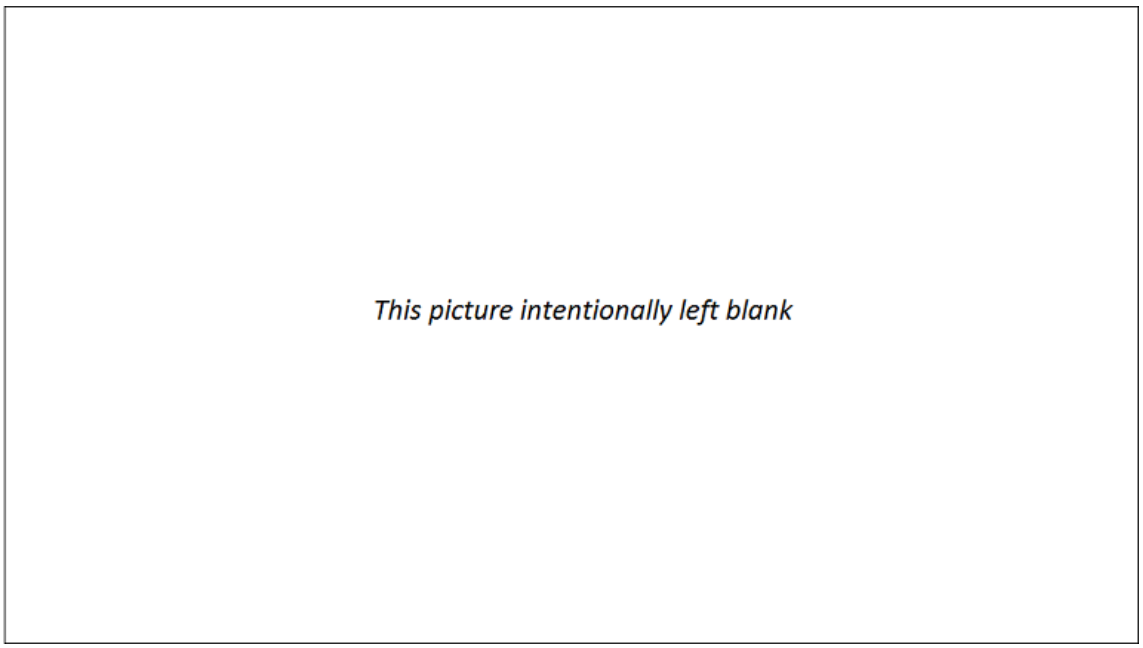
The client has accessed to the web page of the server, forcing a TLS connection using a symmetric cipher suite (AES).

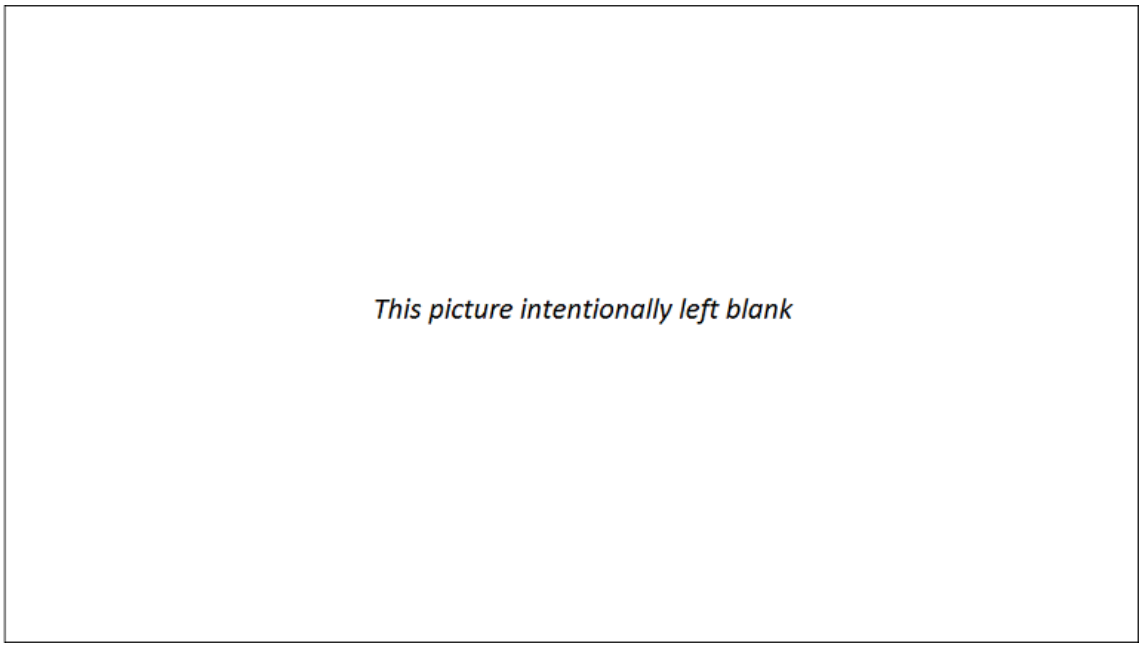
As part of the TLS protocols two symmetric keys are generated:

- inbound key
- outbound key

The 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):

inbound key value:





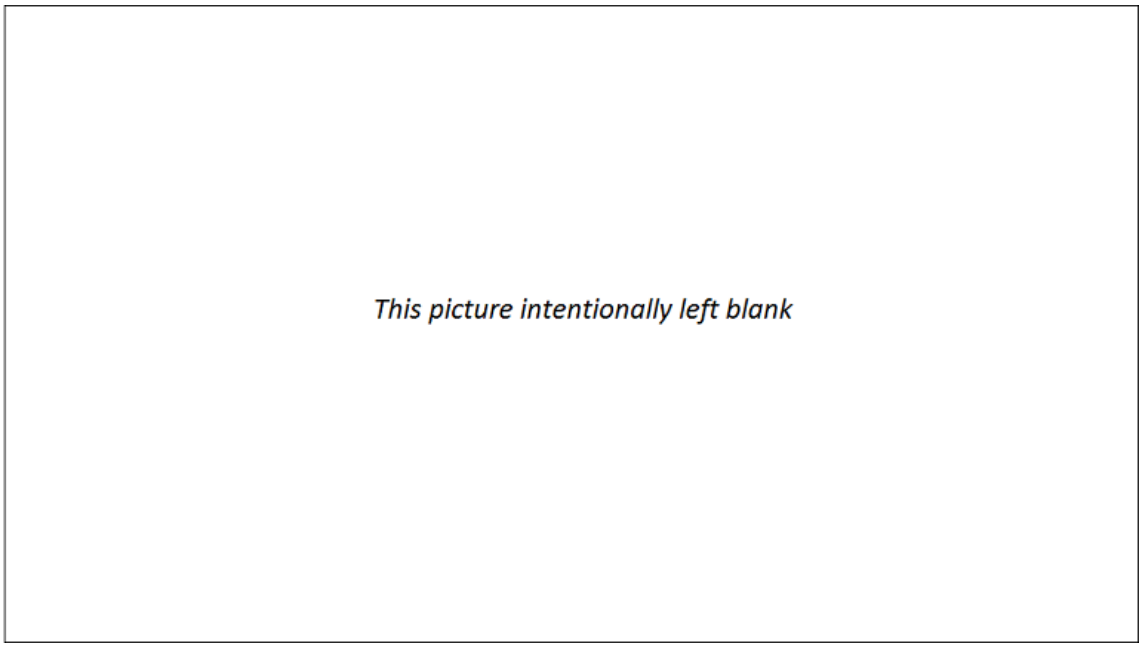
Once the keys are obtained, the client then closes the connection, forcing the zeroization of the keys (given that the keys are only available in memory while the client session is alive).

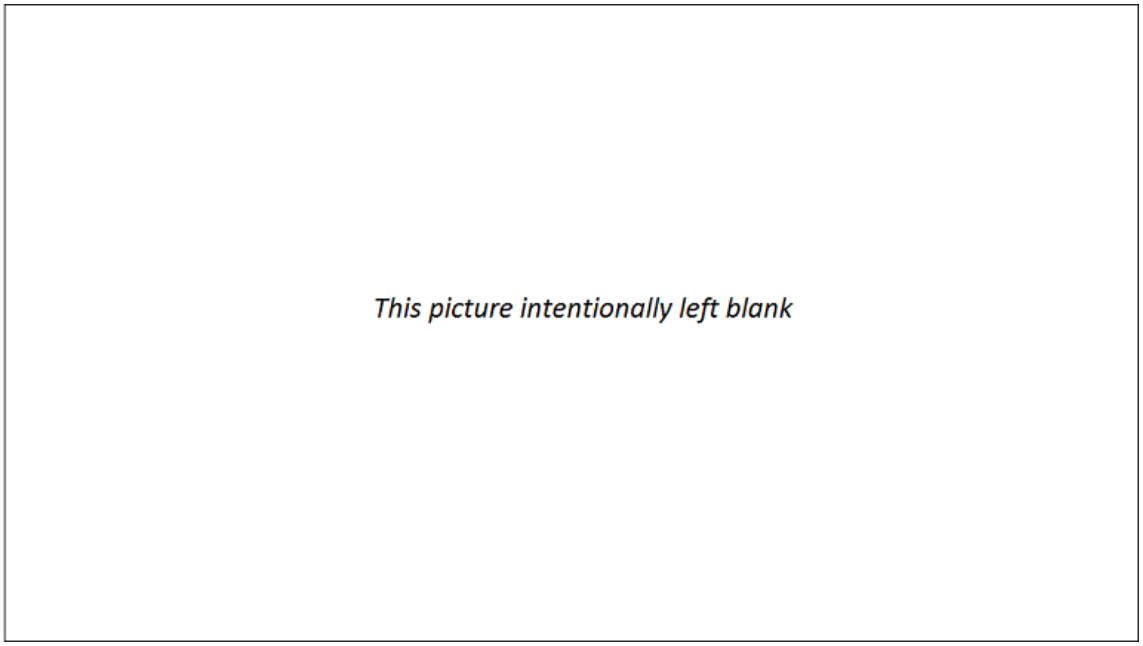
The next images show how when the session is closed, the inbound key is overwritten with zeros:

Dump the key memory after zeroization

```
kd> db 000000acbe7ab750 l 0x26e
000000ac`be7ab750  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab760  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab770  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab780  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab790  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7a0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7c0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7d0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7e0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab7f0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab800  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab810  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab820  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab830  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab840  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab850  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab860  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab870  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab880  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab890  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8a0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8c0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8d0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8e0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab8f0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab900  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab910  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab920  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab930  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab940  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab950  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab960  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab970  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab980  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab990  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab9a0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7ab9b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00        ..............
```

outbound key value:





The next images show how when the session is closed, the outbound key is overwritten with zeros:

Dump the key memory after zeroization

```
kd> db 000000acbe7badc0 l 0x26e
000000ac`be7badc0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7badd0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bade0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7badf0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae00  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae10  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae20  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae30  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae40  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae50  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae60  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae70  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae80  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bae90  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baea0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baeb0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baec0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baed0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baee0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baef0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf00  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf10  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf20  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf30  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf40  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf50  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf60  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf70  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf80  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baf90  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bafa0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bafb0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bafc0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bafd0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bafe0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7baff0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bb000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bb010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
000000ac`be7bb020  00 00 00 00 00 00 00 00-00 00 00 00 00 00
```

Finally the memory is dumped.

The generated file is opened with a Hex editor, the keys shown above have been searched without success.

Scenario 2-2: Windows Server 2012 R2 – HTTPS (TLS 1.2) communication with Asymmetric cipher suite (ECDSA)

Testing platforms:

- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition Hyper-V
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition Hyper-V

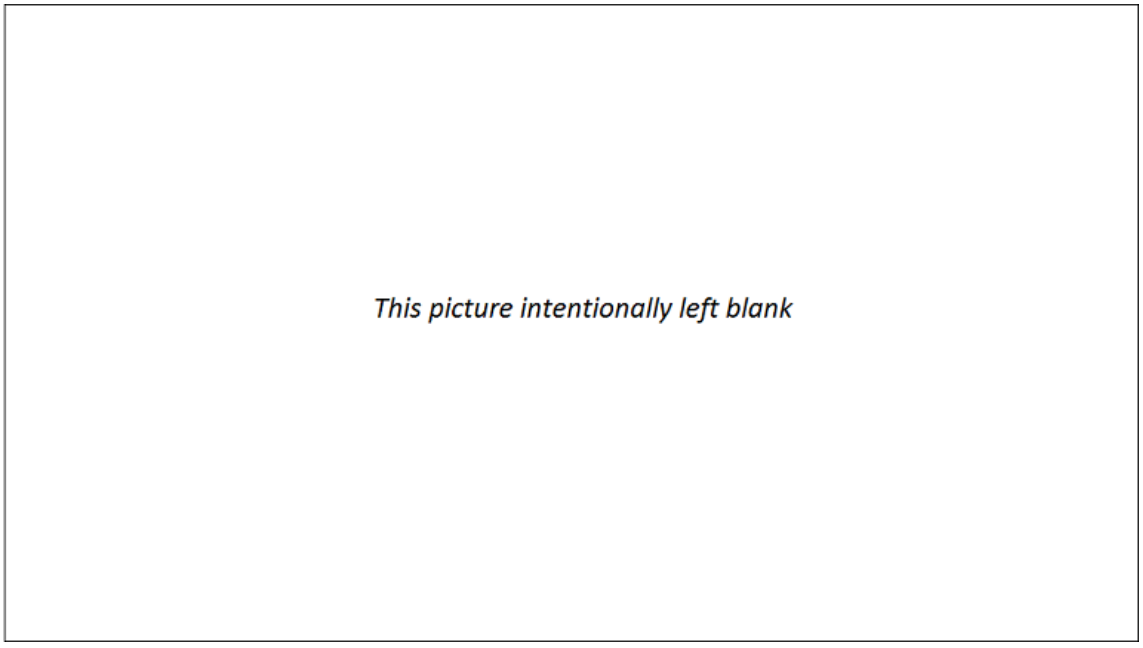
Test case:

The client has accessed to the web page of the server, forcing a TLS connection an asymmetric cipher suite (ECDSA).

The following server keys are used:

- Private Key
- Public Key

The next image shows the values of the keys structure:

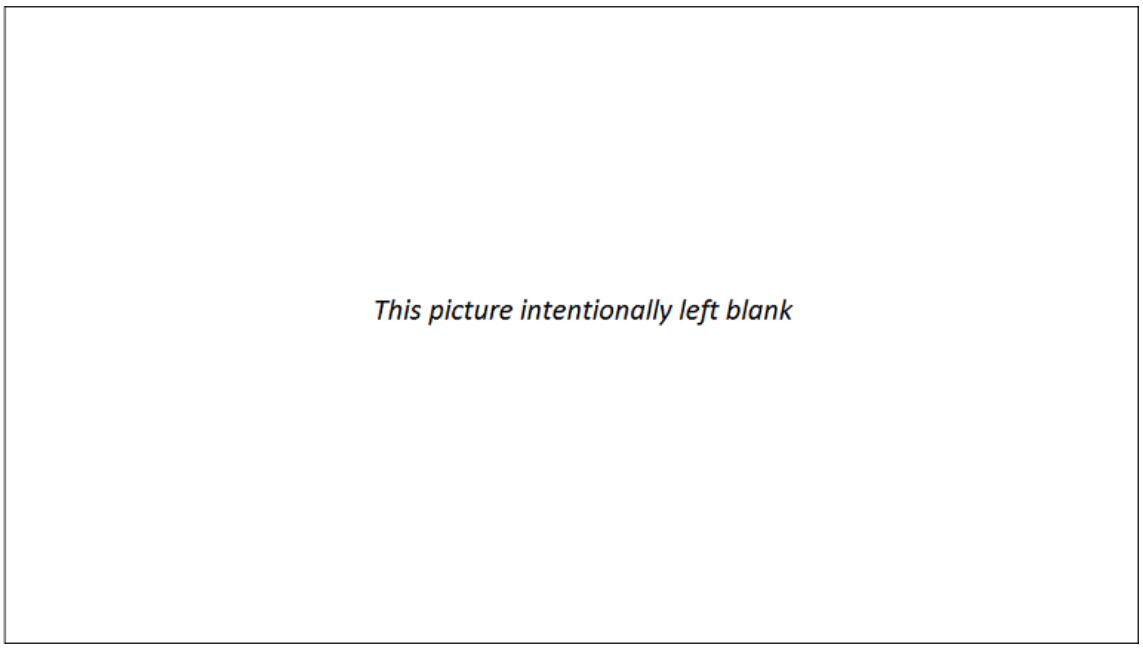


The below is a dump of the private key:

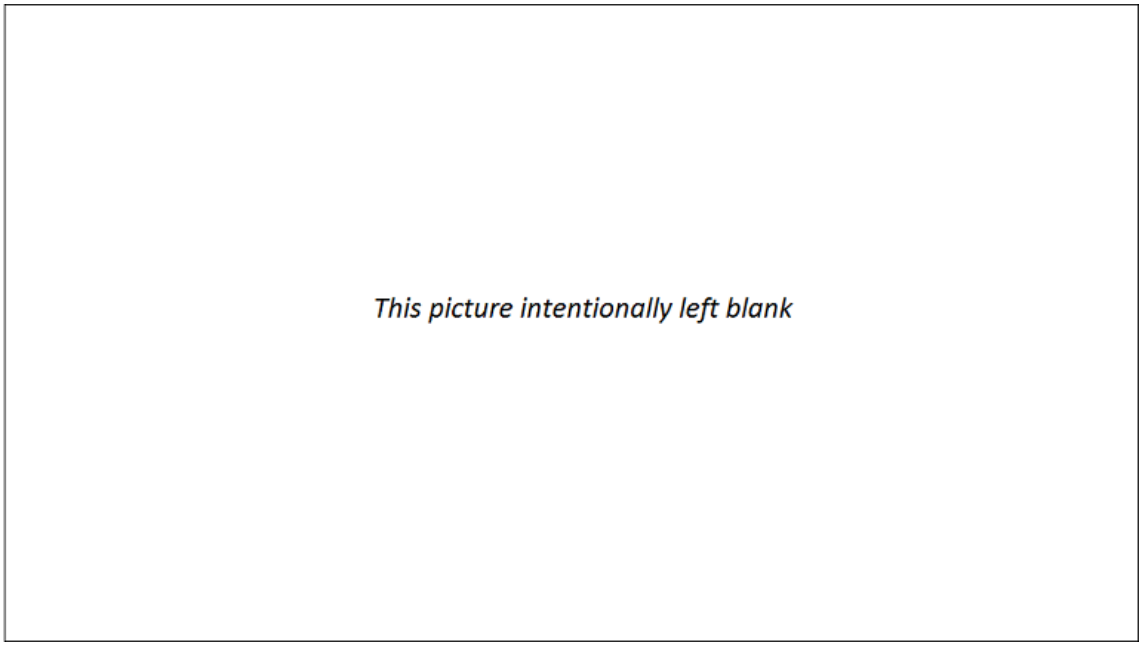
```
Dump the exponent (which is the private portion of the key)
notice that it is 256 bits (0x20 bytes)

kd> db 0x000000ac`be7fa310 l 0x20
000000ac`be7fa310  ac ad 24 c7 28 a7 22 b7-97 be 6e d5 c2 48 8d fb  ..$.(."...n..H..
000000ac`be7fa320  39 48 58 80 8d 53 31 b8-ff f7 52 f3 55 5c c3 c6  9HX..S1...R.U\..
```

Once the keys are obtained, the client then 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 images show how when the session is closed, the "BCryptDestroyKey" function overwrites with zeros the structure which contains the keys:



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



Finally the memory is dumped.

The generated file is opened with a Hex editor, the keys shown above have been searched without success.

For volatile memory keys ("secure storage" DPAPI keys)

Scenario 3-1: Encrypt and decrypt sensitive data using the CryptProtectData and CryptUnprotectData interfaces showing how the volatile keys are zeroized after using them.

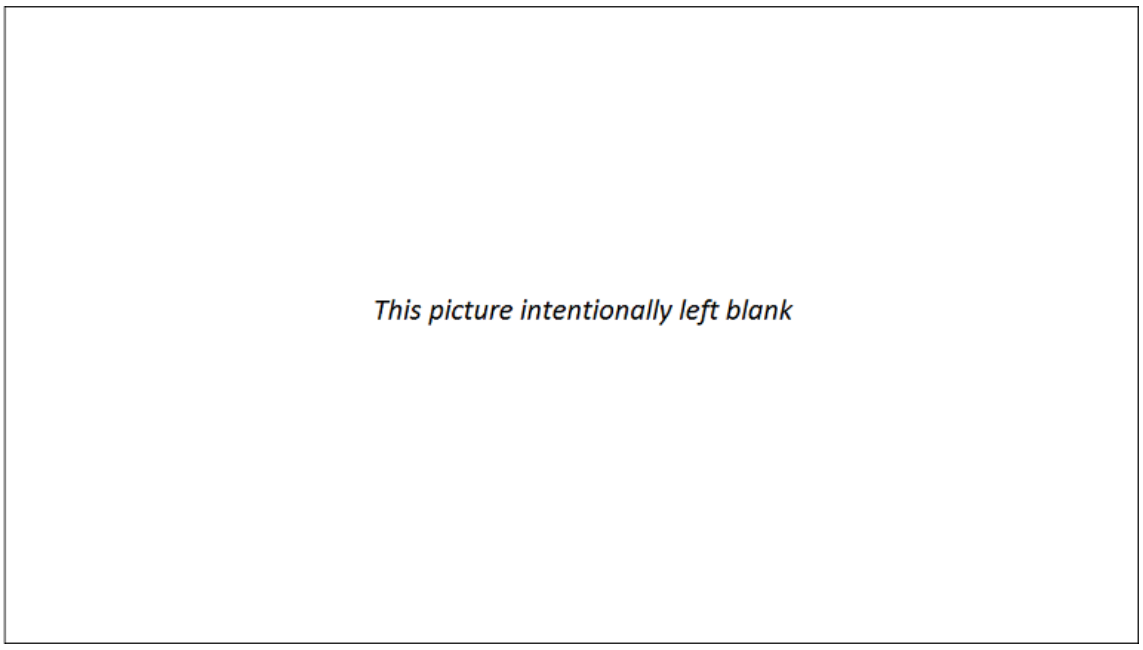
Testing platform:

- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition Hyper-V
- Surface 3 Pro Windows 10 x64 Enterprise Edition

Test case:

Debugging screenshot of rgbMKEncryptionKey zeroized

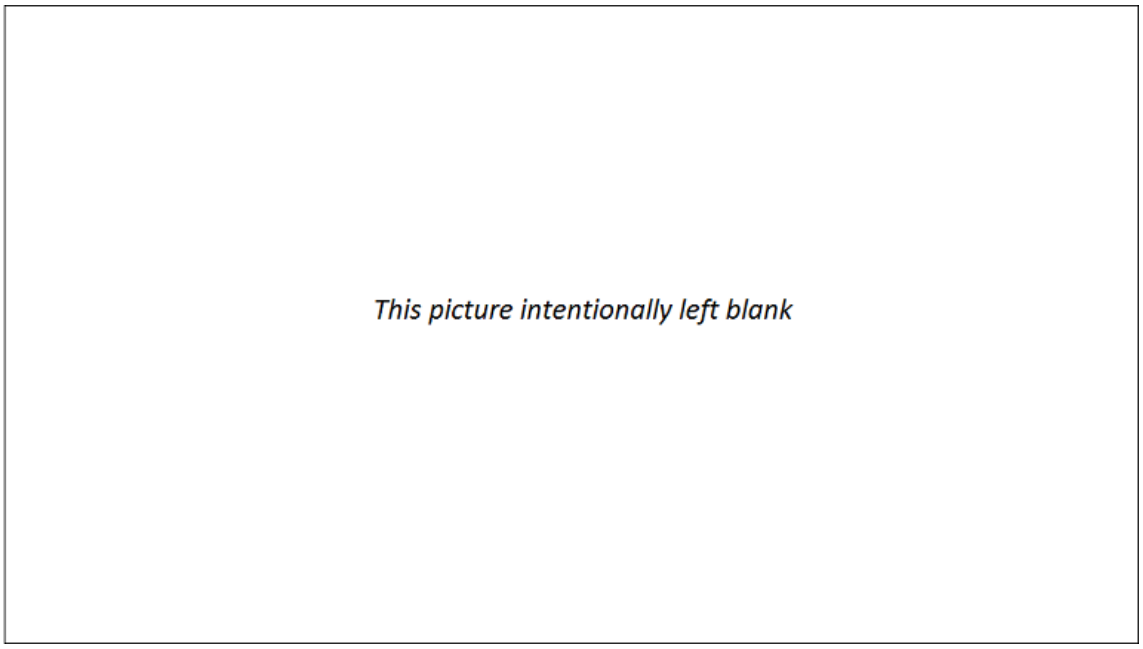
The screenshot of the debugging session below is for the zeroization of the rgbMKEncryptionKey and relates to the code screenshot in the "PBKDF2 HMAC key zeroization" section of this document.

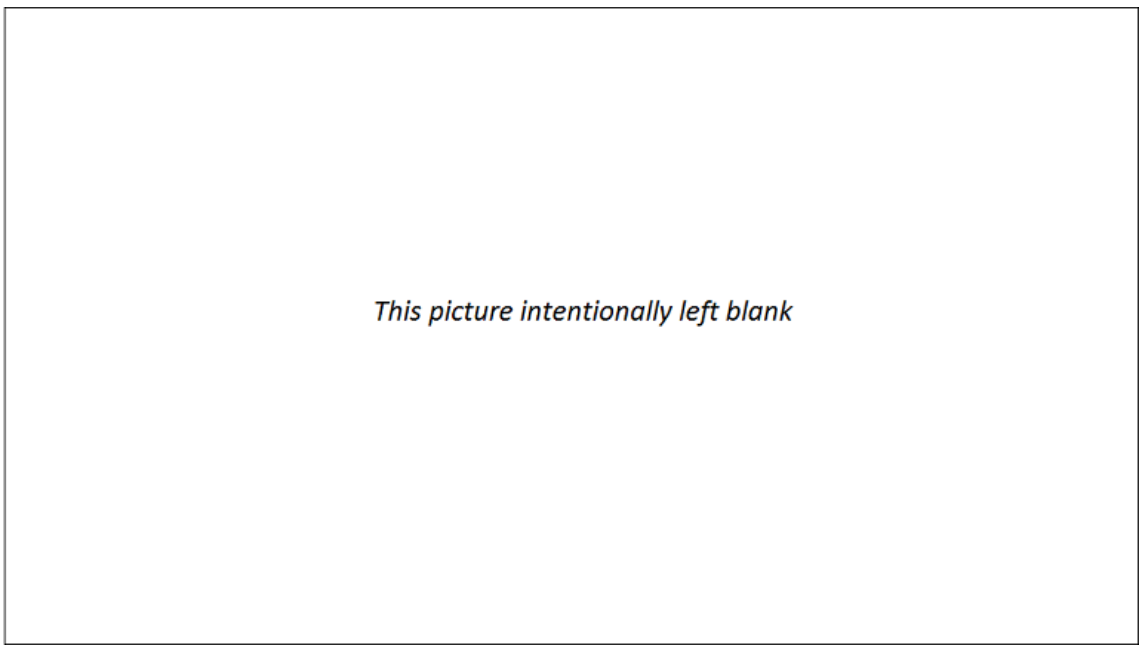


Debugging screenshot of AES key zeroized when BCryptDestroyKey is called

The screenshots of the debugging session below is for the zeroization of an AES key when BCryptDestroyKey is called. Note that the zeroization of AES keys actually occurs in the MSCryptDestroyKey function and that is the function that was being debugged in the screenshots below. These screenshots relate 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.

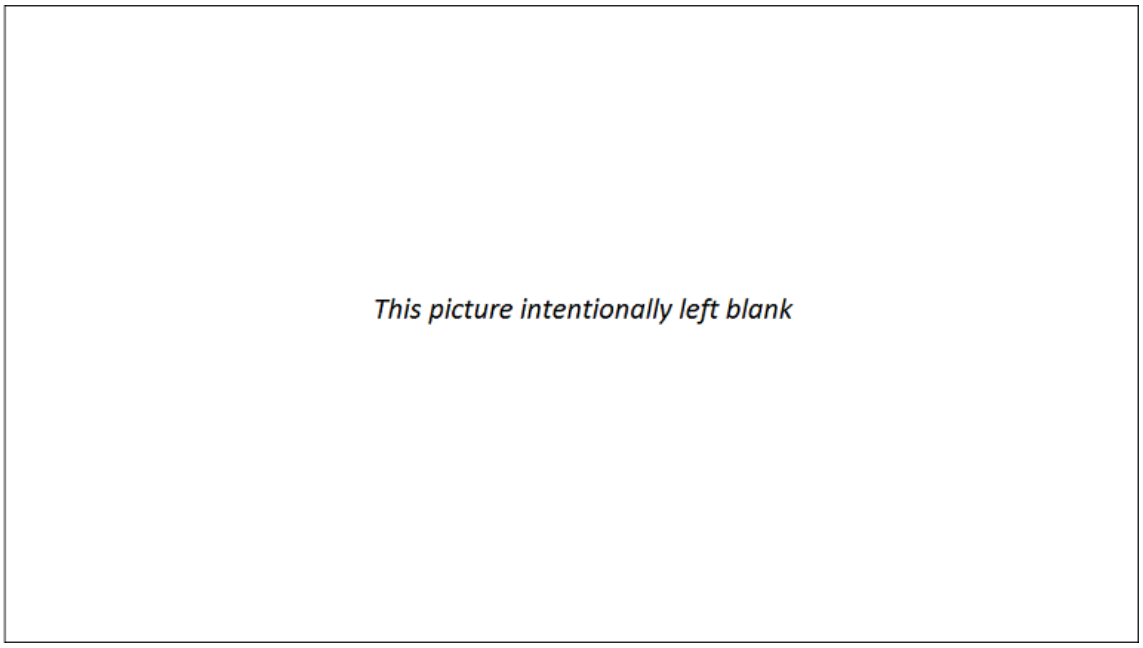
Note that for this screenshot the AES key buffer that is zeroized is larger than just the 256 bit key, the buffer zeroized is 0x26e bytes. This is because the AES key is expanded to make encryption/decryption faster and the entire expanded key must be destroyed. In addition there are header bytes that are zeroized. Because this screenshot is larger we have broken it into two screenshots.





Debugging screenshot of Master Key zeroized

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.



6.3.1.4. Verdict

The evaluator has performed several tests considering different scenarios. In these tests the evaluator has reviewed the source code parts where the networking and DPAPI volatile keys are zeroized and performed debug testing verifying the fulfillment of all scenarios. Since in all the scenarios the result obteaines has been the same as expected the PASS verdict is assigned to this testing activity.

6.4. Final Verdict

Due to this, all test activities 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 this FCS_CKM_EXT.3 requirement.

7. FCS_COP.1.1 (SYM)

7.1. Assurance activity

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 128bit 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.

- *KAT1. 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.*

- *KAT2. 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 128bit keys, and the other five shall be 256bit 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.*

- *KAT3. 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 128bit keys, and the second set shall have 256 256bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost Ni 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 128bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost Ni bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.*

- *KAT4. 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 128bit key value of all zeros with an IV of all zeros and using a 256bit 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 128i 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 MultiBlock Message Test

The evaluator will test the encrypt functionality by encrypting an iblock message where 1 < i ≤ 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 iblock message where 1 < i ≤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 3tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128bit blocks. For each 3tuple, 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

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 nonzero 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 nonzero 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 5tuples 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.*
- *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 3tuples 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 802.1102/362r6 document "Proposed Test vectors for IEEE 802.11 TGi", dated September 10, 2002, Section 2.1
AES-CCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.112007 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 nonzero 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 nonzero 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 5tuples 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 AES128) and 512 bit (for AES256) 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 128bit random tweak value) 3tuples 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 base10 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 XTS-AES decrypt.

AES Key Wrap (AESKW) and Key Wrap with Padding (AESKWP) 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 unit length shall be the longest supported plaintext length less than or equal to 64 semiblocks (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 AESKW 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.

7.2. Documentation review activity

7.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

The vendor has specified the NIST CAVP certificate number where it is defined the AES algorithms supported by each OS.

On the other hand, the AGD guidance (Windows 10 and Server 2012 R2 GP OS Operational Guidance.docx) states that for the evaluated version the next security policy needs to be applied (section 1.2.1):

- 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 algorithms can be used, including the AES algorithms defined in the tables above.

7.2.2. Verdict

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

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

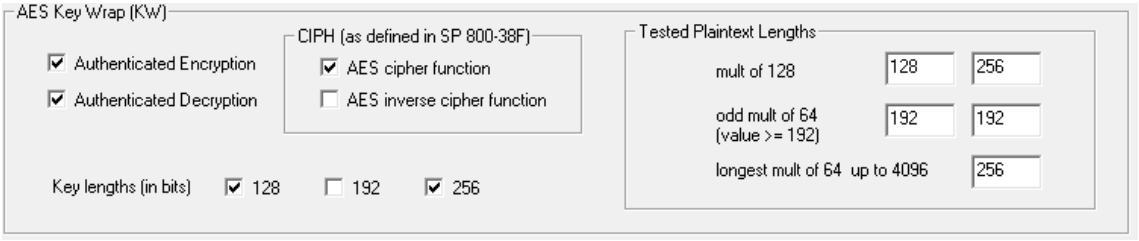
7.3. Test Activity

The evaluator has reviewed the NIST certificates and considers that all AES algorithms are covered by FIPS certifications, except the AES-KW algorithm for Windows Server 2012 R2. Based on that, the Lab has carried out the testing required for meeting the Protection Profile and getting the certificate for this algorithm, as follows:

7.3.1. Test 1

7.3.1.1. Setup

The NIST CAVS tool (19.2) is configured for the input vectors generation as follows:



The Lab has developed two different SWs (AES_KW_DEC.cpp for unwrapping keys and AES_KW_ENC.cpp for wrapping keys) for the AES-KW testing, based on the public APIs for WS2012R2. The following screen shots show the AES_KW_DEC.cpp source code, where the main function is *BCryptImportKey*:

```
    BCRYPT_ALG_HANDLE       hAesAlg = NULL;
    BCRYPT_KEY_HANDLE       hK = NULL;
    BCRYPT_KEY_HANDLE       hP = NULL;
    NTSTATUS                status = STATUS_UNSUCCESSFUL;
    DWORD
        p_length = 0,
        cbData = 0;
    PBYTE
        pK = NULL,
        pC = NULL,
        pP = NULL;

    struct {
        BCRYPT_KEY_DATA_BLOB_HEADER Header;
        UCHAR Key[32];
    } pKBlob;

    pKBlob.Header.dwMagic = BCRYPT_KEY_DATA_BLOB_MAGIC;
    pKBlob.Header.dwVersion = BCRYPT_KEY_DATA_BLOB_VERSION1;
    pKBlob.Header.cbKeyData = K_LENGTH;
    CopyMemory(pKBlob.Key, K, sizeof(pKBlob.Key));

    UNREFERENCED_PARAMETER(argc);
    UNREFERENCED_PARAMETER(wargv);

    pC = (PBYTE)HeapAlloc(GetProcessHeap(), 0, C_LENGTH);
    if (NULL == pC)
    {
        wprintf(L"**** memory allocation failed\n");
        goto Cleanup;
    }
    memcpy(pC, C, C_LENGTH);
```

```
    // Open an algorithm handle.
    if (!NT_SUCCESS(status = BCryptOpenAlgorithmProvider(
        &hAesAlg,
        BCRYPT_AES_ALGORITHM,
        MS_PRIMITIVE_PROVIDER,
        0)))
    {
        wprintf(L"**** Error 0x%x returned by BCryptOpenAlgorithmProvider\n", status);
        goto Cleanup;
    }

    // Calculate the size of the buffer to hold the KeyObject.
    if (!NT_SUCCESS(status = BCryptGetProperty(
        hAesAlg,
        BCRYPT_OBJECT_LENGTH,
        (PBYTE)&p_length,
        sizeof(DWORD),
        &cbData,
        0)))
    {
        wprintf(L"**** Error 0x%x returned by BCryptGetProperty\n", status);
        goto Cleanup;
    }


    pK = new (std::nothrow) BYTE[p_length];
    if (NULL == pK)
    {
        status = E_OUTOFMEMORY;
    }


    if (!NT_SUCCESS(status = BCryptSetProperty(
        hAesAlg,
        BCRYPT_CHAINING_MODE,
        (PBYTE)BCRYPT_CHAIN_MODE_CBC,
        sizeof(BCRYPT_CHAIN_MODE_CBC),
        0)))
    {
        wprintf(L"**** Error 0x%x returned by BCryptSetProperty\n", status);
        goto Cleanup;
    }
```

```
int aux = sizeof(pKBlob);

if (!NT_SUCCESS(status = BCryptImportKey(
    hAesAlg,
    NULL,
    BCRYPT_KEY_DATA_BLOB,
    &hK,
    pK,
    p_length,
    (PUCHAR)&pKBlob,
    sizeof(pKBlob),
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptImportKey\n", status);
    goto Cleanup;
}

pP = (PBYTE)HeapAlloc(GetProcessHeap(), 0, p_length);
if (NULL == pP)
{
    wprintf(L"**** memory allocation failed\n");
    goto Cleanup;
}

if (!NT_SUCCESS(status = BCryptImportKey(
    hAesAlg,
    hK,
    BCRYPT_AES_WRAP_KEY_BLOB,
    &hP,
    pP,
    p_length,
    pC,
    C_LENGTH,
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptImportKey\n", status);
    goto Cleanup;
}

cout << "Unwrapped Key:" << endl;
PrintBytes((BYTE *)pP, p_length);
```

On the other hand, the following screen shot shows the main function in AES_KW_ENC.cpp source code, which it is called *BCryptExportKey*:

```
//Get size of wrapped key
pC = (PBYTE)HeapAlloc(GetProcessHeap(), 0, p_length);
if (NULL == pC)
{
    wprintf(L"**** memory allocation failed\n");
    goto Cleanup;
}

if (!NT_SUCCESS(status = BCryptExportKey(
    hP,
    hK,
    BCRYPT_AES_WRAP_KEY_BLOB,
    pC,
    p_length,
    &p_length,
    0)))
{
    wprintf(L"**** Error 0x%x returned by BCryptExportKey\n", status);
    goto Cleanup;
}

cout << "Wrapped Key:" << endl;
PrintBytes_CAVP((BYTE *)pC, p_length);
```

Additionally, AES_KW_ENC.cpp and AES_KW_DEC.cpp files parser the input vectors generated by the CAVS tool and generate the output vectors according to the expected format for the CAVS tool.

7.3.1.2. Procedure

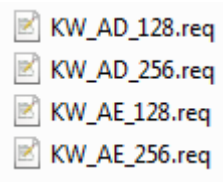
1. Generate the input vectors.
2. Check that the input vectors are correctly generated.
3. Put the input vectors in the same path of the executable file.
4. If a wrapping operation is performed, execute AES_KW_ENC.exe. Otherwise, if an unwrapping operation is performed execute AES_KW_DEC.exe.
5. After the execution, a file with the output vectors (.rsp) will be generated.
6. Put the output vectors in the CAVS tool and verify the results.

7.3.1.3. Results

The test has been performed in the following platform:

- Dell Optiplex 755 with Windows Server 2012 R2.

The input vectors have been properly generated:



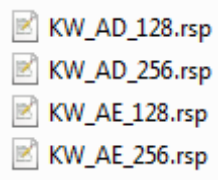
100 vectors have been generated for each KW mode and key size, e.g. KW_AE_128:

```
# CAVS 19.1
# 'NIST SP 800-38F KW-AE with AES-128 cipher function' information for WS2012R2
# Seed = 3aa26f2fdef299403da2447febb92d0760de7f0897c0726f0cf79466cfb84a001329d0e1ae422ce4c2c5501711cca564256f5567bd31b9ffdcfafa3f981d60c4
# Generated on Mon Mar 07 11:11:32 2016


[PLAINTEXT LENGTH = 128]

COUNT = 0
K = 1969d4f5527f32c82c181e9eb37b7c2b
P = bb64007e99bdd8e74bb293d2b322fc50

COUNT = 1
K = 4b9964e1790c63bcf73a3f5edbc640b9
P = 6828fbd628af4bdae2225fbec2f9f690

COUNT = 2
K = 22fdce07dac80acfbed32f748ed51d00
P = 4fadf2d7427bccf7eb08923a42107d7d
```

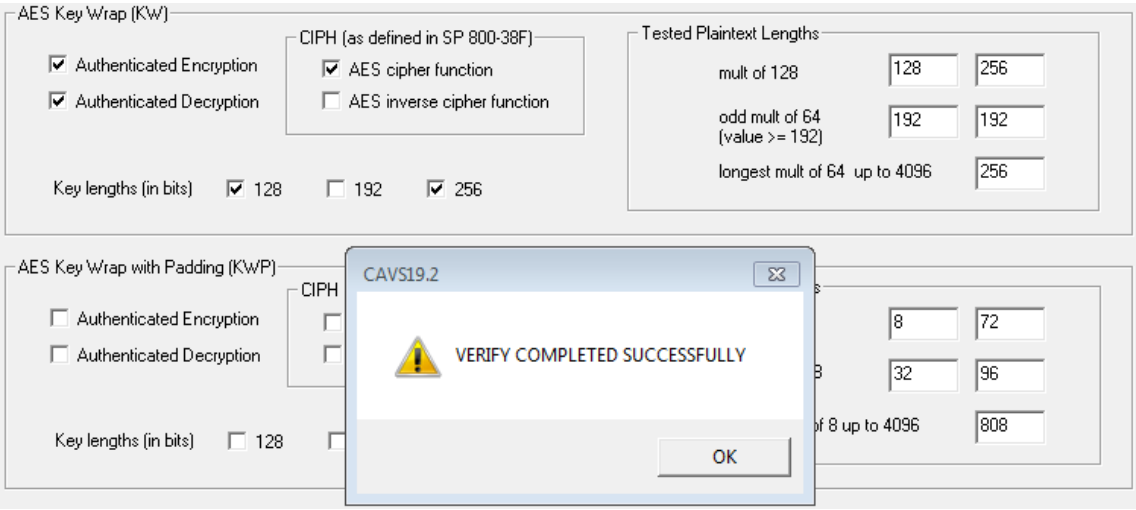
. . .

```
COUNT = 97
K = 4a877fd96151cc8d6ddf34201c23c9a3
P = 9635c9c18d7f9774042617c96991fdbc

COUNT = 98
K = 43fb3ee02c52a864cc7959c2232d2438
P = b7e9ca0112259ea7bd4b8afaa7a47a1b

COUNT = 99
K = ef2ac5549ecb61de366880174aaa7196
P = 733bd0fb12d36c1dab3561d9df753565

[PLAINTEXT LENGTH = 256]
```

After executing the AES_KW_ENC.exe and AES_KW_DEC.exe the following files are generated:



The CAVS tool has verified the files above checking proper implementation of the KW algorithm:



7.3.1.4. Verdict

The evaluator considers that, the tests results obtained during the test activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the PASS verdict is assigned to the test activity.

7.4. Final Verdict

Due to all activities 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 this FCS_COP.1 .1 (SYM) requirement.

8. FCS_COP.1 .1(HASH)

8.1. Assurance activity

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 ≤ i ≤ 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 + 8□99□i, where 1 ≤ i ≤ 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.

- ***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.*

8.2. Documentation review activity

8.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494,

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
		#1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

The vendor has specified the NIST CAVP certificate number where it is defined the hashing algorithms supported by each OS, e.g.:

2396	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 8.1, Microsoft Windows Server 2012 R2, Microsoft Windows Storage Server 2012 R2, Microsoft Windows RT 8.1, Microsoft Surface with Windows RT 8.1, Microsoft Surface Pro with Windows 8.1, Microsoft Surface 2, Microsoft Surface Pro 2, Microsoft Surface Pro 3, Microsoft Windows Phone 8.1, Microsoft Windows Embedded 8.1 Industry RSA32 Algorithm Implementations** Version 6.3.9600	NVIDIA Tegra 4 Quad-Core w/ Microsoft Surface 2 w/ Windows RT 8.1 (ARMv7 Thumb-2) ; NVIDIA Tegra 3 Quad-Core w/ Windows RT 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon S4 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 400 w/ Windows Phone 8.1 (ARMv7 Thumb-2); Qualcomm Snapdragon 800 w/ Windows Phone 8.1 (ARMv7 Thumb-2);	6/6/2014	**SHA-1** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only) "The Microsoft Windows RSA32 Library algorithm implementation is used by various Microsoft libraries to provide AES, RSA, and SHS (SHA) support." *07/10/14: Added new tested information;* *12/16/14: Added new tested information;* *03/13/15: Added new tested information;*
2886	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399 USA -Tim Myers TEL: 800-Microsoft	**Microsoft Windows 10, Microsoft Surface Pro 3 with Windows 10, Microsoft Surface 3 with Windows 10, Microsoft Surface Pro 2 with Windows 10, Microsoft Surface Pro with Windows 10 SymCrypt Cryptographic Implementations** Version 10.0.10240	Intel Core i7 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64); AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 (x64); Intel Core i3 without AES-NI or PCLMULQDQ or SSSE 3 w/ Windows 10 (x86); AMD A4 with AES-NI and PCLMULQDQ and SSSE 3 w/ Windows 10 Enterprise (x64); Intel x64 Processor with AES-NI w/ Microsoft Surface Pro w/ Windows 10 Enterprise (x64); Intel Core i5 with AES-NI w/ Microsoft Surface Pro 2 w/	8/15/2015	**SHA-1** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only) "The Microsoft Windows Next Generation Cryptographic algorithm implementation provides enhanced support for AES, AES DRBG, HMAC, SHS (SHA), and Triple-DES. All implementations are packaged into a library used by Microsoft and other third-party applications." *09/17/15: Updated implementation information;* *10/09/15: Added new tested information;*

On the other hand, 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)1, 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).”

8.2.2. Verdict

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

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

8.3. 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.

8.4. Final Verdict

Due to all activities 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 this FCS_COP.1 .1(HASH) requirements.

9. FCS_COP.1 .1(SIGN)

9.1. Assurance activity

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 1864 Signature Generation Test. For each supported NIST curve (i.e., P256, P384 and P521) and SHA function pair, the evaluator will generate 10 1024bit 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 1864 Signature Verification Test. For each supported NIST curve (i.e., P256, P384 and P521) and SHA function pair, the evaluator will generate a set of 10 1024bit 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 reponses indicate success and 5 reponses 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*

9.2. Documentation review activity

9.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method

Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

The vendor has specified the NIST CAVP certificate number where it is defined the Signature algorithms supported by each OS.

9.2.2. Verdict

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

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

9.3. 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.

9.4. Final Verdict

Due to all activities 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 this FCS_COP.1 .1(SIGN) requirement.

10. FCS_COP.1.1 (HMAC)

10.1. Assurance activity

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.

10.2. Documentation review activity

10.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

The vendor has specified the NIST CAVP certificate number where it is defined the Keyed-Hashing algorithms (HMAC) supported by each OS.

10.2.2. Verdict

The assurance activity does not require any documentation information. Anyway the evaluator has provided information about the Hashing algorithms (HMAC) supported by each OS.

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

10.3. 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.

10.4. Final Verdict

Due to all activities 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 this FCS_COP.1.1 (HMAC) requirement.

11. FCS_RBG_EXT.1.1

11.1. Assurance activity

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.*

11.2. Documentation review activity

11.2.1. 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 the OS versions:

Cryptographic Operation	Standard	Windows 10 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW, XTS, and GCM modes	NIST CAVP #3497, #3498, #3507, #3476
Digital signature	FIPS 186-4 RSA	NIST CAVP #1802, #1783, #1784, #1798
Digital signature	FIPS 186-4 DSA	NIST CAVP #983
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #706
Hashing	FIPS 180-4 SHA-1 and SHA-256, SHA-384, SHA-512	NIST CAVP #2886, #2871
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	NIST CAVP #2233
Random number generation	NIST SP 800-90 CTR_DRBG	NIST CAVP #868
Key agreement	NIST SP 800-56A ECDH NIST SP 800-56B RSA	NIST CAVP #64 Tested by the CC evaluation lab
Key-based key derivation	SP800-108	NIST CAVP #66
IKEv1	SP800-135	NIST CVL #575
IKEv2	SP800-135	NIST CVL #575
TLS	SP800-135	NIST CVL #575

Cryptographic Operation	Standard	Server 2012 R2 Evaluation Method
Encryption/Decryption	FIPS 197 AES For ECB, CBC, CFB8, CCM, KW and GCM modes	NIST CAVP #2848, #2832, #2853 KW is tested by the CC evaluation lab
Digital signature	FIPS 186-4 DSA	NIST CAVP #855
Digital signature	FIPS 186-4 RSA	NIST CAVP #1487, #1493, #1494, #1519
Digital signature	FIPS 186-4 ECDSA	NIST CAVP #505
Hashing	FIPS 180-4 SHA-1, SHA-256, SHA-384, and SHA-512	NIST CAVP #2373, #2396
Key Agreement	NIST SP 800-56A EC DH NIST SP 800-56B RSA	NIST CAVP #47 Tested by the CC evaluation lab
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #1773
Random number generation	NIST SP 800-90	NIST CAVP #489 for CTR_DRBG
Key-based key derivation	SP800-108	NIST CAVP #30
IKEv1	SP800-135	NIST CVL #323
IKEv2	SP800-135	NIST CVL #323
TLS	SP800-135	NIST CVL #323

The vendor has specified the NIST CAVP certificate number where it is defined the Random number generation algorithm (CTR_DRBG) supported by each OS.

11.2.2. Verdict

The assurance activity does not require any documentation information. Anyway the evaluator has provided information about the Random number generation algorithm (CTR_DRBG) supported by each OS. For both OS versions, W10 and WS12R2 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

11.3. 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.

11.4. Final Verdict

Due to all activities 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 this FCS_RBG_EXT.1.1 requirement.

12. FCS_RBG_EXT.1.2

12.1. Assurance activity

Documentation shall be produced - and the evaluator will perform the activities - in accordance with Appendix E 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.

12.2. Documentation review activity

12.2.1. Findings

The vendor has provided the following documentation regarding the entropy:

- Windows 8 1 and Server 2012 R2 RNG Documentation
- Windows 10 RNG Documentation
- Entropy Validation on Windows 10 for GP OS Evaluation

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 includes more information about how the entropy source is health-tested before using it, as it is 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, 384 bits thereafter on demand based on an OS timer).*
- *Random values gathered periodically by calling the RDRAND CPU instruction, (256 bits of entropy on demand based on an OS timer).*

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. The raw entropy data is separated in blocks of an established size, and each generated block is compared with previous one, if they are equals the entropy data is dropped, if not, it is considered good enough for seeding a DRBG instance."

Moreover, the "entropy validation" 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 table intentionally left blank

Table 2 – Entropy Validation Results

This table intentionally left blank

12.2.2. Verdict

The evaluator has reviewed the security target (ST v0.10), in particular TSS content section 6.2.1 Cryptographic Algorithms and Operations. This section describes how the entropy source is health-tested before using it comparing the generated entropy blocks between them. The vendor has also checked the new statistical tests results provided in the new release of the "entropy validation" document, verifying that all platforms under evaluations have been considered.

Due to this, the evaluator considers that the information provided in the new release of the security target and "entropy validation" document 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.

12.3. Test Activity

The assurance activity does not require testing activities. Therefore the **PASS** verdict is assigned.

12.4. Final Verdict

Due to all activities 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 this FCS_RBG_EXT.1.2 requirement.

13. FCS_STO_EXT.1

13.1. Assurance activity

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.

13.2. Documentation review activity

13.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.2.3 Protecting Data with DPAPI.** This section states that Windows provides the Data Protection API, DPAPI, it is used to protect any persisted data which the developer deems to be sensitive using the CryptProtectData and CryptUnprotectData interfaces. DPAPI use AES(256)-CBC algorithm for encryption and decryption.

In addition the vendor has made a pointer to the “msdn” where detail information about the usage of these API function is provided. The following images obtained from the “msdn” describe the syntax of these API functions:

```
BOOL WINAPI CryptProtectData(
  _In_       DATA_BLOB                 *pDataIn,
  _In_opt_   LPCWSTR                   szDataDescr,
  _In_opt_   DATA_BLOB                 *pOptionalEntropy,
  _Reserved_ PVOID                     pvReserved,
  _In_opt_   CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  _In_       DWORD                     dwFlags,
  _Out_      DATA_BLOB                 *pDataOut
);
```

```
BOOL WINAPI CryptUnprotectData(
  _In_       DATA_BLOB                 *pDataIn,
  _Out_opt_  LPWSTR                    *ppszDataDescr,
  _In_opt_   DATA_BLOB                 *pOptionalEntropy,
  _Reserved_ PVOID                     pvReserved,
  _In_opt_   CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  _In_       DWORD                     dwFlags,
  _Out_      DATA_BLOB                 *pDataOut
);
```

13.2.2. Verdict

The evaluator considers that the information provided in the TSS describes the method used for sensitive data protection. It has also been identified the available interfaces for data protection (CryptProtectData and CryptUnprotectData) as well as cryptographic algorithm used (AES(256)-CBC) which is one of the selected in the FCS_COP.1(SYM) requirement.

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

13.3. Test Activity

The assurance activity does not require testing activities. However the evaluator has carried out the following test in order to check the API functions functionality:

```
#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);

//-------------------------------------------------------------------
//  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.

//--------------------------------------------------------------------
//  Clean up.

LocalFree(pDescrOut);
LocalFree(DataOut.pbData);
LocalFree(DataVerify.pbData);
} // 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
```

The evaluator has verified that the API functions work as expected. Therefore the **PASS** verdict is assigned.

13.4. Final Verdict

Due to all activities 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 this FCS_STO_EXT.1 requirement.

14. FCS_TLSC_EXT.1.1

14.1. Assurance activity

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 128bit AES and not 256bit 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 nonsupported 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. The evaluator will verify that the client rejects the connection after receiving the Server Hello.*

- ***Test 5.4:*** *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.*

14.2. Documentation review activity

14.2.1. Findings

In the *"section 6.2.2"* of the TSS of the *"Windows 10 Security Target"* document, specifies a *"http://msdn.microsoft.com/en-us/library/windows/desktop/aa374757(v=vs.85).aspx" web* site, where describes the cipher suites supported by the "Schannel" library.

According to the security target document, the evaluated cipher suites are:

- 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
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- 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.

The cipher suites listed above are a subset of the implemented in *"Schannel"* library as it can be checked in the previous URL.

The section 4.1 of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"* document describes how to configure the TLS cipher suites. This information can be found in the following MSDN links.

- http://msdn.microsoft.com/en-us/library/windows/desktop/bb870930(v=ws.10).aspx
- http://support.microsoft.com/kb/245030

The selection of TLS cipher suite in the handshake process is performed according to the order of the cipher suites in the "SSL Cipher Suite Order" and "ECC Curve Order". The URLs listed above describes how configure these files to select a certain cipher suites and their order.

In addition, the section 4.1 of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"* explains how to configure the cipher suites for Windows Server 2012 R2 to use elliptic curves. The following statement has been included:

"Server 2012 R2: The elliptic curves supported for a particular cipher suite are part of the cipher suite configuration. For example in the table above one of the supported cipher suites is TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256. To enable this cipher suite with an elliptic curve, e.g. secp256r1, it needs to be used the SSL cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256. The difference is the final four characters which indicate the elliptic curve that is to be used where P256 is equals to secp256r1"

14.2.2. 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.

14.3. Test Activity

14.3.1. Test 1

14.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"createCertificate.sh"* and *"createCertificateECDSA.sh <curveType>"* scripts.

The first script generates a certificate with RSA algorithm. The second script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall generate three certificates with the following curves: *"secp256r1"* and *"secp384r1"* and RSA.

- certificate (RSA)
- certificate (secp256r1)
- certificate (secp384r1)

The scripts listed above, generate the certificates in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following items:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* and *"createCertificateECDSA.sh <curveType>"* scripts shall be copied in the Support Machine.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

14.3.1.2. Procedure

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

1. Import the RSA certificate with *".pfx"* format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
2. Run the *"testCipherSuite.ps1 RSA"* script on the Client Machine, this script forces to Client Machine to use a certain cipher suite. The tested cipher suites are:
    - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_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
3. Open a *"Wireshark"* application to verifies that the handshake operation is performed correctly.
4. Open the browser and attempt to navigate to the test web (https://192.168.1.101).
5. Import the *"sepc256r1"* certificate in *".pfx"* format created in the Client Machine.

- Open *"Server Manager"* from the task bar.
- Click *"IIS"*.
- Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
- Double click in *"Server Certificates"*.
- Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
- Click *"OK"*.
- Expand *"Sites"* and click *"Default Web Site"*.
- Click *"Bindings..."*.
- Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.

6. Run the script *"testCipherSuite.ps1 sepc256r1"* on the Client Machine, this script forces to Client Machine to use a certain cipher suite. The tested cipher suites are:
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
   - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
7. Open a *"Wireshark"* application to verifies that the handshake operation is performed correctly.
8. Open the browser and attempt to navigate to the test web (https://192.168.1.101).
9. Import the *"secp384r1"* certificate created in *".pfx"* format in the Client Machine.
   - Open *"Server Manager"* from the task bar
   - Click *"IIS"*.
   - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
   - Double click in *"Server Certificates"*.
   - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
   - Click *"OK"*.
   - Expand *"Sites"* and click *"Default Web Site"*.
   - Click *"Bindings..."*.

- Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.

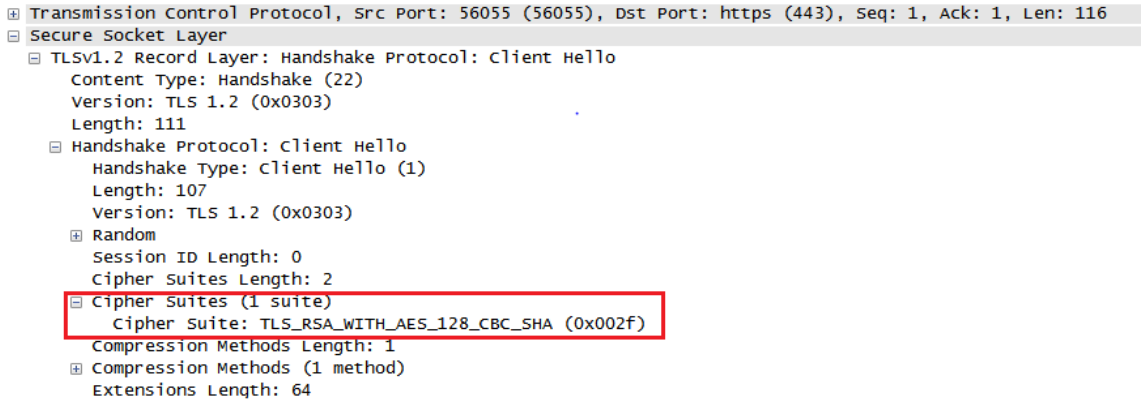
10. Run the script *"testCipherSuite.ps1 sepc384r1"* on the client machine, this script forces to client machine to use a certain cipher suites. The tested cipher suites are:
    - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
11. Open a *"Wireshark"* application to verifies that the handshake operation is performed correctly.
12. Open the browser and attempt to navigate to the test web (https://192.168.1.101).

14.3.1.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that, for each cipher suite, the TLS connection is established successfully. The cipher suite selected in the *"testCipherSuite.ps1"* script (e.g. TLS_RSA_WITH_AES_256_CBC_SHA) can be appreciated in the following picture:



In addition, the server hello handshake message response with the same cipher suite that the client, as it is shown in the next picture.

```
Secure Socket Layer
  TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 853
    Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 77
      Version: TLS 1.2 (0x0303)
      Random
      Session ID Length: 32
      Session ID: F11200002444E10A6785762C64BE08D7170ECD227C2CEC7A...
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Compression Method: null (0)
      Extensions Length: 5
      Extension: Unknown 65281
    Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 764
      Certificates Length: 761
      Certificates (761 bytes)
    Handshake Protocol: Server Hello Done
      Handshake Type: Server Hello Done (14)
      Length: 0
```

On the other hand, in the browser the evaluator can confirm that the connection has been established correctly, as it is shown in the following picture:

14.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.2. Test 2

14.3.2.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"createCertificate.sh <configurationFile>"* script. This script generates a certificate with RSA algorithm with a certain *"extendedKeyUsage"* purpose.

- certificate (RSA) with "Server Auth"
- certificate (RSA) with "Code Signing"

The script listed above, generates the certificates in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Kali Linux)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.100
- Client Machine, IP = 192.168.1.102

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh <configurationFile>", "opensslServer.sh <certificate.pem> <key.pem>"* and the *"openssl"* configuration files shall be copied in the Support Machine.

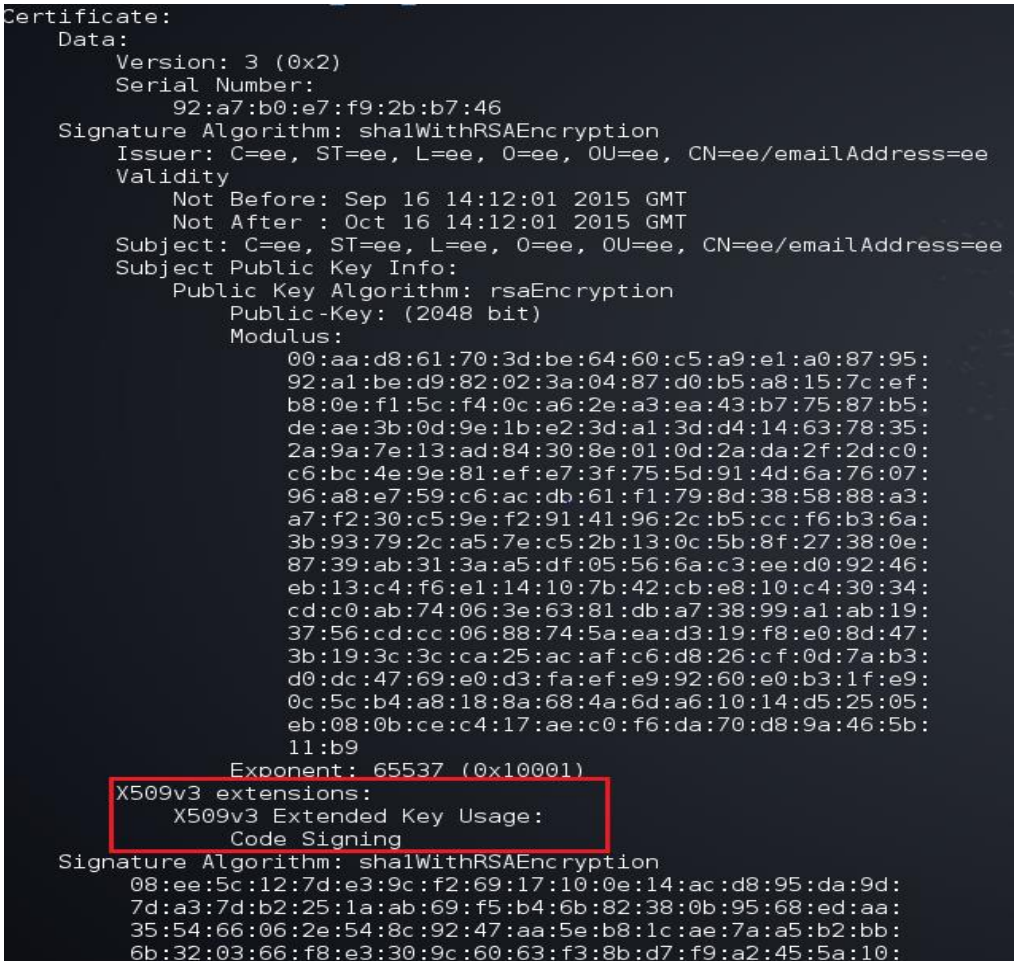
14.3.2.2. Procedure

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

1. Open a terminal in the Server Machine and checks that the certificate.pem contains the *"extendedKeyUsage"* purpose *"serverAuth"* using *"openssl x509 -text -noout -in certificate.pem"* command, as it can be appreciated in the next picture.

2. Open a terminal in the Server Machine and type the following commands:
   - *"cd Desktop"*
   - *"chmod 777 opensslServer.sh"*
   - *"./opensslServer.sh certificate.pem private.pem"*
3. Open a *"Wireshark"* application in the Client Machine to verify that the handshake operation is performed correctly.

4. Open the browser and attempt to navigate to the test web (https://192.168.1.100:4433).
5. Stop *"Wireshark"* and *"openssIServer"* applications.
6. Open a terminal in the Server Machine and checks that the *"certificate.pem"* contains the *"extendedKeyUsage"* purpose *"codeSigning"* using *"openssl x509 -text -noout -in certificate.pem"* command, as it can be appreciated in the next picture.



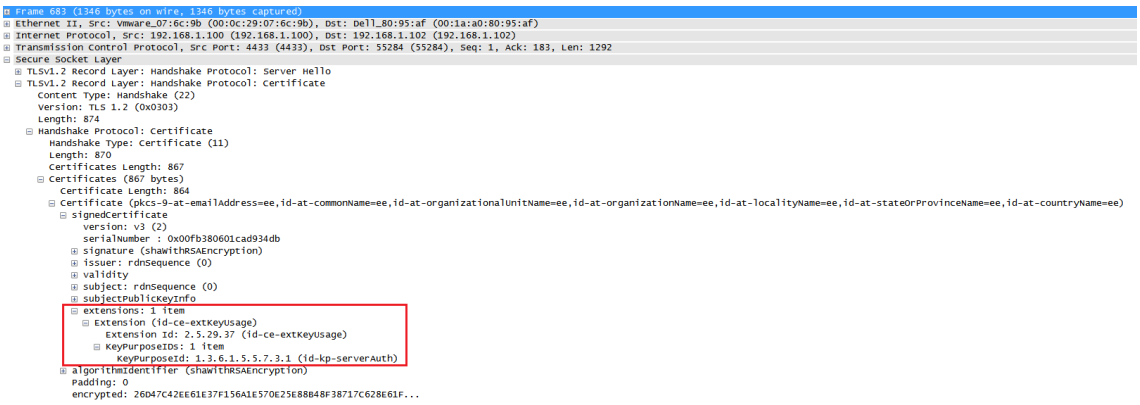
7. Open a terminal in the Server Machine and type the following commands:
   - *"cd Desktop"*
   - *"chmod 777 openssIServer.sh"*
   - *"./openssIServer.sh certificate.pem private.pem"*
8. Open a *"Wireshark"* application in the Client Machine to verify that the handshake operation is not performed correctly.
9. Open the browser and attempt to navigate to the test web (https://192.168.1.100:4433).

14.3.2.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86

Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that the connection is established correctly using a certificate with a *"serverAuth"* purpose in the *"extendedKeyUsage"* extension. The following picture shows the *"Server Certificate"* message that contains the *"serverAuth"* purpose.



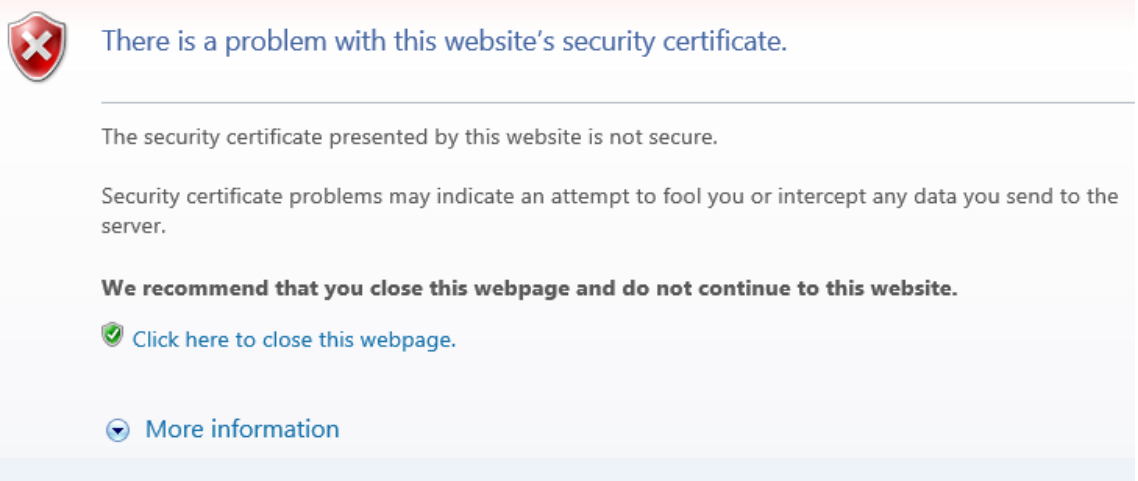
When the connection is established, the server send the following picture to the browser.

```
s_server -www -cert certificate.pem -key certificateKey.pem
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1/SSLv3:ECDHE-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA384   TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDHE-RSA-AES256-SHA      TLSv1/SSLv3:ECDHE-ECDSA-AES256-SHA
TLSv1/SSLv3:SRP-DSS-AES-256-CBC-SHA   TLSv1/SSLv3:SRP-RSA-AES-256-CBC-SHA
TLSv1/SSLv3:DHE-DSS-AES256-GCM-SHA384TLSv1/SSLv3:DHE-RSA-AES256-GCM-SHA384
TLSv1/SSLv3:DHE-RSA-AES256-SHA256     TLSv1/SSLv3:DHE-DSS-AES256-SHA256
TLSv1/SSLv3:DHE-RSA-AES256-SHA        TLSv1/SSLv3:DHE-DSS-AES256-SHA
TLSv1/SSLv3:DHE-RSA-CAMELLIA256-SHA   TLSv1/SSLv3:DHE-DSS-CAMELLIA256-SHA
TLSv1/SSLv3:ECDH-RSA-AES256-GCM-SHA384TLSv1/SSLv3:ECDH-ECDSA-AES256-GCM-SHA384
TLSv1/SSLv3:ECDH-RSA-AES256-SHA384    TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA384
TLSv1/SSLv3:ECDH-RSA-AES256-SHA       TLSv1/SSLv3:ECDH-ECDSA-AES256-SHA
TLSv1/SSLv3:AES256-GCM-SHA384         TLSv1/SSLv3:AES256-SHA256
TLSv1/SSLv3:AES256-SHA                TLSv1/SSLv3:CAMELLIA256-SHA
TLSv1/SSLv3:PSK-AES256-CBC-SHA        TLSv1/SSLv3:ECDHE-RSA-DES-CBC3-SHA
TLSv1/SSLv3:ECDHE-ECDSA-DES-CBC3-SHA TLSv1/SSLv3:SRP-DSS-3DES-EDE-CBC-SHA
TLSv1/SSLv3:SRP-RSA-3DES-EDE-CBC-SHA TLSv1/SSLv3:EDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:EDH-DSS-DES-CBC3-SHA      TLSv1/SSLv3:ECDH-RSA-DES-CBC3-SHA
TLSv1/SSLv3:ECDH-ECDSA-DES-CBC3-SHA   TLSv1/SSLv3:DES-CBC3-SHA
TLSv1/SSLv3:PSK-3DES-EDE-CBC-SHA      TLSv1/SSLv3:ECDHE-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDHE-ECDSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDHE-RSA-AES128-SHA256
TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA256TLSv1/SSLv3:ECDHE-RSA-AES128-SHA
TLSv1/SSLv3:ECDHE-ECDSA-AES128-SHA    TLSv1/SSLv3:SRP-DSS-AES-128-CBC-SHA
TLSv1/SSLv3:SRP-RSA-AES-128-CBC-SHA   TLSv1/SSLv3:DHE-DSS-AES128-GCM-SHA256
TLSv1/SSLv3:DHE-RSA-AES128-GCM-SHA256TLSv1/SSLv3:DHE-RSA-AES128-SHA256
TLSv1/SSLv3:DHE-DSS-AES128-SHA256     TLSv1/SSLv3:DHE-RSA-AES128-SHA
TLSv1/SSLv3:DHE-DSS-AES128-SHA        TLSv1/SSLv3:DHE-RSA-SEED-SHA
TLSv1/SSLv3:DHE-DSS-SEED-SHA          TLSv1/SSLv3:DHE-RSA-CAMELLIA128-SHA
TLSv1/SSLv3:DHE-DSS-CAMELLIA128-SHA   TLSv1/SSLv3:ECDH-RSA-AES128-GCM-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-GCM-SHA256TLSv1/SSLv3:ECDH-RSA-AES128-SHA256
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA256 TLSv1/SSLv3:ECDH-RSA-AES128-SHA
TLSv1/SSLv3:ECDH-ECDSA-AES128-SHA     TLSv1/SSLv3:AES128-GCM-SHA256
TLSv1/SSLv3:AES128-SHA256             TLSv1/SSLv3:AES128-SHA
TLSv1/SSLv3:SEED-SHA                  TLSv1/SSLv3:CAMELLIA128-SHA
TLSv1/SSLv3:PSK-AES128-CBC-SHA        TLSv1/SSLv3:ECDHE-RSA-RC4-SHA
TLSv1/SSLv3:ECDHE-ECDSA-RC4-SHA       TLSv1/SSLv3:ECDH-RSA-RC4-SHA
TLSv1/SSLv3:ECDH-ECDSA-RC4-SHA        TLSv1/SSLv3:RC4-SHA
TLSv1/SSLv3:RC4-MD5                   TLSv1/SSLv3:PSK-RC4-SHA
TLSv1/SSLv3:EDH-RSA-DES-CBC-SHA       TLSv1/SSLv3:EDH-DSS-DES-CBC-SHA
TLSv1/SSLv3:DES-CBC-SHA               TLSv1/SSLv3:EXP-EDH-RSA-DES-CBC-SHA
TLSv1/SSLv3:EXP-EDH-DSS-DES-CBC-SHA   TLSv1/SSLv3:EXP-DES-CBC-SHA
TLSv1/SSLv3:EXP-RC2-CBC-MD5           TLSv1/SSLv3:EXP-RC4-MD5
---
Ciphers common between both SSL end points:
```

On the other hand, the Client Machine rejects the connection when the server certificate contains the *"extendedKeyUsage"* extension with the purpose *"codeSigning".* The following captures demonstrate that the certificate contains the purpose *"codeSigning"* and the browser rejects the connection.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 874
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 870
      Certificates Length: 867
    ⊟ Certificates (867 bytes)
        Certificate Length: 864
      ⊟ Certificate (pkcs-9-at-emailAddress=ee,id-at-commonName=ee,id-at-organizationa
        ⊟ signedCertificate
            version: v3 (2)
            serialNumber: -5908056846347319539
          ⊞ signature (shaWithRSAEncryption)
          ⊞ 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)
        ⊞ algorithmIdentifier (shaWithRSAEncryption)
          Padding: 0
```

When the Client Machine receives the certificate message, it rejects the communication. The error message that shows the browser can be appreciated in the next picture.



14.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.3. Test 3

14.3.3.1. Setup

The following certificate shall be used to perform the assurance activities listed in the Protection Profile. The certificate listed below has been created using the *"createCertificateECDSA.sh secp256r1"* script. This script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall a certificate with the following curve: *"secp256r1"*.

- certificate (secp256r1)

The script listed above generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) must be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select "Role-based or features-based" installation from the "Installation Type" screen, and click next.
- The current server is selected by default. Click next.
- From the "Server Roles" screen check a mark in the box "Web Server (IIS)". An additional pop-up screen must appear explain all the features required to install the Domain Services. Click "Add features". A new screen is shown and click next.
- On "Select features", Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services. Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

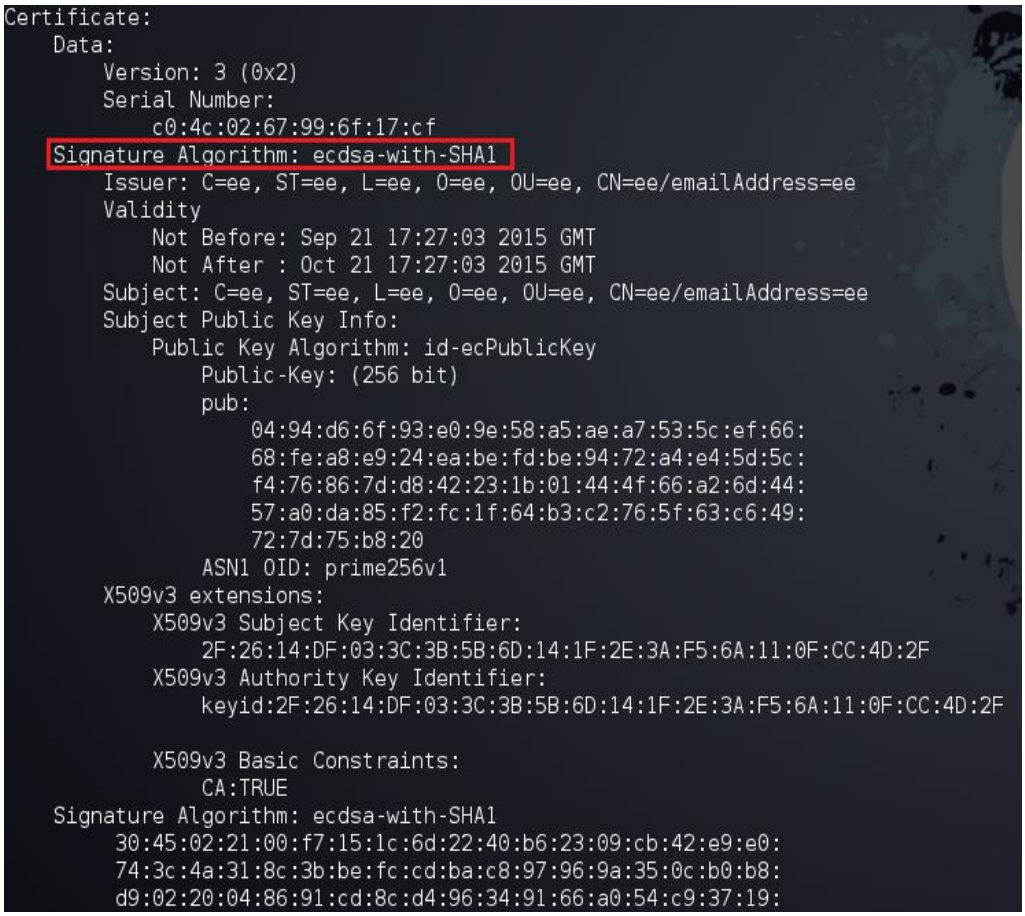
The *"createCertificateECDSA.sh"* script shall be copied in the Support Machine.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

14.3.3.2. Procedure

In order to perform this test, the evaluator has followed the steps listed below:

1. Run the script *"testCipherSuite.ps1 TLS_RSA_WITH_AES_128_CBC_SHA"* in the Client Machine.
2. Open a terminal in the Server Machine and checks that the certificate.pem has been created with the *"secp256r1"* curve. This information can be displayed using *"openssl x509 -text -noout -in certificate.pem"* command, as it can be appreciated in the next picture.



3. Import the *"secp256r1"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - *Click "Add", "type: https", "IP address: 192.168.1.101" and in the "SSL Certificate"* load the certificate previously loaded.

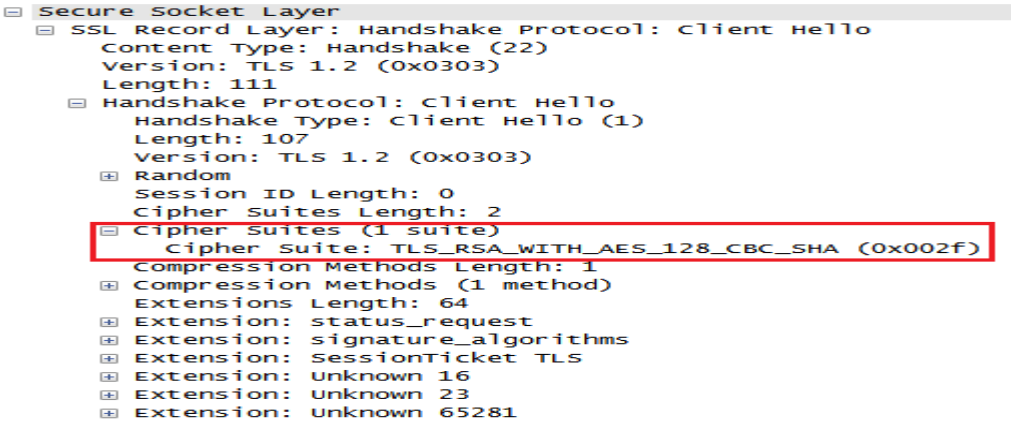
4. Open a *"Wireshark"* application to verifies that the handshake operation is not performed correctly.
5. Open the browser and attempt to navigate to the test web (https://192.168.1.101).

14.3.3.3. Results

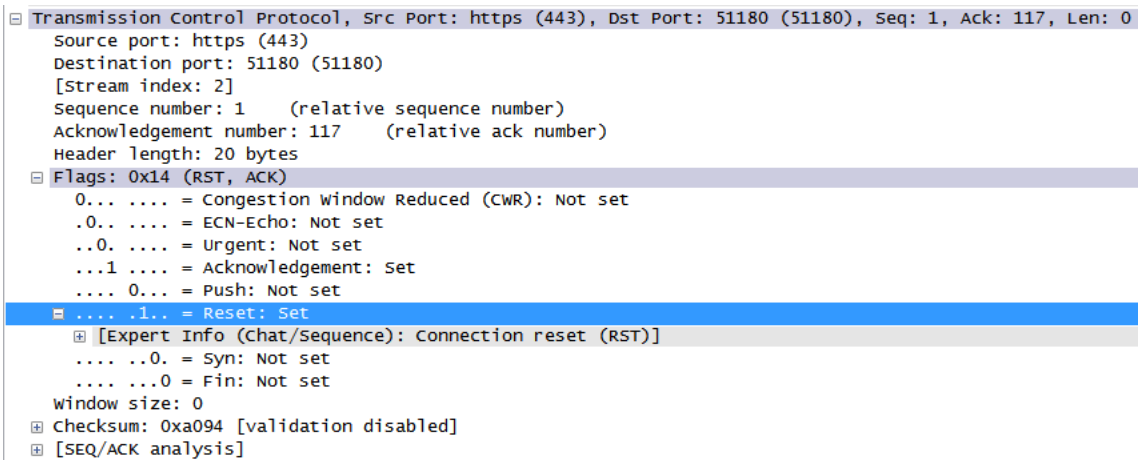
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

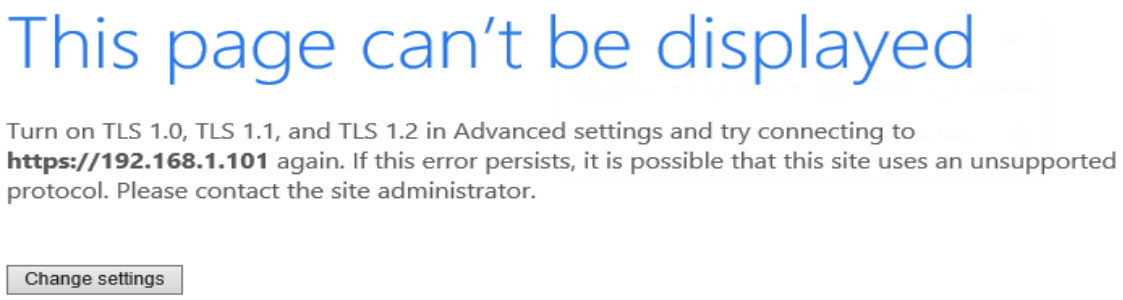
Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that, the connection is not established correctly using a *"ECDSA"* certificate. The following picture shows "Client Hello" message and the cipher suite used by the client.



The server response with a *"RST, ACK"* packet, as it is shown in the following picture:



In addition, the browser shows to the user the following screen when the Client Machine receives the reset packet.



14.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.4. Test 4

14.3.4.1. Setup

The following certificate shall be used to perform the assurance activities listed in the Protection Profile. The certificate listed below have been created using the *"createCertificate.sh"* script. This script generates a certificate with RSA algorithm.

- certificate (RSA)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

14.3.4.2. Procedure

In order to perform this test, the evaluator has followed the steps listed below:

1. In the Server Machine, left click on *"Start"*, press *"search"* and write *"Run"*.
2. In the Run application write *"gpedit.msc"* and press *"OK"* button.
3. Expand *"Administrative Templates -> Network "* in *"Local Group Policy Editor"* and open the file SSL Cipher Suite Order.
4. Click on the radio button *"Enable"*. In the text box SSL Cipher Suites, erases all cipher suites and write *"TLS_NULL_WITH_NULL_NULL"*.
5. Press *"Apply"* and reboot the Server Machine.
6. Import the *"RSA"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.

- Double click in *"Server Certificates"*.
- Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
- Click *"OK"*.
- Expand *"Sites"* and click *"Default Web Site"*.
- Click *"Bindings..."*.
- *Click "Add", "type: https", "IP address: 192.168.1.101" and in the "SSL Certificate"* load the certificate previously loaded.

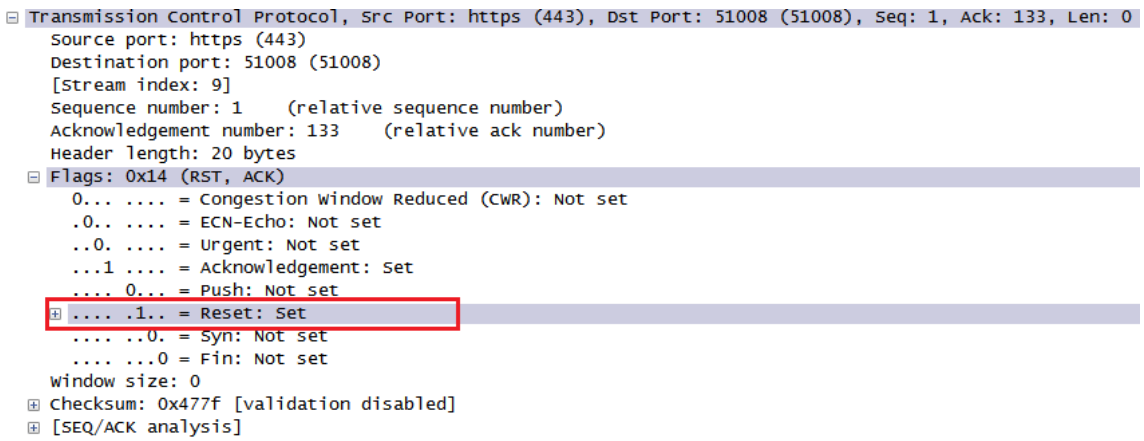
6. Open in the Client Machine a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
7. Open the browser and attempt to navigate to the test web (https://192.168.1.101).

14.3.4.3. Results

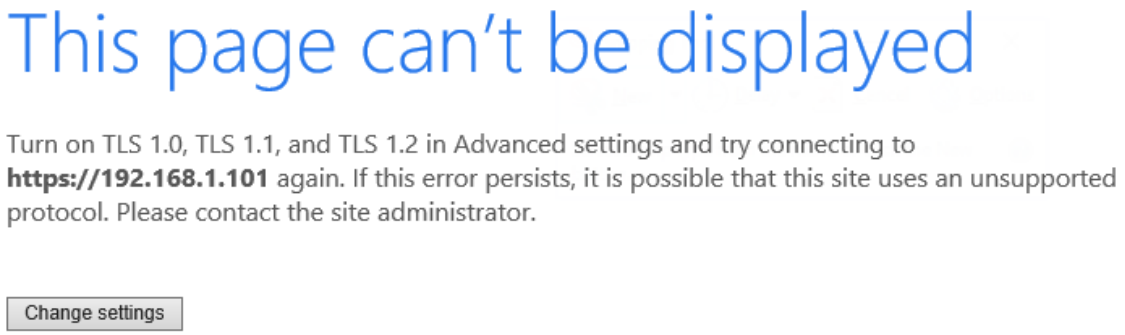
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that, when the Client Machine sends the *"Client Hello"* message, the server returns a *"RSA, ACK"* packet. Therefore the server machine does not start the handshake process. The next picture shows the reset packet.



In addition, the browser shows to the user the following screen when the Client Machine receives the reset packet.



14.3.4.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.5. Test 5.1

14.3.5.1. Setup

The following certificate shall be used to perform the assurance activities listed in the Protection Profile. The certificate listed below has been created using the *"createCertificate.sh"* script. This script generates a certificate with RSA algorithm.

- certificate (RSA)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificate.sh"* script shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.5.2. Procedure

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

1. Import the *"RSA"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.

- Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
- Click *"OK"*.
- Expand *"Sites"* and click *"Default Web Site"*.
- Click *"Bindings..."*.
- Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the *"SSL Certificate"* load the certificate previously loaded.

2. In the MITM Machine open a terminal and type the followings commands:
   - *"cd Desktop/SSL_Proxy"*
   - *"chmod 777 run_mitm"*
   - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).

14.3.5.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

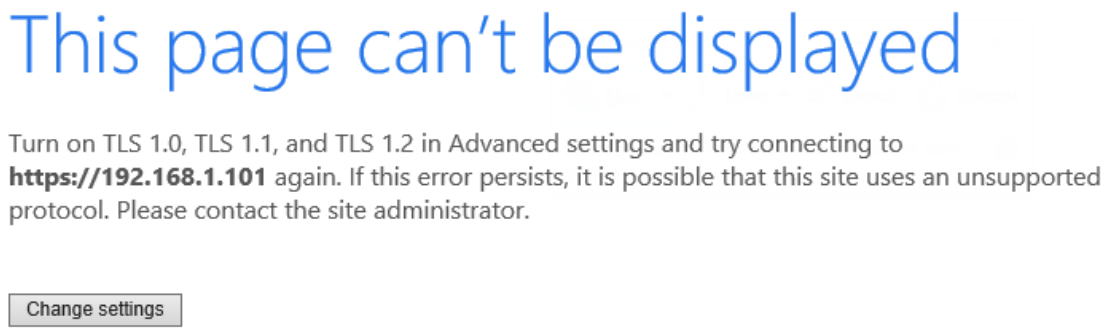
Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_ssl_version(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset #Ethernet header + IP header + Transmission header

    if packet[0] == '\x16':
        indexPacket += 2
        newPacket = packet[:indexPacket] + '\x04' + packet[indexPacket + 1:]  #TLS version modified

    return newPacket
```

In the *"server hello"* message can be appreciated the modified version *"0x0304"*, as it can be appreciated in the next picture:

```
Secure Socket Layer
  TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1349
    Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 77
      Version: Unknown (0x0304)
      Random
      Session ID Length: 32
      Session ID: 5D020000BB45C84195EFE24B89A565C1047F2D8FCAF1A4D4...
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
      Compression Method: null (0)
      Extensions Length: 5
      Extension: Unknown 65281
    Handshake Protocol: Certificate
    Handshake Protocol: Server Key Exchange
    Handshake Protocol: Server Hello Done
```

After the client machine receives the *"server hello"* message, the browser shows the next picture.



14.3.5.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.6. Test 5.2

14.3.6.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"createCertificate.sh"* and *"createCertificateECDSA.sh <curveType>"* scripts. The first script

generates a certificate with RSA algorithm. The second script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall generate two certificates, one with the *"secp256r1"* curve and a RSA certificate.

- certificate (RSA)
- certificate (secp256r1)

The scripts listed above, generate the certificates in *"pem"* and *"pkcs12"* formats. The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificate.sh"* and *"createCertificateECDSA.sh"* scripts shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.6.2. Procedure

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

1. Import the *"RSA"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
2. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.

4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).
5. Import the *"secp256r1"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
6. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
7. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
8. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101)

14.3.6.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_nonce(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 2 #tls version
        indexPacket += 4 #tls gtm_unix_time
        indexPacket += 1 #first byte of random_bytes

        newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

The first part of this test uses the cipher suite *"TLS_RSA_WITH_AES_128_CBC_SHA"*. The next capture shows the *"Server Hello"* packet before the modification of the first byte into the nonce structure.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1016
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.2 (0x0303)
    Random
      gmt_unix_time: Oct 26, 2015 10:24:26.000000000
      random_bytes: 21F77F7741F982ACE680AA3020537D1617525FF2FC86AB4B...
    Session ID Length: 32
    Session ID: 7B480000B76A5495104D2F6B34DD0DD32C1AD3EEADD21827...
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
```

The modification performed the packet can be appreciated in the following picture.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1016
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.2 (0x0303)
    Random
      gmt_unix_time: Oct 26, 2015 10:24:26.000000000
      random_bytes: AAF77F7741F982ACE680AA3020537D1617525FF2FC86AB4B...
    Session ID Length: 32
    Session ID: 7B480000B76A5495104D2F6B34DD0DD32C1AD3EEADD21827...
```

When the server machine receives the packet *"Client Key exchange"*, it is send a reset packet.

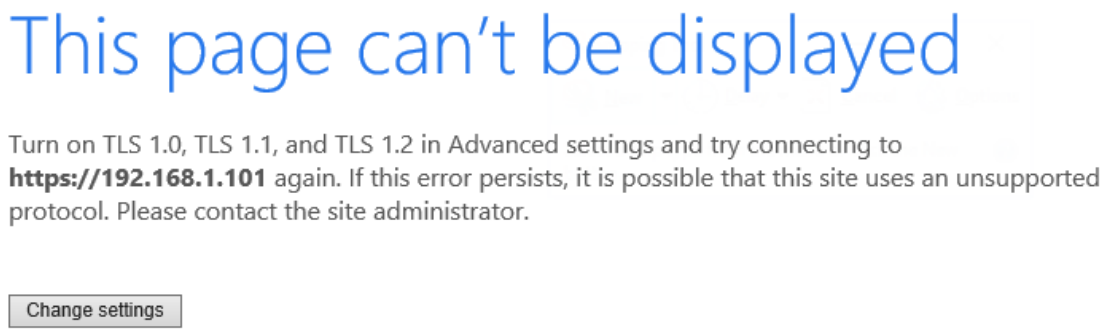
The second part of the test uses the cipher suite *"TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256"*. The following capture displays the *"Server Hello"* before the modification of the packet.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 766
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.2 (0x0303)
    Random
      gmt_unix_time: Oct 26, 2015 10:45:20.000000000
      random_bytes: E7DA183963BB87B720EA13D4EDB4B95EE9BC8F15D1530AB6...
    Session ID Length: 32
    Session ID: AC160000749E3E5AA631EC8AB41060209E1D8A40D214F8B5...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
```

After the modification, the *"Server Hello"* packet can be appreciated in the next picture.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 766
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.2 (0x0303)
    Random
      gmt_unix_time: Oct 26, 2015 10:45:20.000000000
      random_bytes: AADA183963BB87B720EA13D4EDB4B95EE9BC8F15D1530AB6...
    Session ID Length: 32
    Session ID: AC160000749E3E5AA631EC8AB41060209E1D8A40D214F8B5...
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
```

Anyway the browser shows the following message to the user.



14.3.6.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.7. Test 5.3

14.3.7.1. Setup

The following certificate shall be used to perform the assurance activities listed in the Protection Profile. The certificate listed below has been created using the *"createCertificate.sh"* script. This script generates a certificate with RSA algorithm.

- certificate (RSA)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.

- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificate.sh"* script shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.7.2. Procedure

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

1. Import the *"RSA"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
2. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.

4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).

14.3.7.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_cipher_suite(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 2 #tls version
        indexPacket += 4 #tls gtm_unix_time
        indexPacket += 28 #random_bytes
        indexPacket += 1 #sessionID len
        indexPacket += 32 #sessionID
        indexPacket += 2 #first byte of cipher Suite
        |
        newPacket = packet[:indexPacket] + '\x3D' + packet[indexPacket + 1:] #modify the cipher Suite from 0x003C to 0x003D

    return newPacket
```

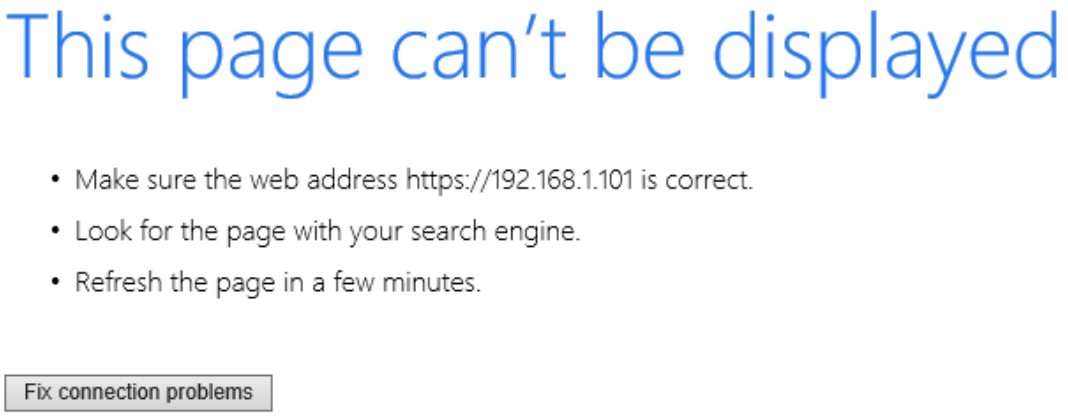
Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can assure that the modification have been performed. The following picture shows the *"Client Hello"* and the cipher suite used.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 111
  ⊟ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 107
      Version: TLS 1.2 (0x0303)
    ⊞ Random
      Session ID Length: 0
      Cipher Suites Length: 2
    ⊟ Cipher Suites (1 suite)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
```

The MITM machine modifies the cipher suite in the *"Server Hello"* message, as it can be appreciate in the next picture:

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1016
  Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 77
      Version: TLS 1.2 (0x0303)
    Random
      Session ID Length: 32
      Session ID: BC2400001AEE088568B6296F5AE4EB4E1850954EF3EE8875...
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
```

In addition, the browser shows to the user the following screen:



14.3.7.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.8. Test 5.4

14.3.8.1. Setup

The certificate listed below has been created using the *"createCertificateECDSA.sh <curveType>"* script. This script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall generate a certificate with the *"secp256r1"* curve.

- certificate (secp256r1)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificateECDSA.sh"* script shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.8.2. Procedure

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

1. Import the *"secp256r1"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the *"SSL Certificate"* load the certificate previously loaded.
2. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).

14.3.8.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

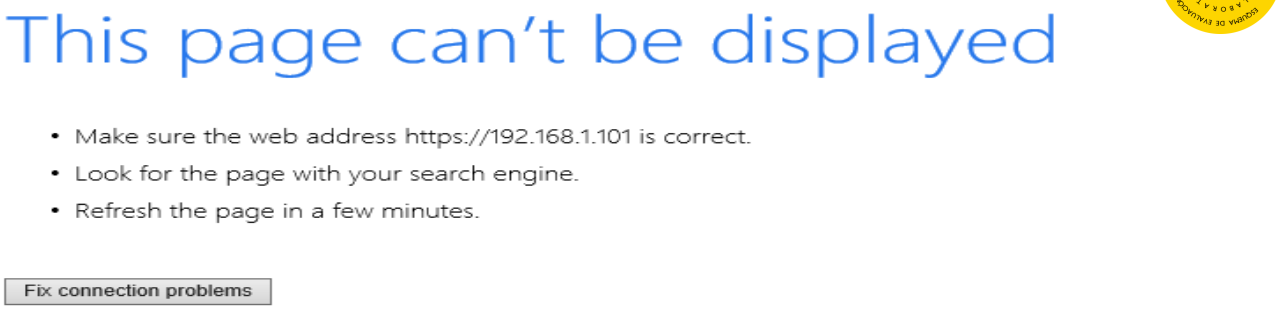
```
def modify_server_key_exchange(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':

        print "First byte:" + binascii.hexlify(packet[indexPacket])
        if packet[indexPacket] == '\x0C':
            indexPacket += 1 #server key exchange
            indexPacket += 3 #len
            indexPacket += 1 #name curve
            indexPacket += 2 #curve
            indexPacket += 1 #point len
            indexPacket += 1 #uncomprees
            indexPacket += 64 # point
            indexPacket += 1 #hash
            indexPacket += 1 #Signature
            indexPacket += 32 # random value into the signature

            print "Byte to modify:" + binascii.hexlify(packet[indexPacket])

            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

The following picture shows the *"Server Hello"* and the modification of the signature, the value introduced is "0xAA".

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 767
  ⊞ Handshake Protocol: Server Hello
  ⊞ Handshake Protocol: Certificate
  ⊟ Handshake Protocol: Server Key Exchange
      Handshake Type: Server Key Exchange (12)
      Length: 145
    ⊟ EC Diffie-Hellman Server Params
        Curve Type: named_curve (0x03)
        Named Curve: secp256r1 (0x0017)
        Pubkey Length: 65
        Pubkey: 04661e878006224c6cc391a8b9f35ecadc437128974ef432...
      ⊟ Signature Hash Algorithm: 0x0203
          Signature Hash Algorithm Hash: SHA1 (2)
          Signature Hash Algorithm Signature: ECDSA (3)
        Signature Length: 72
        Signature: 3046022100847bcf8a2713e111c8d8dc67c12fe9c5297425...
  ⊞ Handshake Protocol: Server Hello Done
```

```
0270  8c 3b be fc cd ba c8 97  96 9a 35 0c b0 b8 d9 02   .;.....  ..5.....
0280  20 04 86 91 cd 8c d4 96  34 91 66 a0 54 c9 37 19    .......  4.f.T.7.
0290  04 4b fc 90 9b 1e 75 1f  53 fd 54 2a ec c6 62 ba   .K....u.  S.T*..b.
02a0  61 0c 00 00 91 03 00 17  41 04 66 1e 87 80 06 22   a.......  A.f....."
02b0  4c 6c c3 91 a8 b9 f3 5e  ca dc 43 71 28 97 4e f4   Ll.....^  ..Cq(.N.
02c0  32 ea 8f 8b 51 3d 26 c8  dc 47 35 dd fe 47 9a 24   2...Q=&.  .G5..G.$
02d0  7d b2 44 74 fc e3 eb e6  fe c9 29 95 cc ce ca 22   }.Dt....  ..)....."
02e0  80 95 76 2c ea 21 27 f8  53 1d 02 03 00 48 30 46   ..v,.!'.  S....H0F
02f0  02 21 00 84 7b cf 8a 27  13 e1 11 c8 d8 dc 67 c1   .!..{..'  ......g.
0300  2f e9 c5 29 74 25 fc 6f  77 27 ef 09 aa 22 62 5c   /..)t%.o  w'..."b\
0310  13 6f 14 02 21 00 89 b5  85 e1 17 1b 45 42 4a 48   .o..!...  ....EBJH
0320  30 6c 30 c1 f8 09 56 79  1f bb ca a0 16 e2 b5 0a   0l0...Vy  ........
0330  74 d8 fa 31 bf 28 0e 00  00 00                     t..1.(..  ..
```

In addition, the browser shows to the user the following screen.



14.3.8.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.9. Test 5.5

14.3.9.1. Setup

The certificate listed below has been created using the *"createCertificateECDSA.sh <curveType>"* script. This script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall generate a certificate with the *"secp256r1"* following curve

- certificate (secp256r1)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.

- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificateECDSA.sh"* script shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.9.2. Procedure

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

1. Import the *"secp256r1"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.

- Click *"Bindings..."*.
- Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the *"SSL Certificate"* load the certificate previously loaded.

2. In the MITM Machine open a terminal and type the followings commands:
   - *"cd Desktop/SSL_Proxy"*
   - *"chmod 777 run_mitm"*
   - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).

14.3.9.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

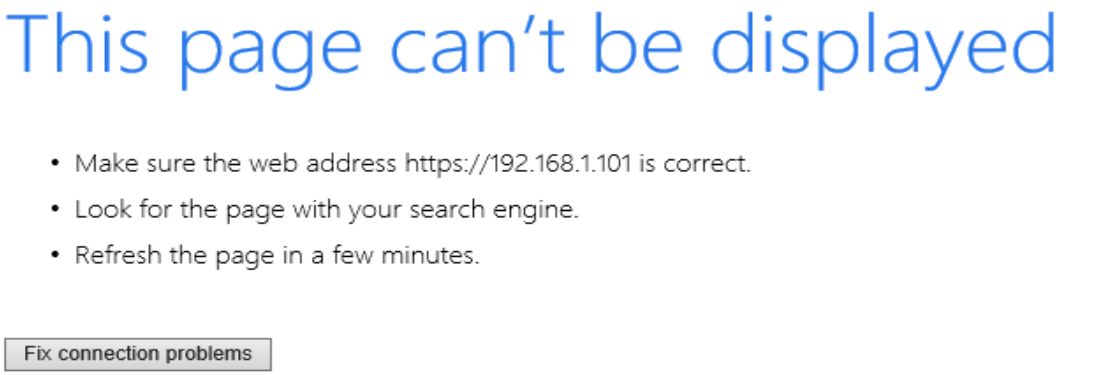
```
def modify_finished_packet(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x14':
        indexPacket += 1 #chage cipher suite
        indexPacket += 2 #tls version
        indexPacket += 2 #len
        indexPacket += 1 #cipher spec

        if packet[indexPacket] == '\x16':
            indexPacket += 1 #handshake
            indexPacket += 2 #tls version
            indexPacket += 2 #len
            indexPacket += 1 #second byte encrypted handshake message

            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

The following picture shows the *"Change Cipher Spec"* packet and modification performed.

```
⊟ 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: 80
    Handshake Protocol: Encrypted Handshake Message
```

```
0000  00 1a a0 80 95 af 00 0c  29 07 6c 9b 08 00 45 00   ........ ).l...E.
0010  00 83 72 3f 40 00 40 06  44 1a c0 a8 01 65 c0 a8   ..r?@.@. D....e..
0020  01 66 01 bb cb cf 41 77  bd e1 d4 c3 c1 72 50 18   .f....Aw .....rP.
0030  00 1f ef 48 00 00 14 03  03 00 01 01 16 03 03 00   ...H.... ........
0040  50 68 aa 40 09 66 8b 3e  20 e9 61 4d 11 f6 6e 1c   Ph.@.f.>  .aM..n.
0050  7f fb a4 22 68 2a 00 8e  7b 39 f2 b8 d4 31 dd 4d   ..."h*.. {9...1.M
0060  d0 32 fe c9 d9 53 e0 f8  52 04 c5 c3 58 e8 da 03   .2...S.. R...X...
0070  77 0b 01 42 f8 ab 8f 06  64 9b 80 27 6d 60 f0 47   w..B.... d..'m`.G
0080  24 a5 72 9a 5c 2b ed db  f1 5e e3 4e 53 de 82 b0   $.r.\+.. .^.NS...
0090  c7                                                .
```

In addition, the browser shows to the user the following screen.



14.3.9.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.3.10. Test 5.6

14.3.10.1. Setup

The certificate listed below has been created using the *"createCertificateECDSA.sh <curveType>"* script. This script generates a certificate using elliptic curve, where the input parameter is the curve type (secp256r1, secp384r1 or secp521r1). For the test purpose, the evaluator shall generate one certificate with the following curves: *"secp256r1"*.

- certificate (secp256r1)

The script listed above, generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- MITM Machine, IP = 192.168.1.100

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"testCipherSuite.ps1"* script shall be copied in the Client Machine.

The *"createCertificateECDSA.sh"* script shall be copied in the MITM Machine.

The MITM Machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

14.3.10.2. Procedure

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

1. Import the *"secp256r1"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
2. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
3. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
4. Open the browser in the Client Machine and attempt to navigate to the test web (https://192.168.1.101).

14.3.10.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

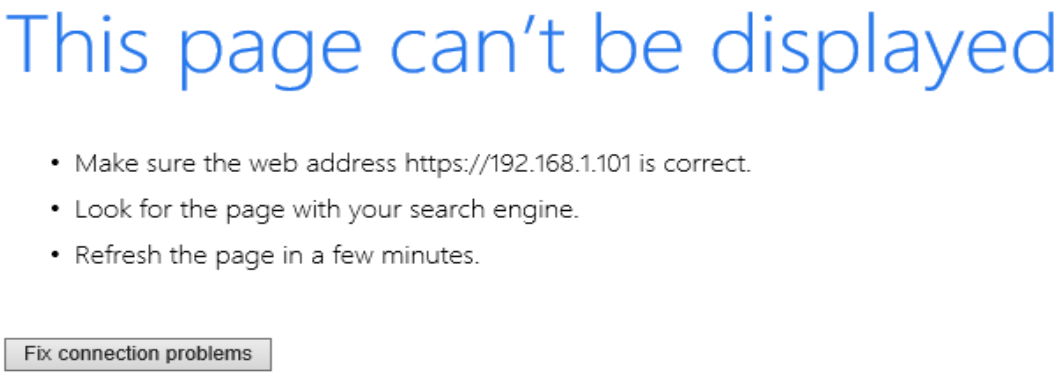
Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def send_garbled_packet(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset
    global changeCipherSpec

    if changeCipherSpec == 0:
        if packet[0] == '\x14':
            changeCipherSpec = 1

    else:
        if packet[0] == '\x17':
            indexPacket += 1 #application data
            indexPacket += 2 #tls version
            indexPacket += 2 #len
            indexPacket += 1 #second byte encrypted application data

            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]
            changeCipherSpec = 0

    return newPacket
```

The following capture shows the modification performed by the evaluator in the *"Application Data"* packet.

```
TLSv1.2 Record Layer: Application Data Protocol: spdy
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 976
    Encrypted Application Data: 75aa5f89fe3bd3371b30df7e9febaa673d9bea0fdf19d8f6...
```

In addition, the browser shows to the user the following screen.



14.3.10.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

14.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.1.1.

15. FCS_TLSC_EXT.1.2

15.1. Assurance activity

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:*** *The evaluator will present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator will verify that the connection succeeds.*

- ***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 leftmost 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 leftmost label but not preceding the public suffix (e.g. *.example.com). The evaluator will configure the reference identifier with a single leftmost label*

(e.g. foo.example.com) and verify that the connection succeeds. The evaluator will configure the reference identifier without a leftmost label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator will configure the reference identifier with two leftmost 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 leftmost label immediately preceding the public suffix (e.g. *.com). The evaluator will configure the reference identifier with a single leftmost label (e.g. foo.com) and verify that the connection fails. The evaluator will configure the reference identifier with two leftmost 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.*

15.2. Documentation review activity

15.2.1. Findings

The section 6.2.2 of the "Windows 10 Security Target" document describes the process to validate a certificate according to the Distinguished Name (DN), Subject Name (SN), or Subject Alternative Name (SAN) attributes. Also this section provides a URL where this process is detailed.

In addition, the following statement is included in the security target document:

"The reference identifier in Windows 10 and Windows Server 2012 R2 for TLS is the URL of the server. There is no configuration of the reference identifier."

Therefore, the TSS identifies clearly the reference identifier and it can be changed.

On the other hand, a certificate the 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.

Check the information provided in the TSS, AGD document or OS manual in order to find information about how to configure the OS to display the advisory warning message.

The section 4.1 of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"* states that the reference identifier for Windows 10 and Windows Server 2012 R2 for TLS is the URL of the server and there is no configuration of the reference identifier.

15.2.2. 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.

15.3. Test Activity

15.3.1. Test 1

15.3.1.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.

- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.

15.3.1.2. Procedure

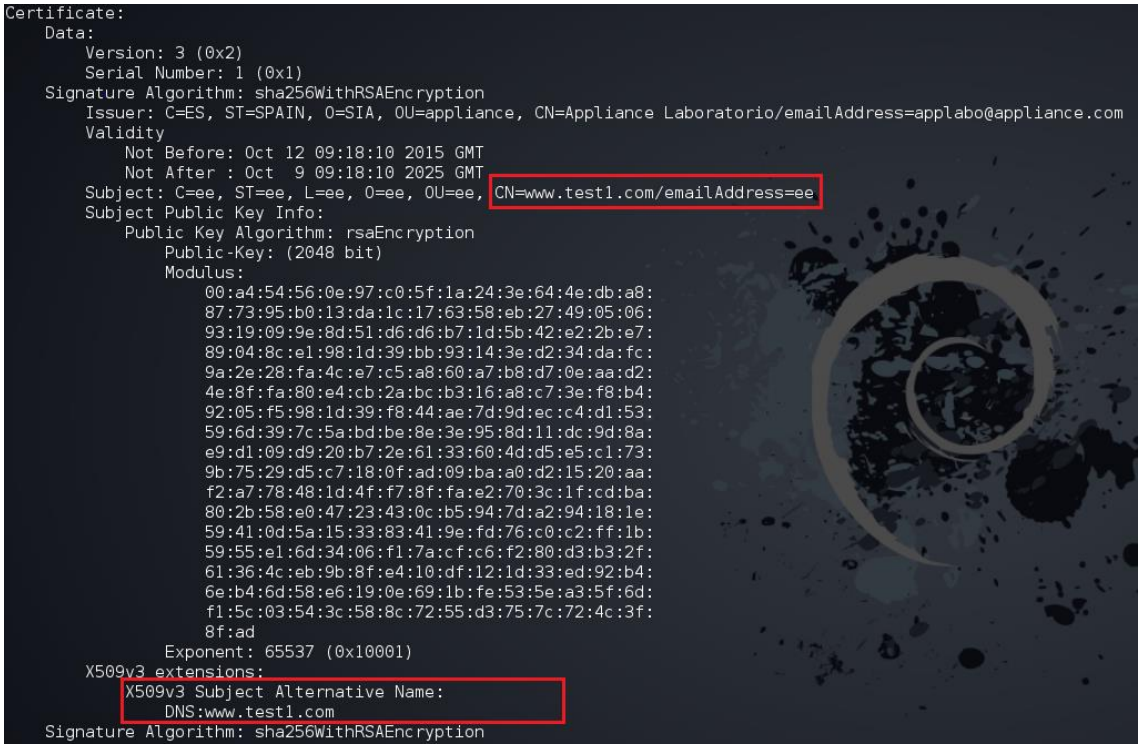
The following steps shall be performed in order to complete this test assurance activity:

1. Open a terminal in the Support Machine and type the following command:
   - *"gpedit /etc/ssl/openssl.cnf"*
2. In the *"openssl.cnf"* file, adds the following information:
   - [v3_custom]
   - subjectAltName = @alt_names

- [alt_names]
- DNS.1 = www.test1.com

3. Create a certified using the *"createCerts.sh"* script in the Support Machine, where the Common Name(CN) is *"www.test1.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (ca).

4. Check that the Common Name and Subject Alternative Name contain the URL *"www.test1.com"*, using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



5. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.

   - Open *"Server Manager"* from the task bar.
   - Click *"IIS"*.
   - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
   - Double click in *"Server Certificates"*.
   - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
   - Click *"OK"*.
   - Expand *"Sites"* and click *"Default Web Site"*.

- Click *"Bindings..."*.
- *Click "Add", "type: https", "IP address: 192.168.1.101", "Host name: www.test.com" and in the "SSL Certificate" load the certificate previously loaded*.

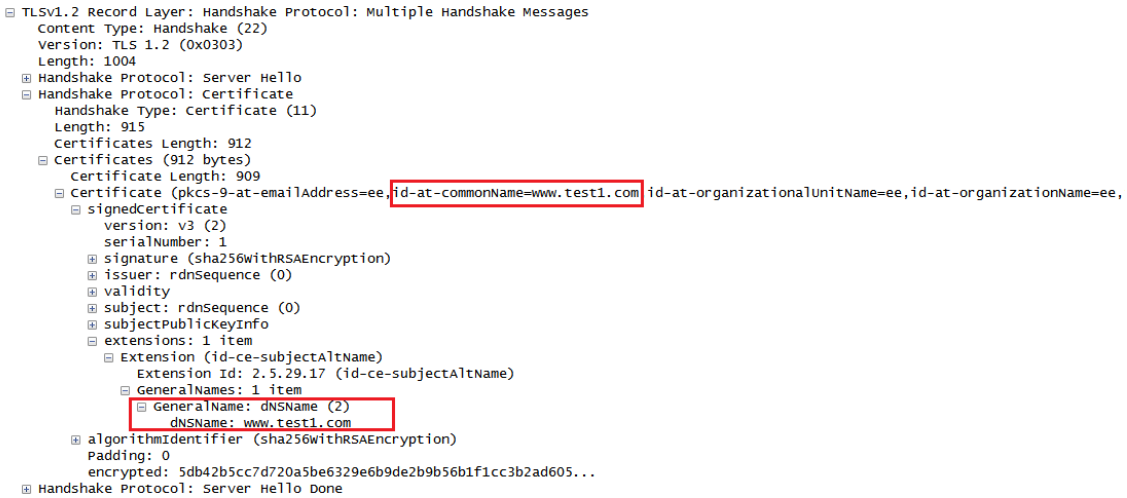
6. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
7. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.1.3. Results

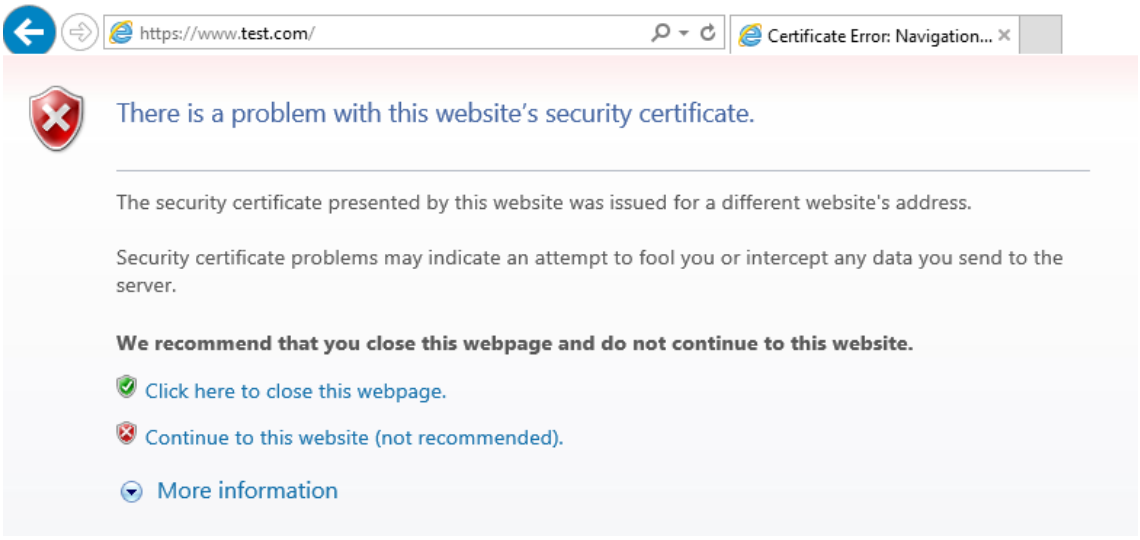
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application where it is shown the certificate sent by the server. In the next picture can be appreciate the Common Name and the Subject Alternative Name in the *"Server Hello"* message.



The following picture shows the browser response, indicating that the web address differs with the URL presented in the certificate.



15.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.2. Test 2

15.3.2.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.

15.3.2.2. Procedure

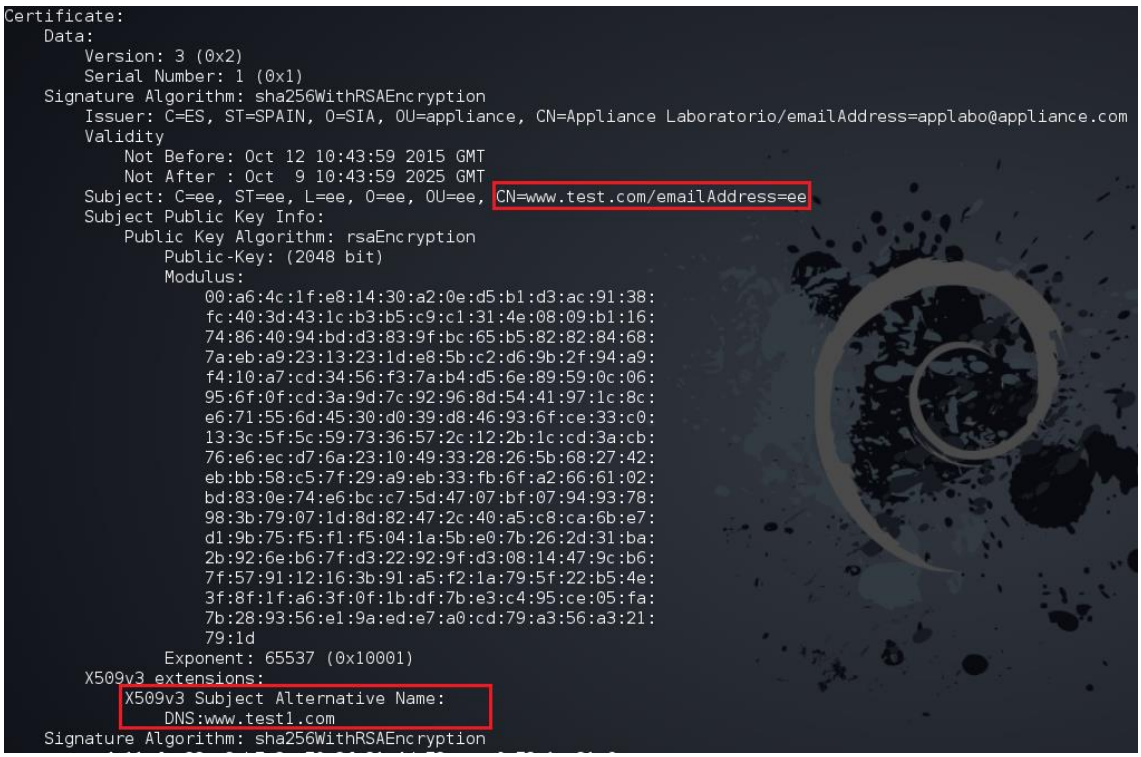
The following steps shall be performed in order to complete this test assurance activity:

1. Open a terminal in the Support Machine and type the following command:
    - *"gpedit /etc/ssl/openssl.cnf"*

2. In the *"openssl.cnf"* file, adds the following information:
    - [v3_custom]
    - subjectAltName = @alt_names
    - [alt_names]
    - DNS.1 = www.test1.com

In order to perform this test, the evaluator has followed the steps listed below:

1. Configure the "openssl.cnf" file, adding the following information:
    - [v3_custom]
    - subjectAltName = @alt_names
    - [alt_names]
    - DNS.1 = www.test1.com
2. Create a certified using the *"createCerts.sh"* script in the Support Machine, where the Common Name(CN) is *"www.test.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).
3. Check that the Common Name and Subject Alternative Name contain the URL *"www.test1.com"*, using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



4. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.

- Click *"IIS"*.
- Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
- Double click in *"Server Certificates"*.
- Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
- Click *"OK"*.
- Expand *"Sites"* and click *"Default Web Site"*.
- Click *"Bindings..."*.
- *Click "Add", "type: https", "IP address: 192.168.1.101", "Host name: www.test.com" and in the "SSL Certificate" load the certificate previously loaded*.

5. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
6. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.2.3. Results

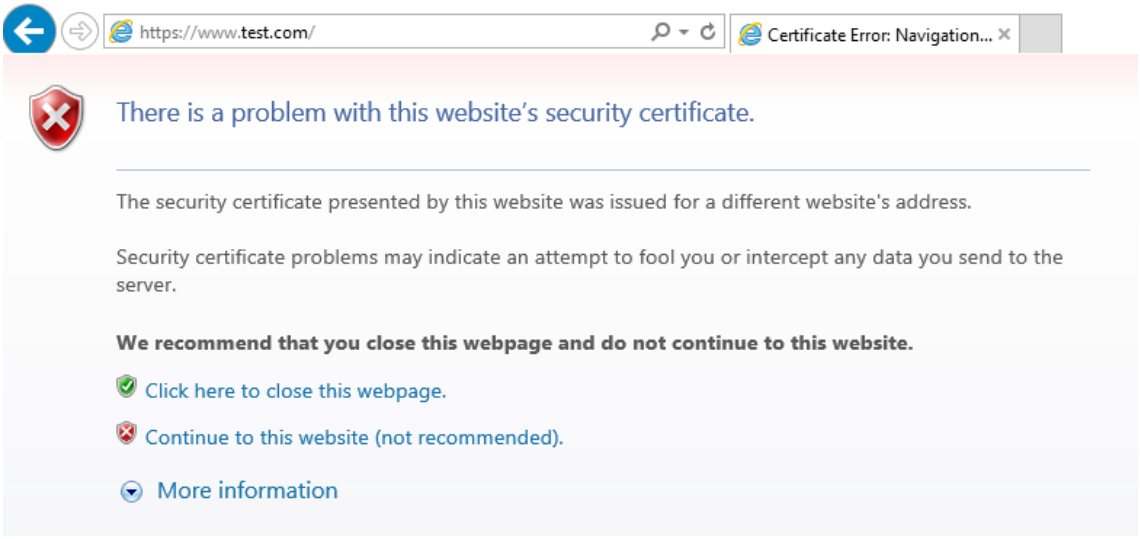
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name and the Subject Alternative Name in the *"Server Hello"* message.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1003
  Handshake Protocol: Server Hello
  Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 914
      Certificates Length: 911
    Certificates (911 bytes)
        Certificate Length: 908
      Certificate (pkcs-9-at-emailAddress=ee,id-at-commonName=www.test.com id-at-organizationalUnitName=ee,
        signedCertificate
            version: v3 (2)
            serialNumber: 1
          signature (sha256WithRSAEncryption)
          issuer: rdnSequence (0)
          validity
          subject: rdnSequence (0)
          subjectPublicKeyInfo
          extensions: 1 item
            Extension (id-ce-subjectAltName)
                Extension Id: 2.5.29.17 (id-ce-subjectAltName)
              GeneralNames: 1 item
                GeneralName: dNSName (2)
                    dNSName: www.test1.com
        algorithmIdentifier (sha256WithRSAEncryption)
```

The following picture shows the browser response, indicating that the web address differs with the URL presented in the certificate.



15.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.3. Test 3

15.3.3.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

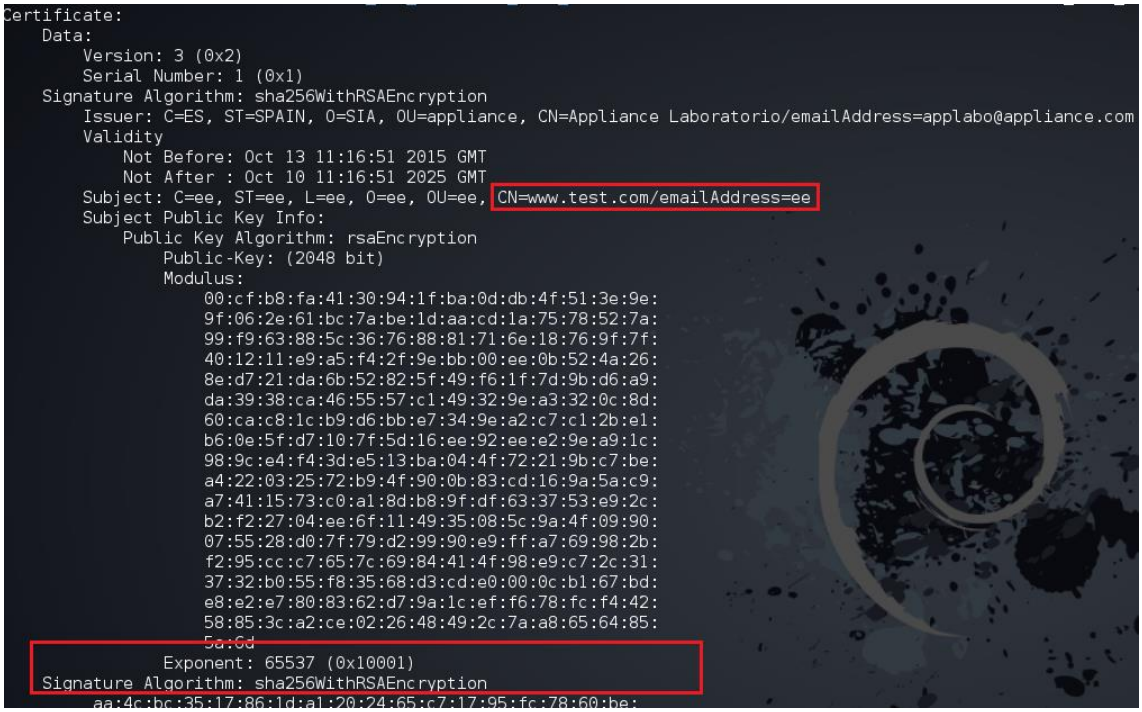
Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select "Trusted Root Certification Authorities".

15.3.3.2. Procedure

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

1. Create a certified using the *"createCerts.sh"* script where the Common Name(CN) is *"www.test.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).
2. Check that the Common Name and that the certificate does not present a Subject Alternative Name, using the *"openssl x509 -text -noout -in <certificate>.pem"* command, as it can be appreciated in the next picture.



3. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, "*type: https", "IP address: 192.168.1.101", "Host name: www.test.com"* and in the *"SSL Certificate"* load the certificate previously loaded.
4. Open a *"Wireshark"* application to verify that the handshake operation is performed correctly without user intervention.
5. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.3.3. Results

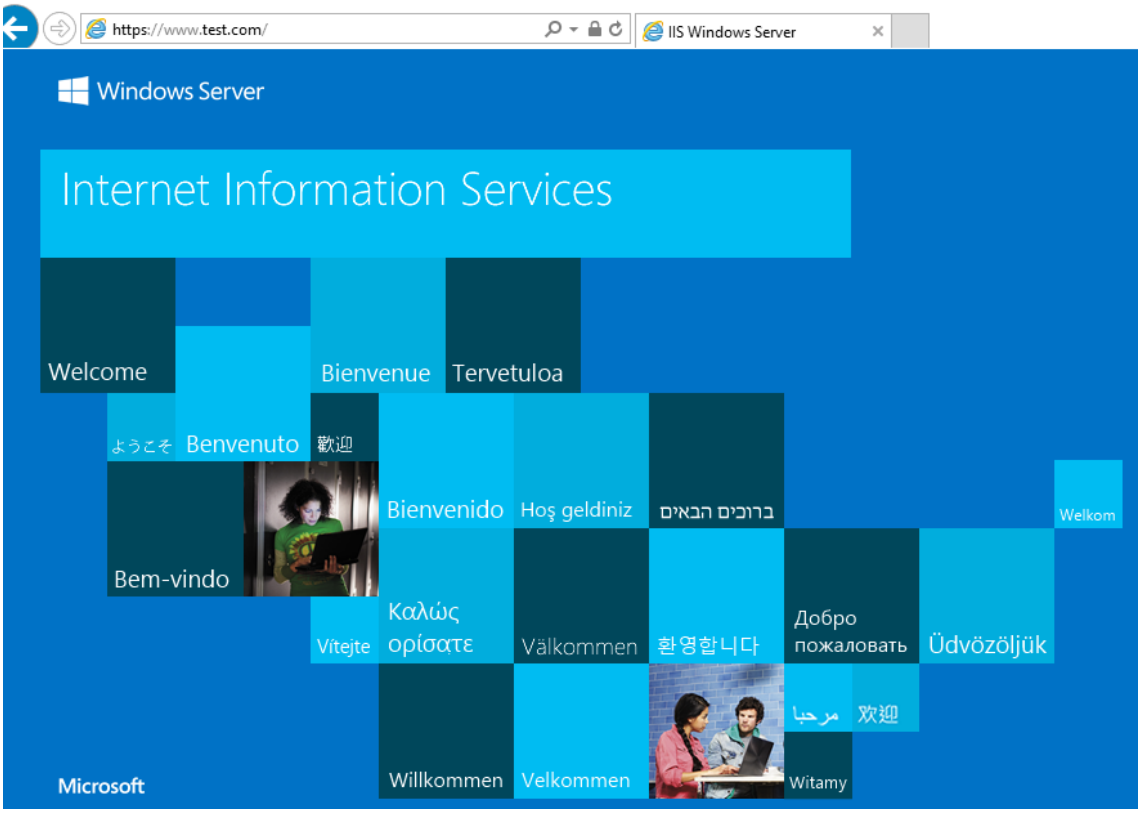
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name and the Subject Alternative Name in the *"Server Hello"* message

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 973
  Handshake Protocol: Server Hello
  Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 884
      Certificates Length: 881
    Certificates (881 bytes)
        Certificate Length: 878
      Certificate (pkcs-9-at-emailAddress=ee,id-at-commonName=www.test.com,id-at-organizationalUnitName=ee,
        signedCertificate
        algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted: aa4cbc3517861da1202465c71795fc7860be08c107e8f638...
  Handshake Protocol: Server Hello Done
      Handshake Type: Server Hello Done (14)
      Length: 0
```

The following picture shows the browser response, where the connection is performed correctly.



15.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.4. Test 4

15.3.4.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

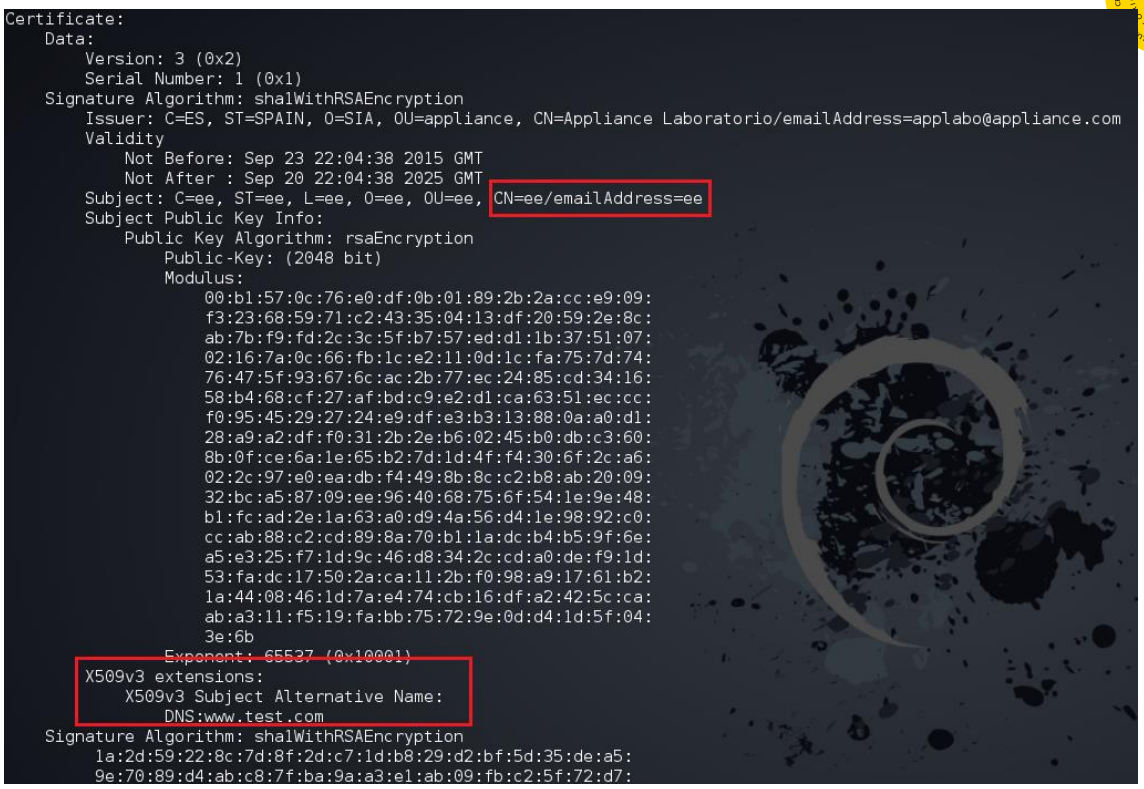
Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER".*
- On the *"File"* menu, click *"Add/Remove Snap-in".*
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add".*
- Press *"OK".*
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK".*
- Expand the Certificates section and select "Trusted Root Certification Authorities".

15.3.4.2. Procedure

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

1. Create a certified using the *"createCerts.sh"* script where the Common Name(CN) is *"ee"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).
2. Check that the Common Name is *"ee"* and that the Subject Alternative Name is *"www.test.com",* using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



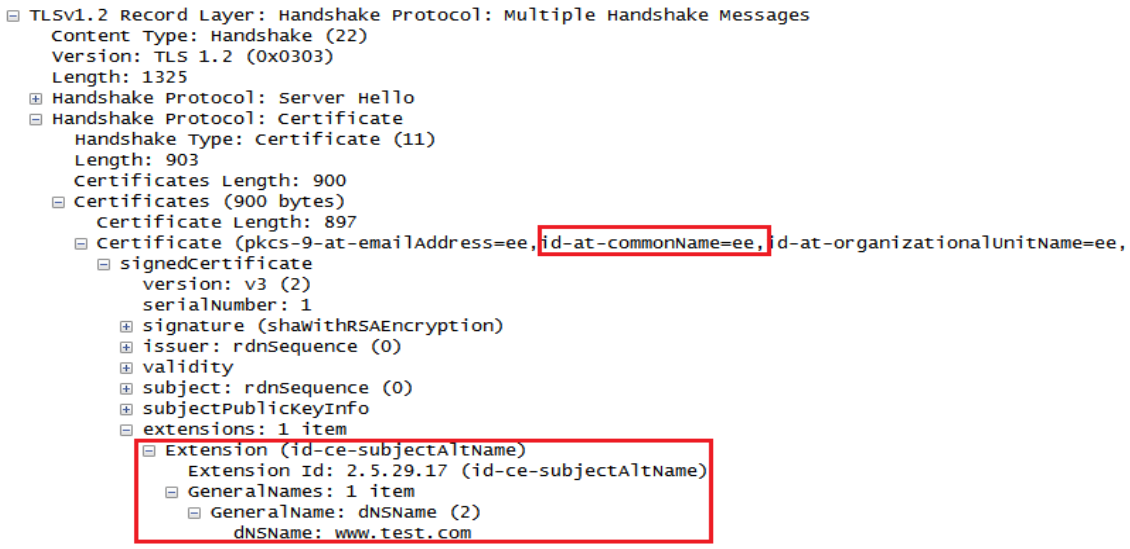
3. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - Click *"Add"*, "*type: https", "IP address: 192.168.1.101", "Host name: www.test.com"* and in the "SSL Certificate" load the certificate previously loaded.
4. Open a *"Wireshark"* application to verify that the handshake operation is performed correctly without user intervention.
5. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.4.3. Results

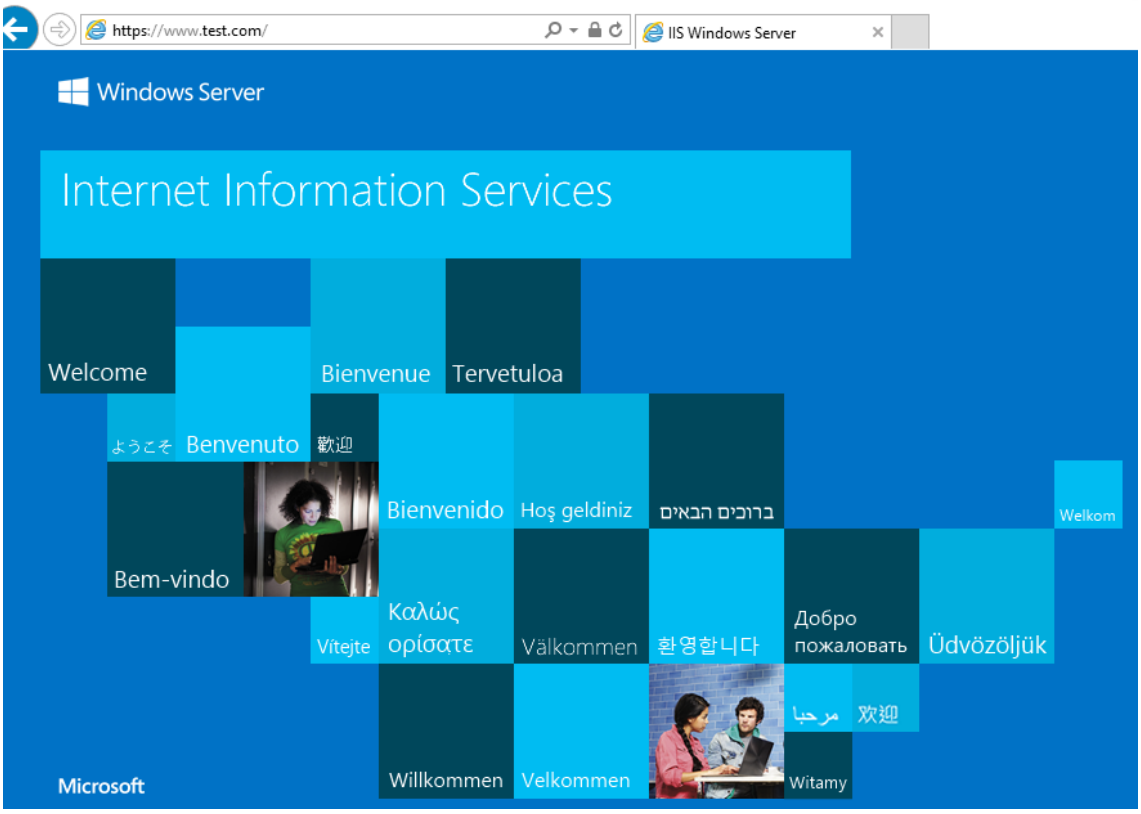
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name and the Subject Alternative Name in the *"Server Hello"* message



The following picture shows the browser response, where the connection is performed correctly.



15.3.4.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.5. Test 5.1

15.3.5.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

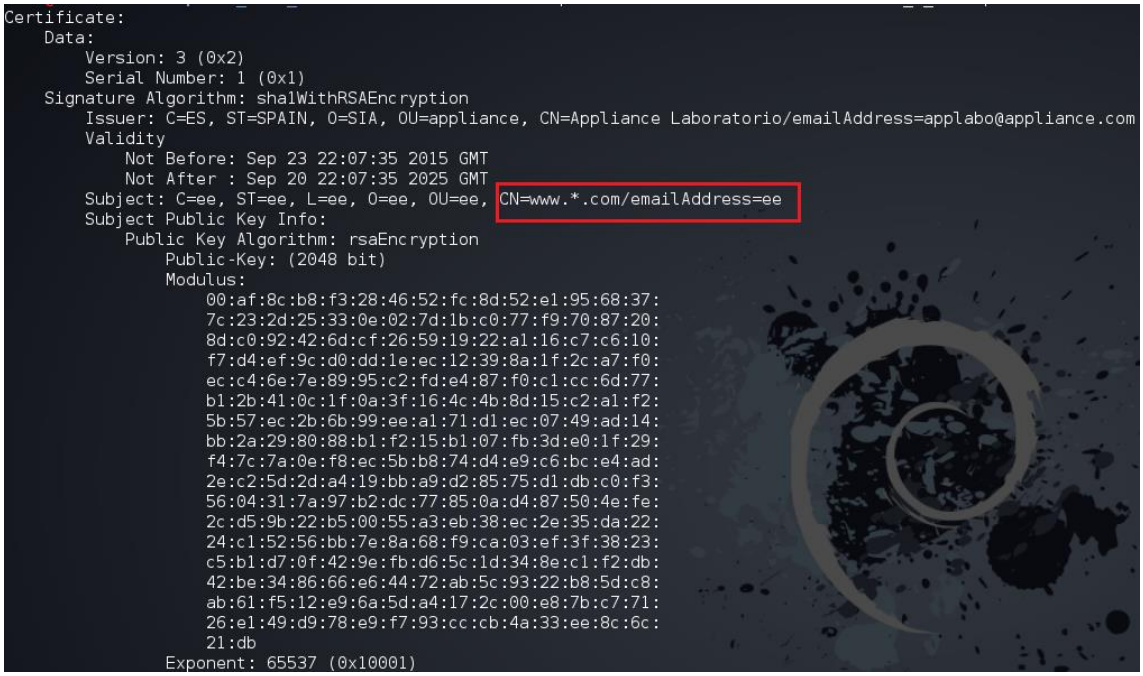
Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select "Trusted Root Certification Authorities".

15.3.5.2. Procedure

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

1. Create a certified using the *"createCerts.sh"* script where the Common Name(CN) is *"www.*.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).
2. Check that the Common Name is *"www.*.com",* using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



3. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.

- Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"*, *"Host name: www.test.com"* and in the *"SSL Certificate"* load the certificate previously loaded.

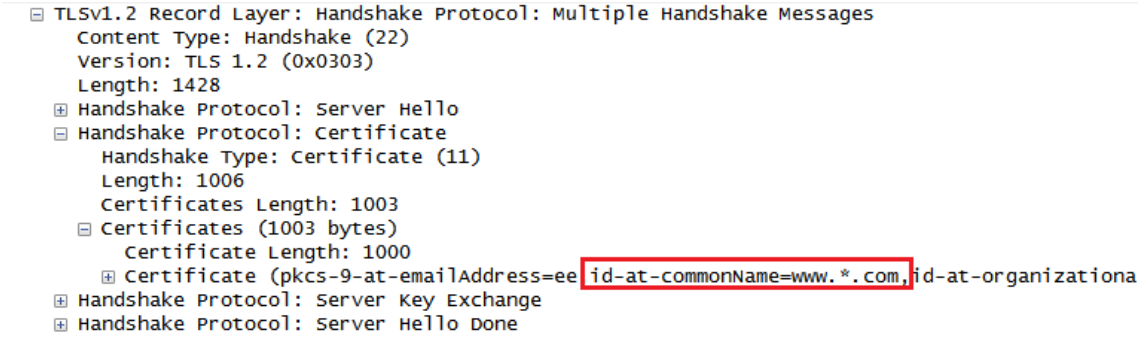
4. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
5. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.5.3. Results

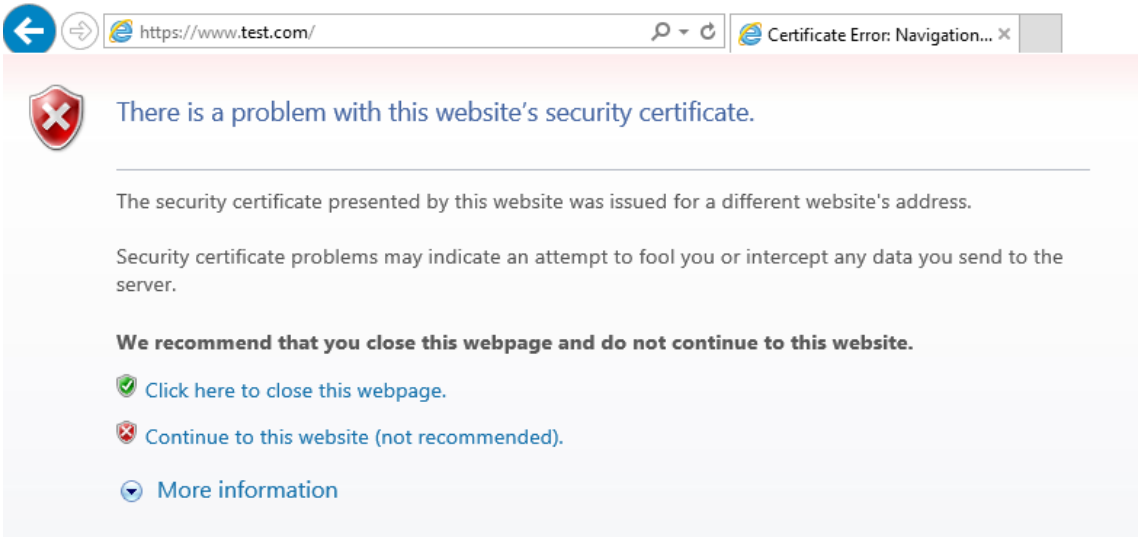
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name in the *"Server Hello"* message.



The following picture shows the browser response, where the connection is not performed correctly.



15.3.5.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.6. Test 5.2

15.3.6.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

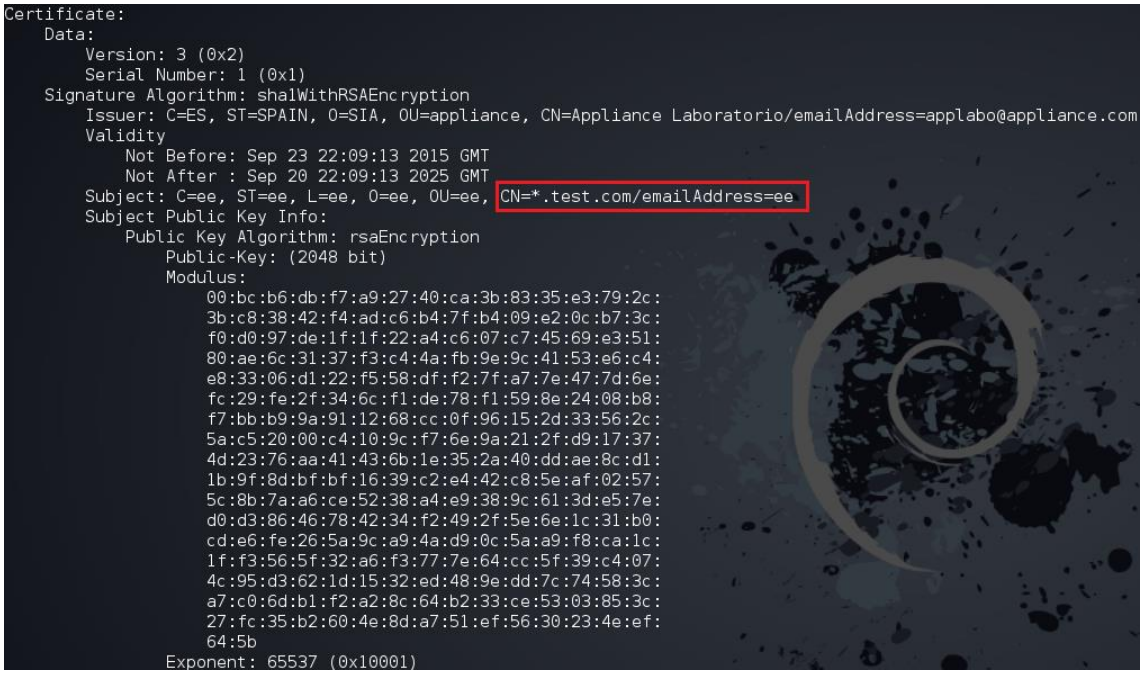
- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.

15.3.6.2. Procedure

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

1. Create a certified using the *"createCerts.sh"* script where the Common Name(CN) is *"*.test.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).

2. Check that the Common Name is *"*.test.com"*, using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



3. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.

   - Open *"Server Manager"* from the task bar.
   - Click *"IIS"*.
   - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
   - Double click in *"Server Certificates"*.
   - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
   - Click *"OK"*.
   - Expand *"Sites"* and click *"Default Web Site"*.
   - Click *"Bindings..."*.
   - Click *"Add"*, "*type: https"*, *"IP address: 192.168.1.101"*, *"Host name: www.test.com"* and in the *"SSL Certificate"* load the certificate previously loaded.

4. Open a *"Wireshark"* application to verify that the handshake operation is performed correctly without user intervention.

5. Open the browser and attempt to navigate to the test web (https://www.test.com).

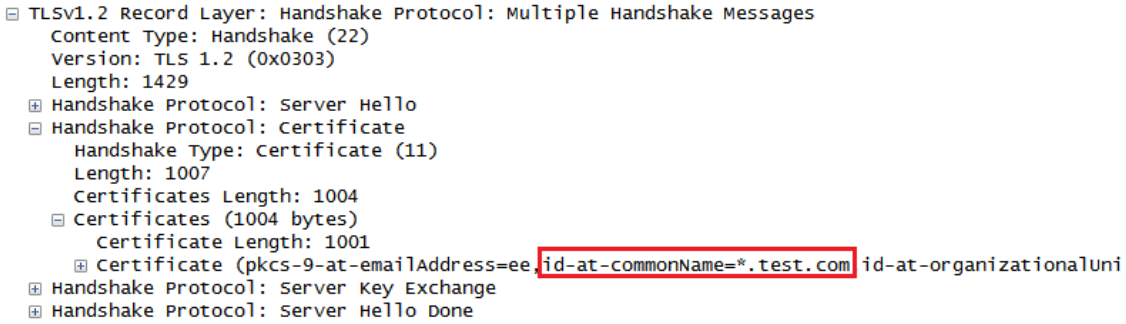
6. Modifies in the Server Machine the *"Host name"* of the web server from *"www.test.com"* to *"test.com".*
7. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
8. Open the browser and attempt to navigate to the test web (https://test.com).
9. Modifies in the Server Machine the *"Host name"* of the web server from *"test.com"* to *"bar.www.test.com".*
10. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
11. Open the browser and attempt to navigate to the test web (https://test.com).

15.3.6.3. Results

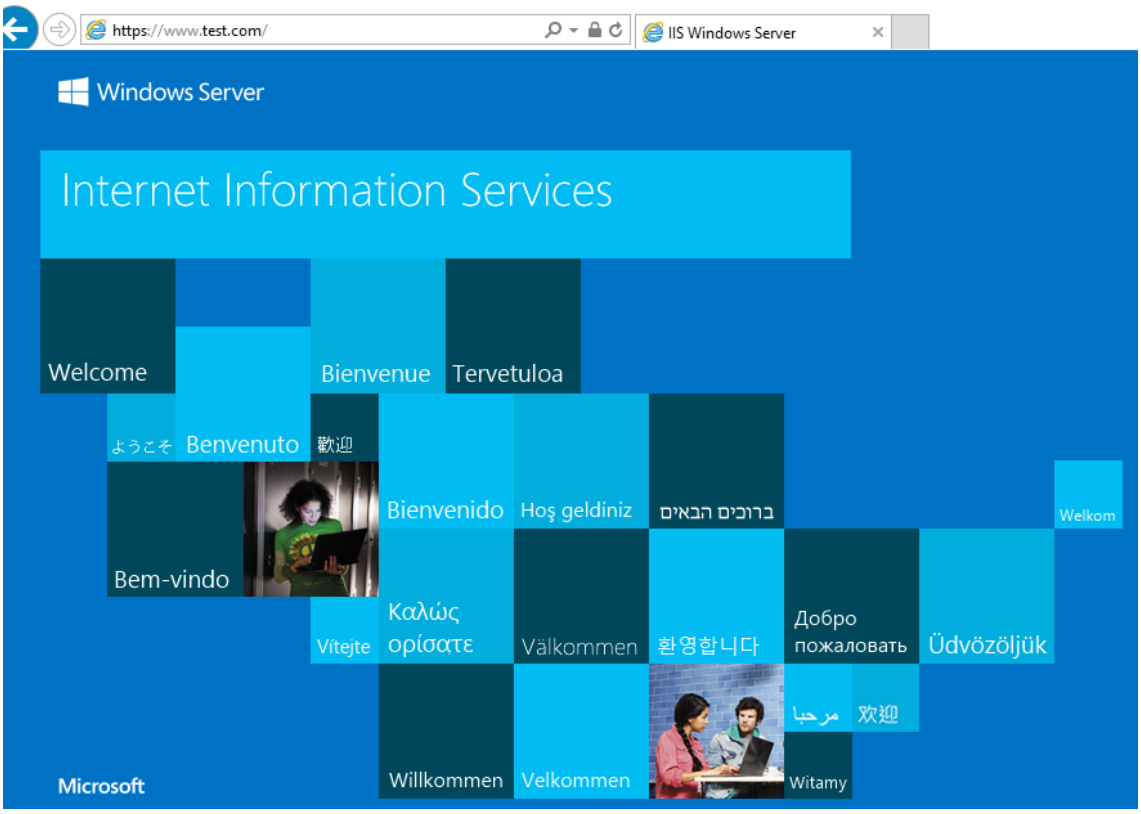
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

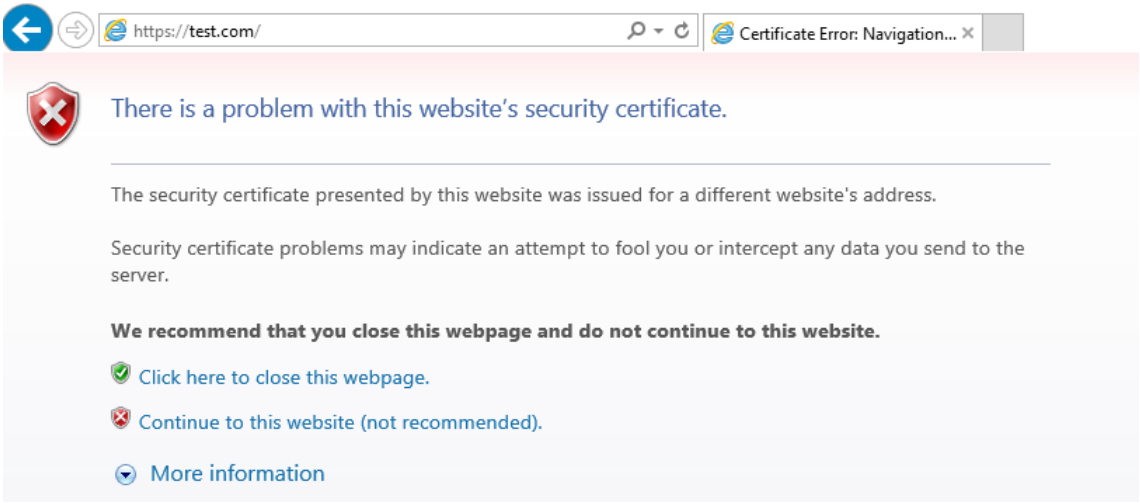
Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name in the *"Server Hello"* message.



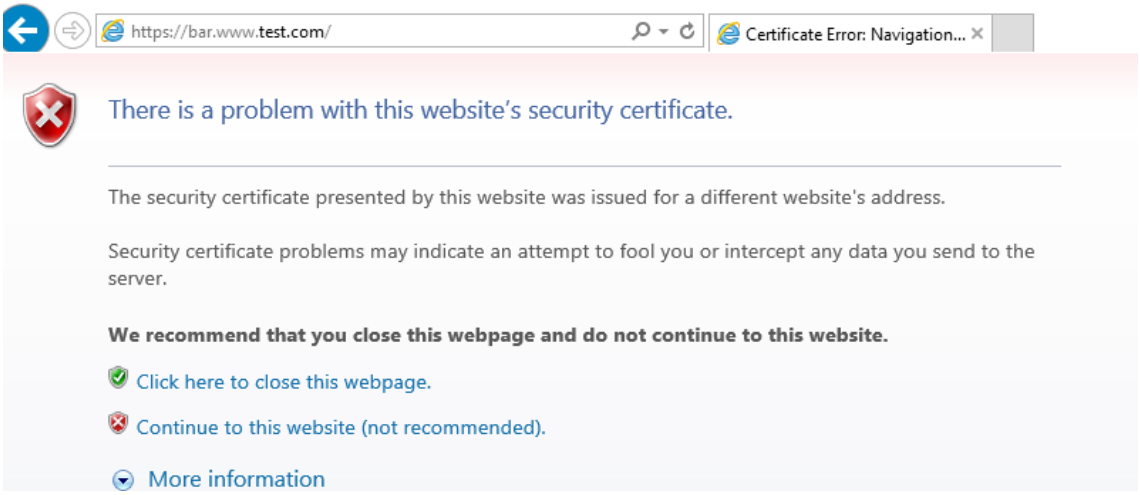
The following picture shows the browser response for the first part of the test, where the connection is performed correctly.



The following picture shows the browser response for the second part of the test, where the connection is not performed correctly.



The following picture shows the browser response for the final part of the test, where the connection is not performed correctly.



15.3.6.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.7. Test 5.3

15.3.7.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

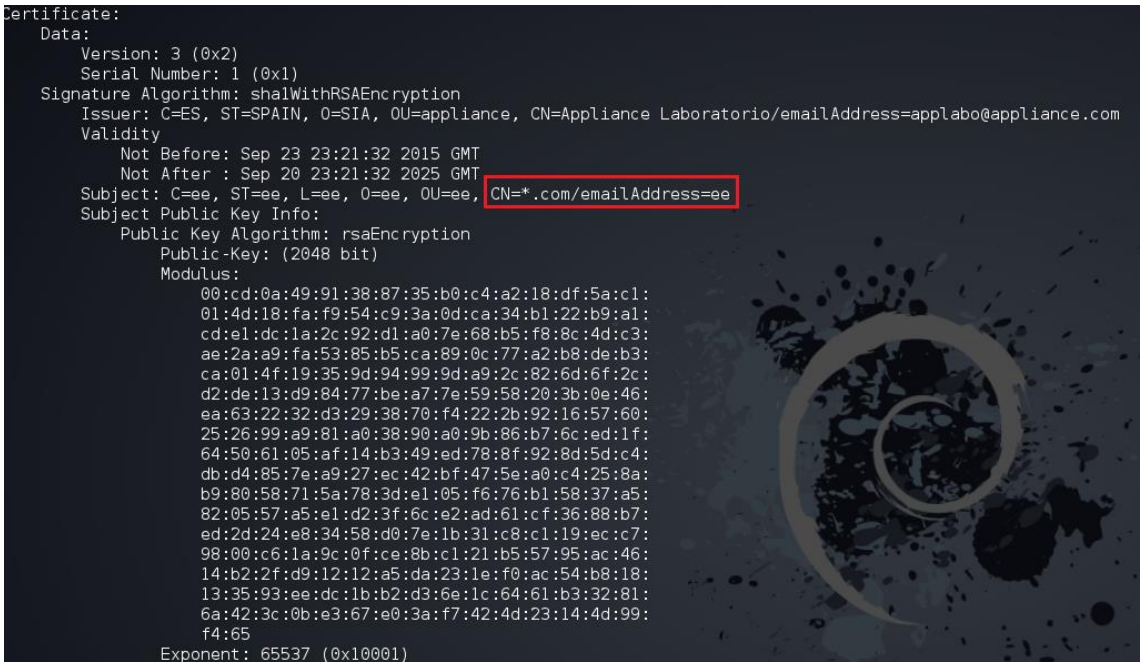
- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER".*
- On the *"File"* menu, click *"Add/Remove Snap-in".*
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add".*
- Press *"OK".*
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press "OK".
- Expand the Certificates section and select "Trusted Root Certification Authorities".

15.3.7.2. Procedure

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

1. Create a certified using the *"createCerts.sh"* script where the Common Name(CN) is *"*.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (CA).

2. Check that the Common Name is *"*.com",* using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



3. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.

   - Open *"Server Manager"* from the task bar.
   - Click *"IIS"*.
   - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
   - Double click in *"Server Certificates"*.
   - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
   - Click *"OK"*.
   - Expand *"Sites"* and click *"Default Web Site"*.
   - Click *"Bindings..."*.
   - Click *"Add"*, "*type: https", "IP address: 192.168.1.101"*, *"Host name: test.com"* and in the "SSL Certificate" load the certificate previously loaded.

4. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.

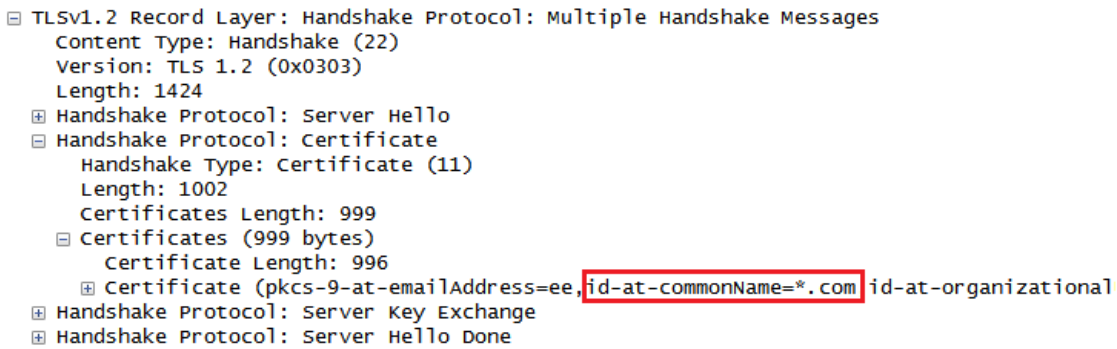
5. Open the browser and attempt to navigate to the test web (https://test.com).
6. Modifies in the Server Machine the *"Host name"* of the web server from *"test.com"* to *"www.test.com".*
7. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly without user intervention.
8. Open the browser and attempt to navigate to the test web (https://www.test.com).

15.3.7.3. Results

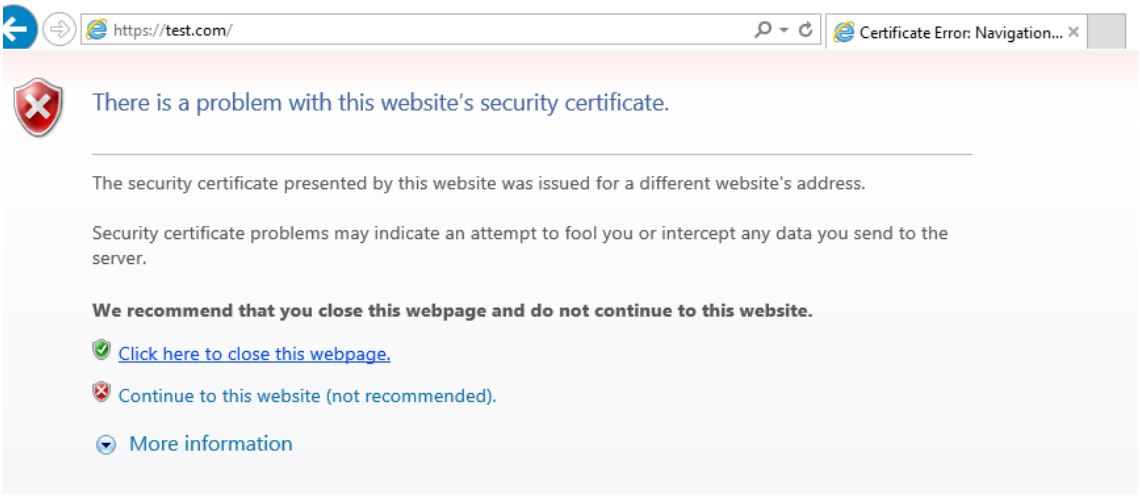
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

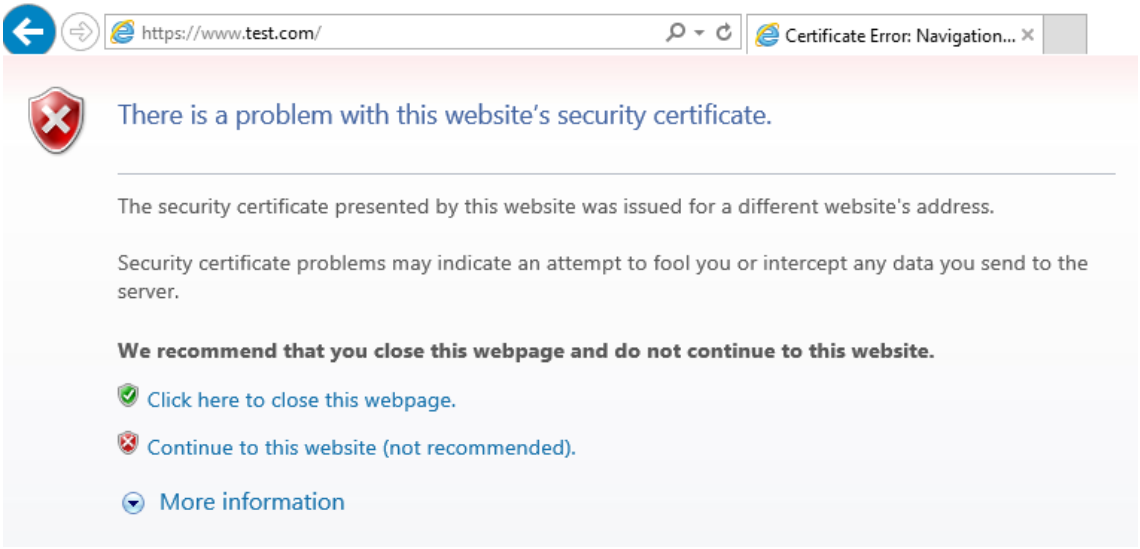
Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name in the *"Server Hello"* message.



The following picture shows the browser response for the first part of the test, where the connection is not performed correctly.



The following picture shows the browser response for the final part of the test, where the connection is not performed correctly.



15.3.7.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.3.8. Test 6

15.3.8.1. Setup

The following Certification Authority shall be used to perform the assurance activities listed in the Protection Profile.

- RootCA (ca)

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) service shall be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
- The current server is selected by default.
- Click next.
- From the *"Server Roles"* screen checks a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explaining all the features required to install the Domain Services.
- Click *"Add features"*. A new screen is shown and click next.
- On *"Select features"*, Click Next.
- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services.
- Click Next.
- Finally, click Install.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

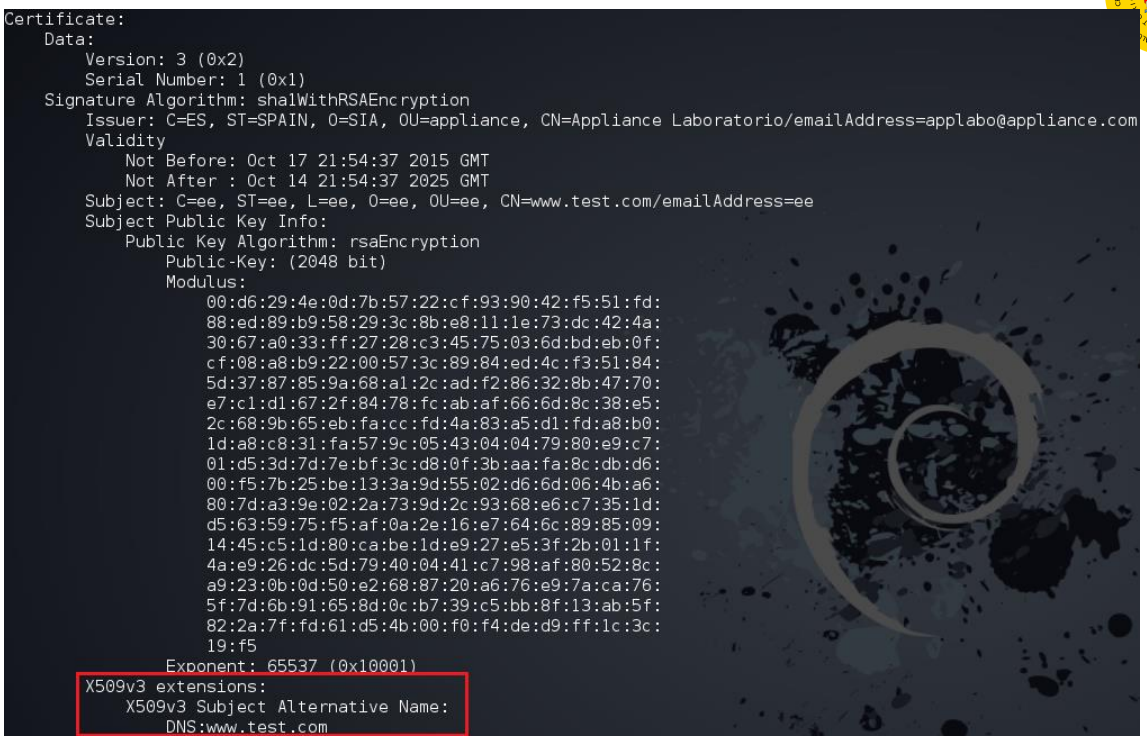
Add the *"RootCA.pkcs"* certificate in the Client and Server Machines following the next steps:

- Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
- At the command prompt, type *"mmc"* and press *"ENTER"*.
- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.

15.3.8.2. Procedure

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

1. Open a terminal in the Support Machine and type the following command:
   - *"gpedit /etc/ssl/openssl.cnf"*
2. In the *"openssl.cnf"* file, adds the following information:
   - [v3_custom]
   - subjectAltName = @alt_names
   - [alt_names]
   - DNS.1 = www.test.com
3. Create a certified using the *"createCerts.sh"* script in the Support Machine, where the Common Name(CN) is *"www.test.com"*. This script generates a certificate with RSA algorithm in *".pem" and ".pfx"* format. The new certificate is signed by a Certificate Authority (ca).
4. Check that the Subject Alternative Name contains the DNS *"www.test.com"*, using the *"openssl x509 -text -noout -in <certificate.pem>"* command, as it can be appreciated in the next picture.



5. Import the RSA certificate with ".pfx" format in the web server running into the Server Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.
    - *Click "Add", "type: https", "IP address: 192.168.1.101", "Host name: www.test.com" and in the "SSL Certificate" load the certificate previously loaded*.
6. Open a *"Wireshark"* application to verify that the handshake operation is performed correctly without user intervention.
7. Open the browser and attempt to navigate to the test web (https://www.test.com).

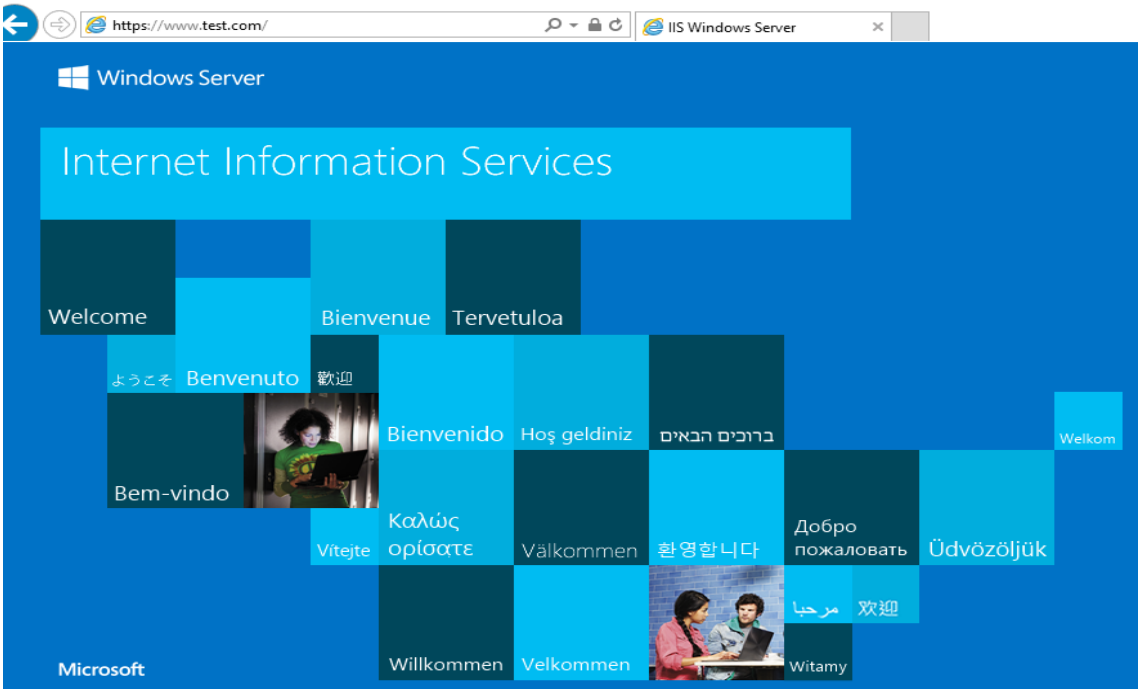
15.3.8.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Taking into account the evidences obtained from the *"Wireshark"* application shows the certificate sent by the server. In the next picture can be appreciate the Common Name and the Subject Alternative Name in the *"Server Hello"* message.

The following picture shows the browser response, indicating that the web address differs with the URL presented in the certificate.



15.3.8.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

15.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.1.2.

16. FCS_TLSC_EXT.1.3

16.1. Assurance activity

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.*

16.2. Documentation review activity

16.2.1. Findings

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

16.2.2. Verdict

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

16.3. Test Activity

16.3.1. Test 1

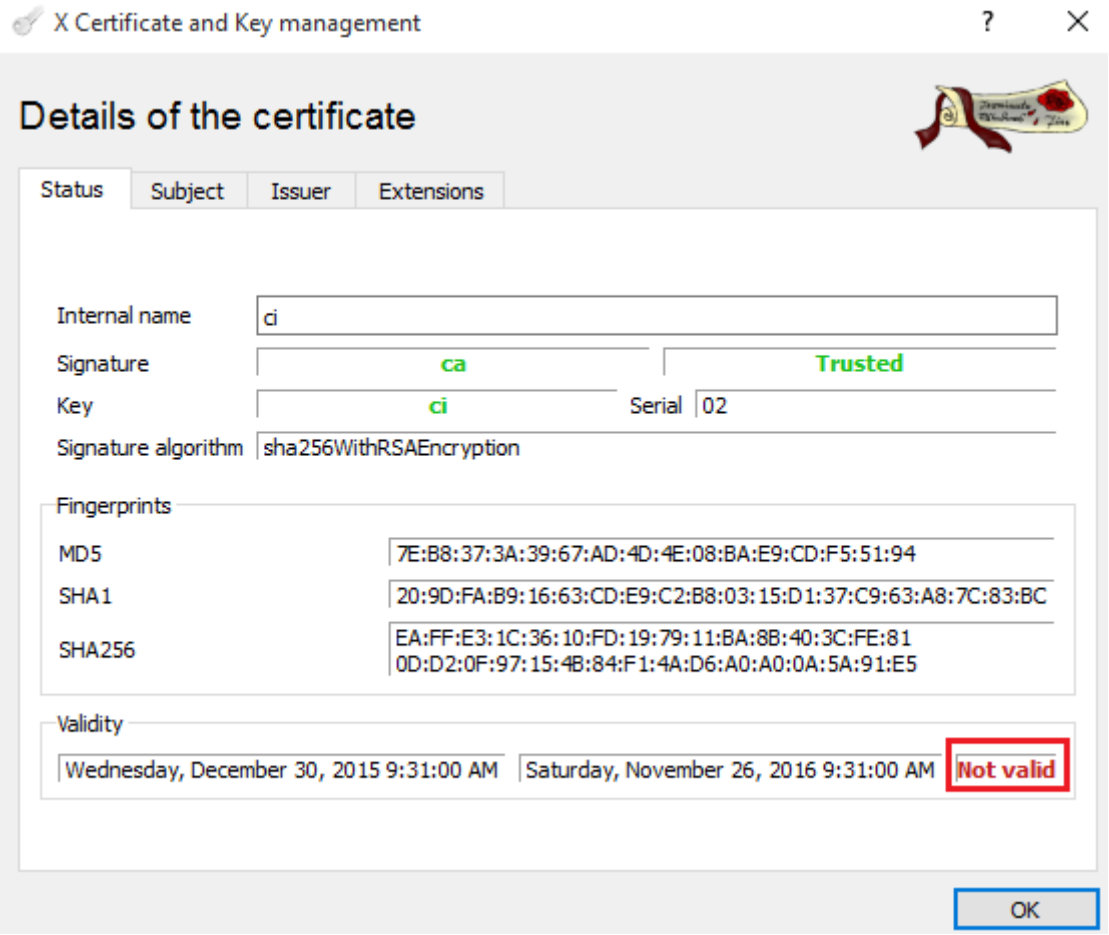
16.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- RootCA.
- IntermediateCA.
- Client with Common Name = *"www.test.com"*.

- Server with Common Name = *"www.test.com"*.
- File that contains the *"RootCA and intermediateCA"* in format *".pem"*(ca.pem).

The certificates listed above form an invalid certification path *"RootCA -> IntermediateCA -> (Server, Client)"*. Due to the certificate *"IntermediateCA"* is not valid, as it can be appreciated in the next picture.



The following certificates also are necessary to fulfill the assurance activity for the Test 1:

- RootCA.
- IntermediateCA.
- Client with Common Name = *"www.test.com"*.
- Server with Common Name = *"www.test.com"*
- File that contains the *"RootCA and intermediateCA"* in format *".pem"*(ca.pem).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA -> (Server, Client)"*.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

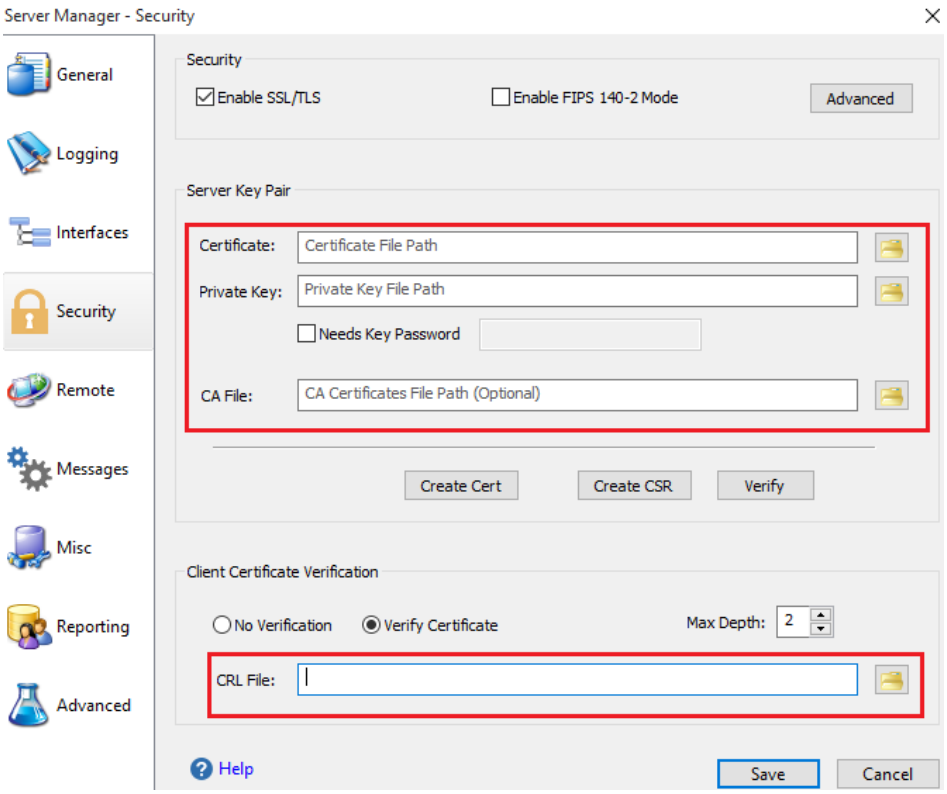
16.3.1.2. Procedure

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

1. Add the invalid certificates used for the test in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press "OK"
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pfx"* is stored.
    - Select "Personal".
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"client.pfx"* is stored.

2. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.

   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"server.crt"*, the *"server.pem" and "ca.crt"*. The *"ca.crt"* contains the *"RootCA and IntermediateCA"*.
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press save.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press save.



3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
4. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
5. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

6. Load the valid certificates following the steps listed in the points 1 and 2.
7. Open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
8. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).
9. In the client machine delete the *"IntermediateCA*" following the steps listed below:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK".*
    - At the command prompt, type *"mmc"* and press *"ENTER".*
    - On the *"File"* menu, click "*Add/Remove Snap-in".*
    - In the Add standalone Snap-in dialog box, select "Certificates".
    - Press *"Add".*
    - Press *"OK".*
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Intermediate Certification Authorities"*.
10. Right-click in *"IntermediateCA"* and select delete.
11. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
12. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

16.3.1.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

The packets captured with the *"Wireshark"* application shows the handshaking process, whereby can be appreciated that the client sends a *"Client hello"* message and the server response with a *"Server hello"* and *"Certificate Request"*. When the client receives the

"Certificate Request" this send an empty *"Certificate"* message due to the validation path is not valid.

After the client tries to authenticate, this send a *"Certificate"* message, as it can be appreciated in the following picture.

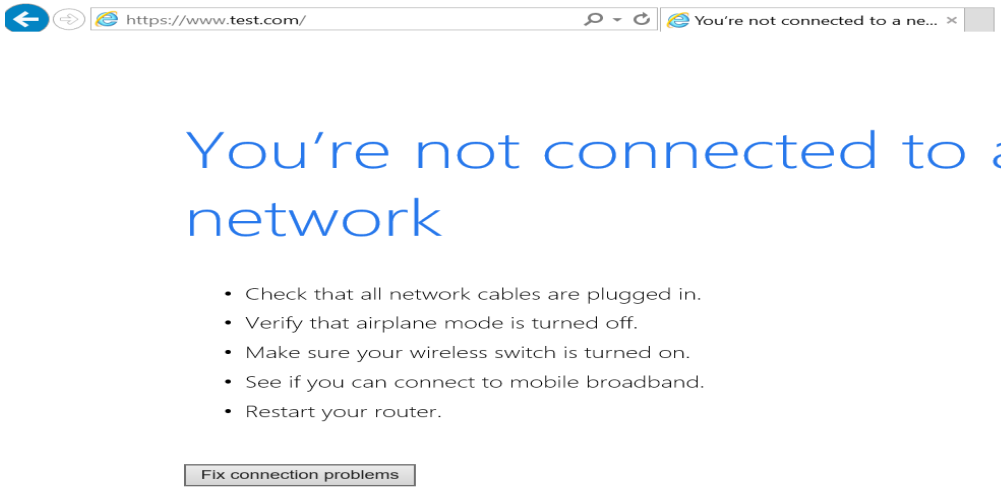
59	9.896937000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
60	9.899068000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
61	9.899069000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Certificate
62	9.899070000	192.168.1.102	192.168.1.120	TLSv1.2	483 Certificate Request, Server Hello Done
63	9.899135000	192.168.1.120	192.168.1.102	TCP	54 1791→443 [ACK] Seq=170 Ack=3350 Win=262144 Len=0
64	9.913801000	192.168.1.120	192.168.1.102	TCP	54 1791→443 [FIN, ACK] Seq=170 Ack=3350 Win=262144 Len=0
65	9.915043000	192.168.1.120	192.168.1.102	TCP	66 1792→443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
66	9.915266000	192.168.1.102	192.168.1.120	TCP	60 443→1791 [ACK] Seq=3350 Ack=171 Win=261968 Len=0
67	9.915268000	192.168.1.102	192.168.1.120	TCP	60 443→1791 [FIN, ACK] Seq=3350 Ack=171 Win=261968 Len=0
68	9.915306000	192.168.1.120	192.168.1.102	TCP	54 1791→443 [ACK] Seq=171 Ack=3351 Win=262144 Len=0
69	9.916198000	192.168.1.102	192.168.1.120	TCP	66 443→1792 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=8 SACK_PERM=1
70	9.916265000	192.168.1.120	192.168.1.102	TCP	54 1792→443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
71	9.916656000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
72	9.918744000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
73	9.918745000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Certificate
74	9.918746000	192.168.1.102	192.168.1.120	TLSv1.2	483 Certificate Request, Server Hello Done
75	9.918815000	192.168.1.120	192.168.1.102	TCP	54 1792→443 [ACK] Seq=170 Ack=3350 Win=262144 Len=0
76	9.928002000	192.168.1.120	192.168.1.102	TLSv1.2	403 Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
77	9.929559000	192.168.1.102	192.168.1.120	TLSv1.2	61 Alert (Level: Fatal, Description: Handshake Failure)

In the following picture it can be appreciated the content of the *"Certificate"* message send by the client.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 269
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 3
      Certificates Length: 0
  ⊞ Handshake Protocol: Client Key Exchange
⊞ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
⊞ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

Analyzing the content of this packet can see that the certificate length is zero, therefore any certificate is sent to the server. When the server receives a *"Certificate"* message without content, this send an *"Alert (Level: Fatal, Description: Handshake failure)"* message to the client.

The following picture shows the browser response.



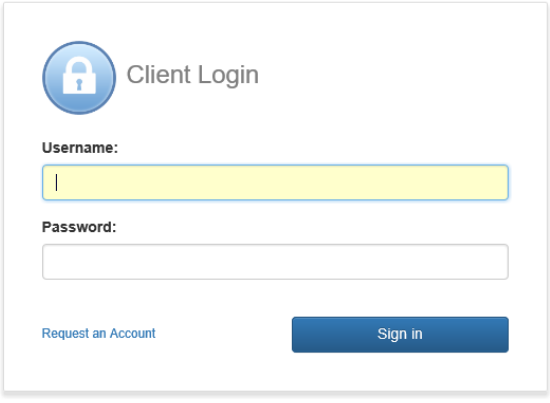
When a valid certification path is loaded in the Client Machine, the handshake is performed correctly, as it can be appreciated in the next *"Wireshark"* capture.

16 2.831733000	192.168.1.120	192.168.1.102	TCP	54	1796→443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
17 2.839790000	192.168.1.120	192.168.1.102	TLSv1.2	223	Client Hello
18 2.841875000	192.168.1.102	192.168.1.120	TLSv1.2	1514	Server Hello
19 2.841877000	192.168.1.102	192.168.1.120	TLSv1.2	1400	Certificate
20 2.841947000	192.168.1.120	192.168.1.102	TCP	54	1796→443 [ACK] Seq=170 Ack=2807 Win=262144 Len=0
21 2.864911000	192.168.1.120	192.168.1.102	TLSv1.2	2595	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Messag
22 2.866586000	192.168.1.102	192.168.1.120	TCP	60	443→1796 [ACK] Seq=2807 Ack=2711 Win=262144 Len=0
23 2.890978000	192.168.1.102	192.168.1.120	TLSv1.2	1216	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
24 2.891073000	192.168.1.120	192.168.1.102	TCP	54	1796→443 [ACK] Seq=2711 Ack=3969 Win=260864 Len=0
25 2.893101000	192.168.1.120	192.168.1.102	TLSv1.2	363	Application Data
26 2.894833000	192.168.1.102	192.168.1.120	TLSv1.2	443	Application Data

The *"Certificate"* message sent by the client is not empty. In addition, in this packet can be appreciate the common name of the *"Client"* and the *"IntermediateCA"* as demonstrate the following picture.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2461
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 1931
      Certificates Length: 1928
    ⊟ Certificates (1928 bytes)
        Certificate Length: 878
      ⊞ Certificate (pkcs-9-at-emailAddress=c1,id-at-commonName=www.test.com,id-at-organizationalUnitName=c1,id-at-organizationName=c1,id
        Certificate Length: 1044
      ⊞ Certificate (pkcs-9-at-emailAddress=ci,id-at-commonName=IntermediateCA,id-at-organizationalUnitName=ci,id-at-organizationName=ci,
```

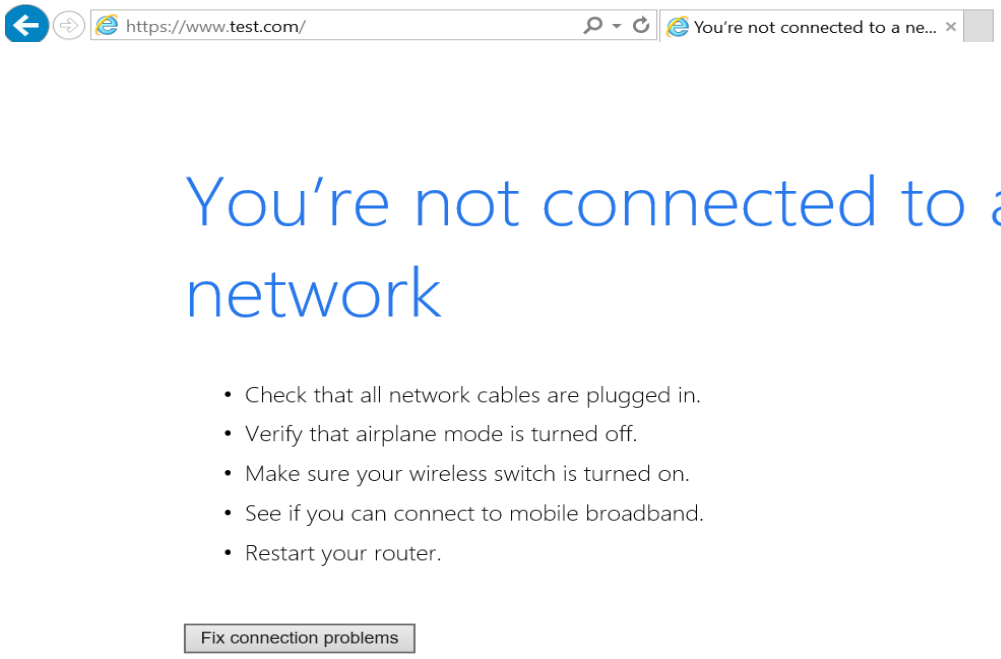
The following picture shows the browser response, this is a web page provided by the Cerberus FTP application.



The last part of the test 1 consist in eliminate the *"RootCA"* in the server machine. The traffic captured shows how the connection is not established correctly, as it can be appreciate in the next picture.

16	0.709298000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
17	0.711297000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
18	0.711299000	192.168.1.102	192.168.1.120.	TLSv1.2	756 Certificate
19	0.711358000	192.168.1.120	192.168.1.102	TCP	54 1894-443 [ACK] Seq=170 Ack=2163 Win=262144 Len=0
20	0.726995000	192.168.1.120	192.168.1.102	TLSv1.2	2595 Certificate, Client Key Exchange, Certificate Verify, Change
21	0.728401000	192.168.1.102	192.168.1.120	TCP	60 443-1894 [ACK] Seq=2163 Ack=2711 Win=262144 Len=0
22	0.728403000	192.168.1.102	192.168.1.120	TLSv1.2	61 Alert (Level: Fatal, Description: Unknown CA)

In addition, the browser shows the next screen.



16.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

16.3.2. Test 2

16.3.2.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- RootCA
- IntermediateCA
- Client revoked with Common Name = "www.test.com"

- Server with Common Name = "www.test.com"
- File that contains the "RootCA and intermediateCA" in format ".pem"(ca.pem).
- CRL file.

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA -> (Server, Client)"*.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

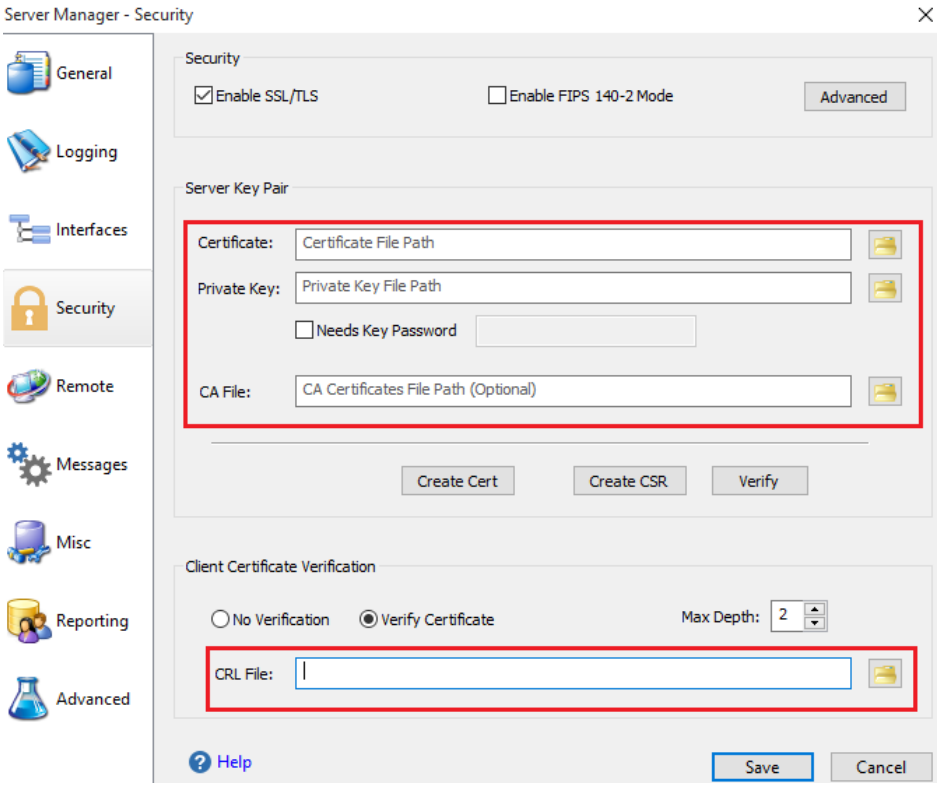
16.3.2.2. Procedure

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

1. Add the certificates used for the test in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.

- Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pfx"* is stored.
- Expand the Certificates section and select *"Intermediate Certification Authorities"*.
- Right-click on *"Intermediate Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"IntermediateCA.pfx"* is stored.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"clientRevoked.pfx"* is stored.

2. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"server.crt"*, the *"server.pem" and "ca.crt"*. The *"ca.crt"* contains the *"RootCA and IntermediateCA".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - In CRL File load the list with the revoked certificates.
   - Press save.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press save.



3. In the client machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
4. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
5. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

16.3.2.3. Results

The test have been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

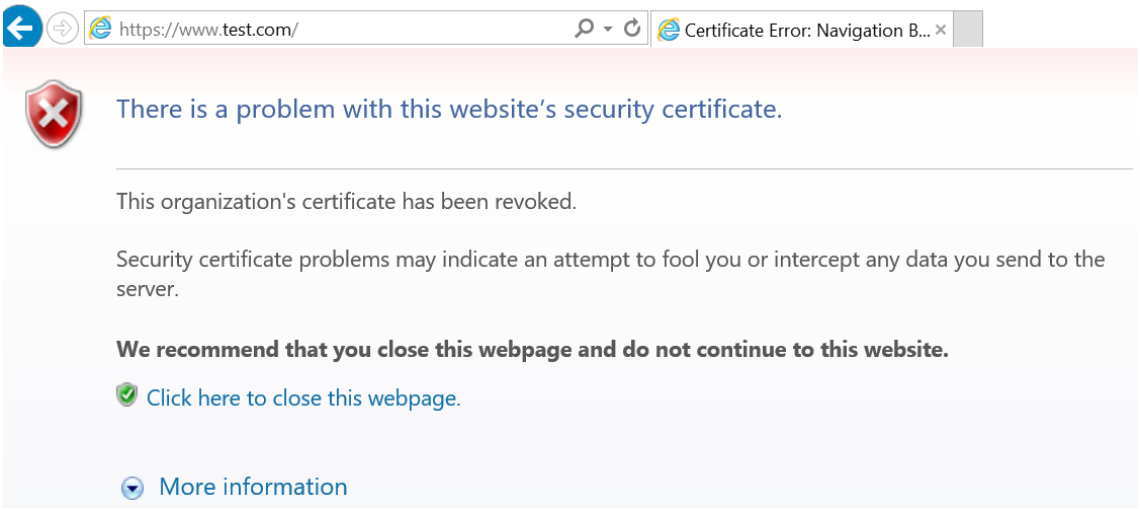
The packets captured with the *"Wireshark"* application shows the handshaking process, whereby it can be appreciated that the client sends a *"Certificate Request"*. The content of this packet is displayed in the following capture.

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2461
  Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 1931
      Certificates Length: 1928
    Certificates (1928 bytes)
        Certificate Length: 878
      Certificate (pkcs-9-at-emailAddress=r1,id-at-commonName=www.test.com,id-at-organizationalUnitName=r1,id-at-organizationName=r1
        Certificate Length: 1044
      Certificate (pkcs-9-at-emailAddress=ci,id-at-commonName=IntermediateCA,id-at-organizationalUnitName=ci,id-at-organizationName=c
  Handshake Protocol: Client Key Exchange
  Handshake Protocol: Certificate Verify
```

The client sends the certificated *"r1"* that corresponds with the certificated revoked in the CRL. When the server receives the revoked certificate, the server responses as show in the next picture.

```
TLSv1.2 Record Layer: Alert (Level: Fatal, Description: Certificate Revoked)
    Content Type: Alert (21)
    Version: TLS 1.2 (0x0303)
    Length: 2
  Alert Message
      Level: Fatal (2)
      Description: Certificate Revoked (44)
```

In addition, the browser responses as follows:



16.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

16.3.3. Test 3

16.3.3.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- RootCA.
- IntermediateCA.
- Client expired with Common Name = "www.test.com".
- Server with Common Name = "www.test.com".
- File that contains the "RootCA and intermediateCA" in format ".pem"(ca.pem).

The certificates listed above form a invalid certification path *"RootCA -> IntermediateCA -> (Server, Client)"*, because the *"Client"* certificate is expired.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

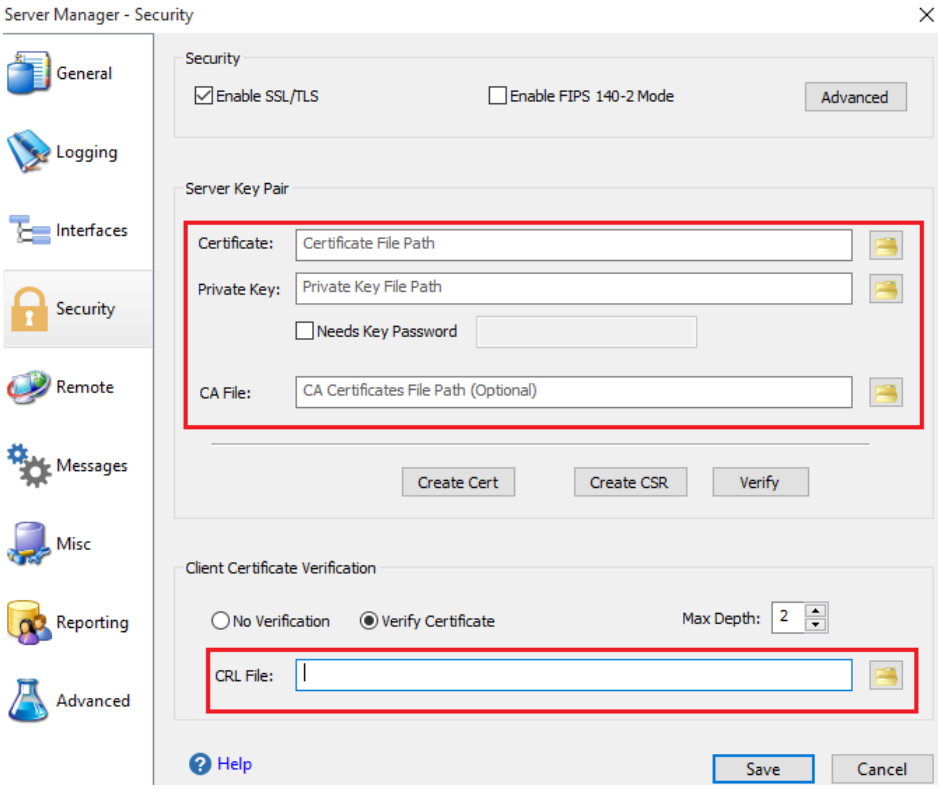
16.3.3.2. Procedure

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

1. Add the invalid certificate used for the test in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.

- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
- Right-click on "*Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pfx"* is stored.
- Expand the Certificates section and select *"Intermediate Certification Authorities"*.
- Right-click on *"Intermediate Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"IntermediateCA.pfx"* is stored.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"clientExpired.pfx"* is stored.

2. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"server.crt"*, the *"server.pem" and "ca.crt"*. The *"ca.crt"* contains the *"RootCA and IntermediateCA".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - In CRL File load the list with the revoked certificates.
   - Press save.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press save.



3. In the client machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
4. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
5. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

16.3.3.3. Results

The test have been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

The packets captured with the *"Wireshark"* application shows the handshaking process, whereby it can be appreciated that the client sends a *"Server hello"* message and the server response with a *"Server hello"* and *"Certificate Request"*. However, the client cannot send the

certificate due to the date is expired, this sends a *"Certificate"* message with a certificate parameter empty. In the following *"Wireshark"* captures can be appreciate this behavior.

20	31.673360000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
21	31.675114000	192.168.1.102	192.168.1.120	TLSv1.2	1161 Server Hello, Certificate, Certificate Request, Serv
22	31.675175000	192.168.1.120	192.168.1.102	TCP	54 1959→443 [ACK] Seq=170 Ack=1108 Win=260864 Len=0
23	31.699304000	192.168.1.120	192.168.1.102	TCP	54 1959→443 [FIN, ACK] Seq=170 Ack=1108 Win=260864 Len=
24	31.700307000	192.168.1.102	192.168.1.120	TCP	60 443→1959 [ACK] Seq=1108 Ack=171 Win=261968 Len=0
25	31.700317000	192.168.1.102	192.168.1.120	TCP	60 443→1959 [FIN, ACK] Seq=1108 Ack=171 Win=261968 Len=
26	31.700349000	192.168.1.120	192.168.1.102	TCP	54 1959→443 [ACK] Seq=171 Ack=1109 Win=260864 Len=0
27	31.700849000	192.168.1.120	192.168.1.102	TCP	66 1960→443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
28	31.701953000	192.168.1.102	192.168.1.120	TCP	66 443→1960 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=
29	31.702022000	192.168.1.120	192.168.1.102	TCP	54 1960→443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
30	31.702366000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
31	31.704361000	192.168.1.102	192.168.1.120	TLSv1.2	1161 Server Hello, Certificate, Certificate Request, Serv
32	31.704426000	192.168.1.120	192.168.1.102	TCP	54 1960→443 [ACK] Seq=170 Ack=1108 Win=260864 Len=0
33	31.717139000	192.168.1.120	192.168.1.102	TLSv1.2	403 Certificate, Client Key Exchange, Change Cipher Spec
34	31.718354000	192.168.1.102	192.168.1.120	TLSv1.2	61 Alert (Level: Fatal, Description: Handshake Failure)

The content of the *"Certificate"* message is shown in the next picture.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 269
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 3
      Certificates Length: 0
  ⊞ Handshake Protocol: Client Key Exchange
⊞ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
⊞ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

In addition, the browser response is shown in the next picture.

This page can't be displayed

Turn on TLS 1.0, TLS 1.1, and TLS 1.2 in Advanced settings and try connecting to **https://www.test.com** again. If this error persists, it is possible that this site uses an unsupported protocol. Please contact the site administrator.

Change settings

16.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

16.3.4. Test 4

16.3.4.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- RootCA.
- IntermediateCA.
- Client with Common Name = "www.test.com".
- Server with Common Name = "www.test.com".
- File that contains the "RootCA and intermediateCA" in format ".pem"(ca.pem).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA -> (Server, Client)"*.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

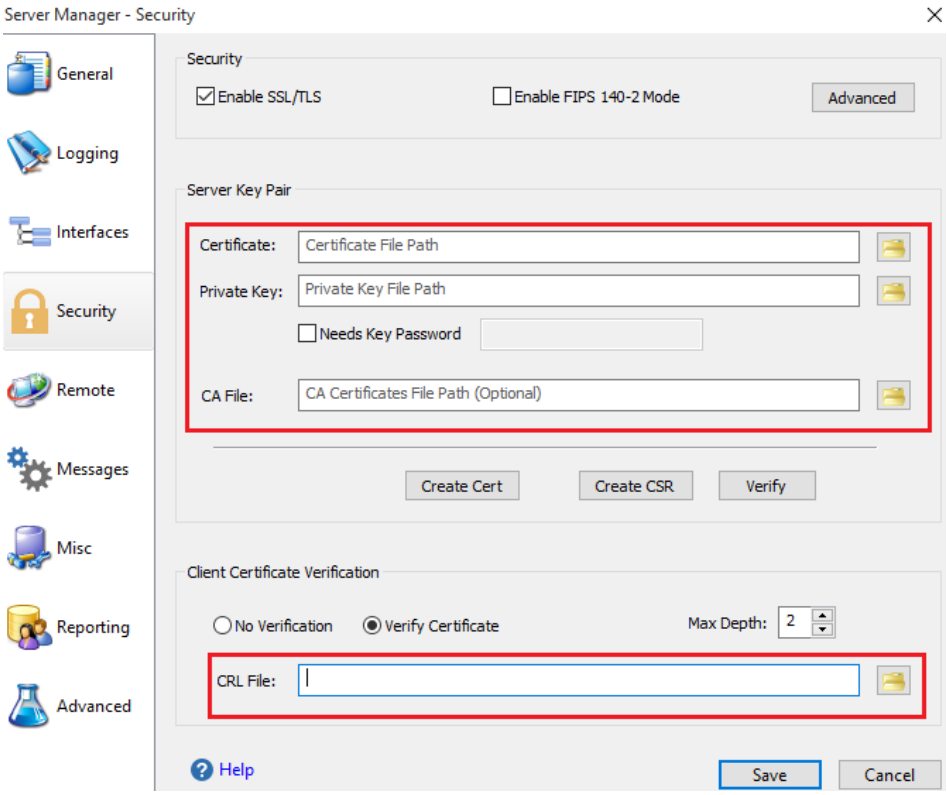
16.3.4.2. Procedure

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

1. Add the certificates used for the test in the Client Machine following the next steps:

- Click "Start", click "Run", type "mmc" and then click "OK".
- At the command prompt, type "mmc" and press "ENTER"
- On the "File" menu, click "Add/Remove Snap-in"
- In the Add standalone Snap-in dialog box, select "Certificates".
- Press "Add"
- Press "OK"
- In the Certificates Snap-in dialog box, select "My user account" and click next.
- Press "OK"
- Expand the Certificates section and select "Trusted Root Certification Authorities".
- Right-click on "Trusted Root Certification Authorities", select "All Tasks", then select import and browse to folder where the "RootCA.pfx" is stored.
- Expand the Certificates section and select "Intermediate Certification Authorities".
- Right-click on " Intermediate Certification Authorities ", select "All Tasks", then select import and browse to folder where the "IntermediateCA.pfx" is stored.
- Select "Personal".
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the "clientExpired.pfx" is stored.

2. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"server.crt"*, the *"server.pem" and "ca.crt"*. The *"ca.crt"* contains the *"RootCA and IntermediateCA".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - In CRL File load the list with the revoked certificates.
   - Press save.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press save.



3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
4. In the MITM Machine open a terminal and type the followings commands:
   - *"cd Desktop/SSL_Proxy"*
   - *"chmod 777 run_mitm"*
   - *"./run_mitm"*
5. In the MITM machine run the *"run_mitm"* script. This script configures the iptables file, activate the *"arp spoofing"* between the client machine and server machine and execute the *"ssl_proxy"* application. The *"ssl_proxy"* application is a tool used to modify the traffic. The following function modifies the *"CountryName"* into the *"issuer"* structure from *"ci"* to *"bi"*.
6. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
7. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

16.3.4.3. Results

The test have been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_client_certificate(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 5 #version + len

        if packet[indexPacket] == '\x0B':
            indexPacket += 54 #countryName
            newPacket = packet[:indexPacket] + '\x62' + packet[indexPacket + 1:]

    return newPacket
```

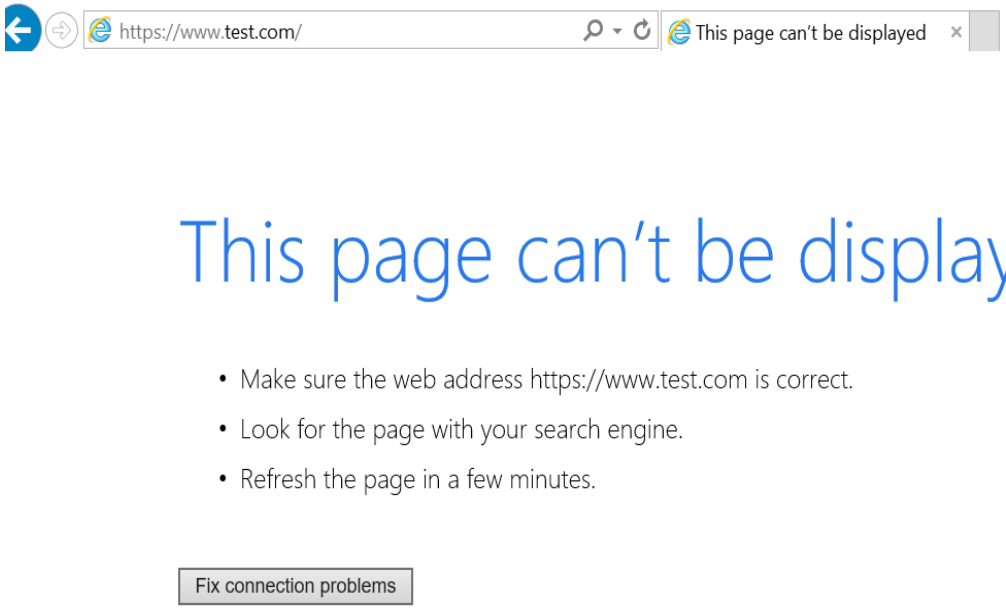
The packets captured with the *"Wireshark"* application shows the handshaking process. When the client sends the *"Certificate"* message, the MITM machine modifies the *"CountryName"* into the *"issuer"* structure from *"ci"* to *"bi"*. Therefore when the server receives the modified certificate, it sends an *"Alert message"* indicating that the *"CA"* is unknown as it can be appreciated in the following picture.

24	39.192658000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
25	39.193024000	192.168.1.102	192.168.1.120	TLSv1.2	1161 Server Hello, Certificate, Certificate Request, Server
26	39.194658000	192.168.1.120	192.168.1.102	TCP	60 1964→443 [ACK] Seq=170 Ack=1108 Win=260864 Len=0
27	39.238851000	192.168.1.120	192.168.1.102	TCP	60 1964→443 [FIN, ACK] Seq=170 Ack=1108 Win=260864 Len=0
28	39.238949000	192.168.1.102	192.168.1.120	TCP	54 443→1964 [ACK] Seq=1108 Ack=171 Win=261968 Len=0
29	39.239673000	192.168.1.102	192.168.1.120	TCP	54 443→1964 [FIN, ACK] Seq=1108 Ack=171 Win=261968 Len=0
30	39.239956000	192.168.1.120	192.168.1.102	TCP	60 1964→443 [ACK] Seq=171 Ack=1109 Win=260864 Len=0
32	40.287146000	192.168.1.120	192.168.1.102	TCP	74 52774→443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PE
33	40.287241000	192.168.1.102	192.168.1.120	TCP	74 443→52774 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1
34	40.288299000	192.168.1.120	192.168.1.102	TCP	66 52774→443 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=4946
35	40.305530000	192.168.1.120	192.168.1.102	TLSv1.2	235 Client Hello
36	40.305897000	192.168.1.102	192.168.1.120	TLSv1.2	1173 Server Hello, Certificate, Certificate Request, Server
37	40.306819000	192.168.1.120	192.168.1.102	TCP	66 52774→443 [ACK] Seq=170 Ack=1108 Win=32768 Len=0 TSval
38	40.332289000	192.168.1.120	192.168.1.102	TCP	1514 [TCP segment of a reassembled PDU]
39	40.332292000	192.168.1.120	192.168.1.102	TCP	78 [TCP segment of a reassembled PDU]
40	40.332411000	192.168.1.102	192.168.1.120	TCP	66 443→52774 [ACK] Seq=1108 Ack=1630 Win=262144 Len=0 TSv
41	40.333390000	192.168.1.120	192.168.1.102	TLSv1.2	1147 Certificate, Client Key Exchange, Certificate Verify
42	40.334424000	192.168.1.102	192.168.1.120	TLSv1.2	73 Alert (Level: Fatal, Description: Unknown CA)

Checking the Certificate message in the Server Machine can be appreciate the modification performed by the MITM machine. The next picture shows in detail the content of the *"Certificate"* message.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2461
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 1931
      Certificates Length: 1928
    ⊟ Certificates (1928 bytes)
        Certificate Length: 878
      ⊟ Certificate (pkcs-9-at-emailAddress=c1,id-at-commonName=www.test.com,id-
        ⊟ signedCertificate
            version: v3 (2)
            serialNumber: 9
          ⊞ signature (sha256WithRSAEncryption)
          ⊟ issuer: rdnSequence (0)
            ⊟ rdnSequence: 7 items (pkcs-9-at-emailAddress=ci,id-at-commonName=
              ⊞ RDNSequence item: 1 item (id-at-countryName=bi)
              ⊞ RDNSequence item: 1 item (id-at-stateOrProvinceName=ci)
              ⊞ RDNSequence item: 1 item (id-at-localityName=ci)
              ⊞ RDNSequence item: 1 item (id-at-organizationName=ci)
              ⊞ RDNSequence item: 1 item (id-at-organizationalUnitName=ci)
              ⊞ RDNSequence item: 1 item (id-at-commonName=IntermediateCA)
              ⊞ RDNSequence item: 1 item (pkcs-9-at-emailAddress=ci)
```

In addition, the browser response is shown in the next picture.



16.3.4.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

16.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.1.3.

17. FCS_TLSC_EXT.2.1

17.1. Assurance activity

The evaluator will verify that 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 will verify that AGD guidance includes configuration of the supported Elliptic Curves Extension.

The evaluator will also perform the following test:

The evaluator will configure the server to perform an ECDHE key exchange message in the TLS connection using a nonsupported ECDHE curve (for example, P192)and shall verify that the OS disconnects after receiving the server's Key Exchange handshake message.

17.2. Documentation review activity

17.2.1. Findings

The section 6.2.2 of the *"Windows 10 Security Target"* document, states that the Windows includes automatically the elliptic curve extension as part of the "Client Hello" message. The elliptic curves references in this section are: *"secp256r1"*, *"secp384r1"* and *"secp521r1"*. In addition, this section indicates that the *"secp521r1"* curve is disabled by default.

The section 4.1 of the "Windows 10 and Server 2012 R2 GP OS Operational Guidance" document, describes how the elliptic curve extension shall be configured in the Windows Server operating system.

For example, for the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256. To enable this cipher suite with an elliptic curve, e.g. secp256r1, you use the SSL cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256. The difference is the final four characters which indicate the elliptic curve that is to be used where P256 equals secp256r1.

The section 6.2.2 of the *"Windows 10 Security Target"* document specifies that for Windows 10, the priority of the curve can be modified editing the "ECC Curve Order", the process to carry out this modification is detailed in the section 4.1 of the "Windows 10 and Server 2012 R2 GP OS Operational Guidance" document.

In addition, the Operational Guidance states that the "secp521r1" curve is not enable by default in Windows 10 nor Windows Server 2012 R2.

17.2.2. Verdict

The evaluator has checked that the TSS describes the elliptic curves supported by the TOE and the curve enabled by default. In addition, the TSS specifies that the elliptic curves can be modified editing the "ECC Curve Order" file.

The AGD documentation describes how configure the "ECC Curve Order" file to enable the "secp521r1" curve or how change the priority of the curves.

Therefore, 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.

17.3. Test Activity

17.3.1. Test 1

17.3.1.1. Setup

The following certificate shall be used to perform the assurance activities listed in the Protection Profile. The certificate listed below has been created using the *"createCertificateECDSA.sh"* script. This script generates a certificate using elliptic curve. For the test purpose, the evaluator must generate a certificate with the following curve: *"secp192r1"*. The certificates are in *".pem"* and *".pfx"* format.

- certificate (secp192r1)

The scripts listed above, generate a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

In the Server Machine shall be installed the *"Cerberus FTP Server enterprise"* and *"Wireshark"* applications.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificateECDSA.sh"* script shall be copied in the Support Machine.

17.3.1.2. Procedure

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

13. Load the EC certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"certificate.pem"* and the *"certificateKey.pem"*.
    - Press save.
14. Modify the cipher suit in the Client Machine following the steps listed down below.
    - Left click on *"Start"*, press *"search"* and type *"Run"*.
    - In the Run application type *"gpedit.msc"* and press *"OK"* button.
    - Expand "*Administrative Templates -> Network"* in *"Local Group Policy Editor"* and open the ECC Curve Order file.
    - Click on the radio button *"Enable"*. In the text box ECC Curve Order, erases all curves and write *"sepc192r1"*.
    - Press *"Apply"* and reboot the machine.
15. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
16. In Client Machine opens the browser and attempt to navigate to the test web (https://192.168.1.102).

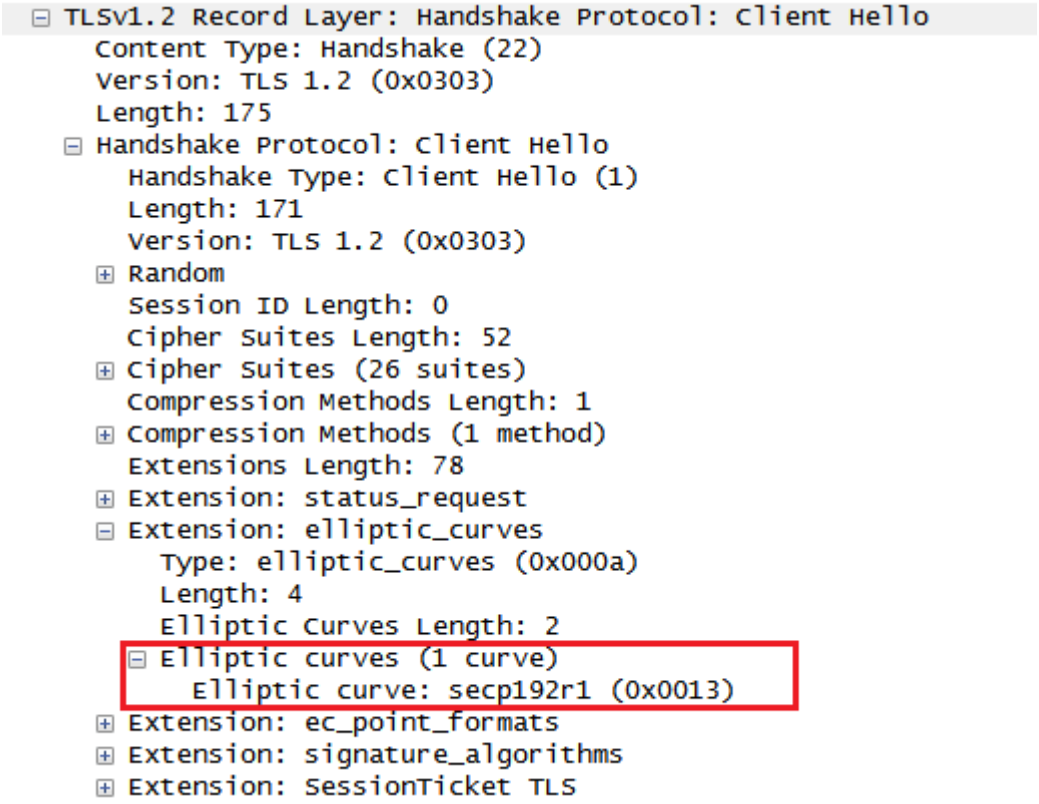
17.3.1.3. Results

The test has been performed in the following platforms:

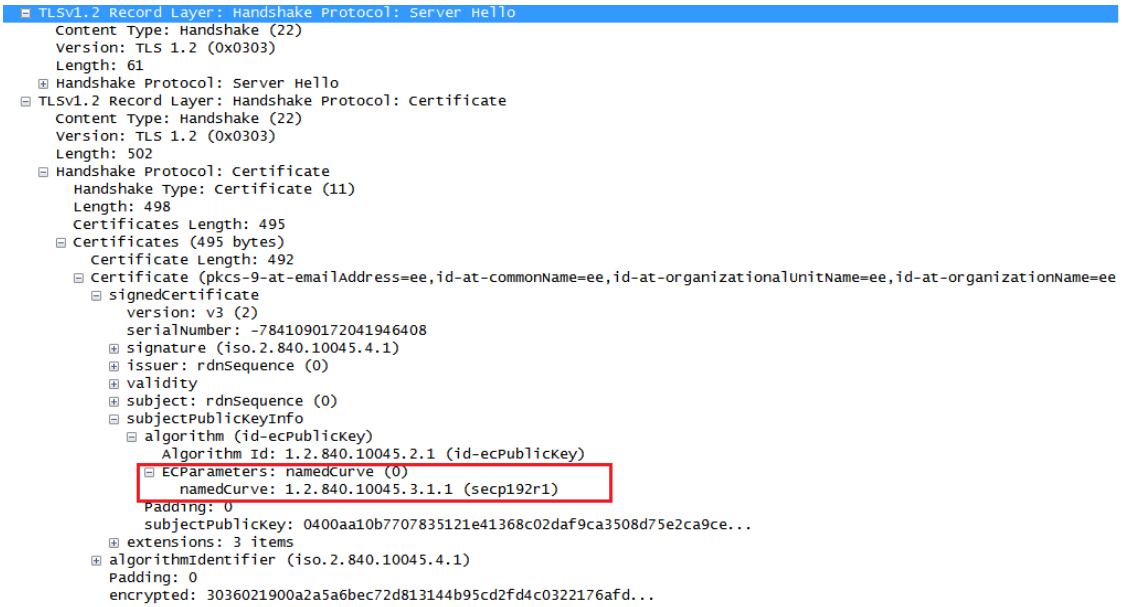
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.

- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

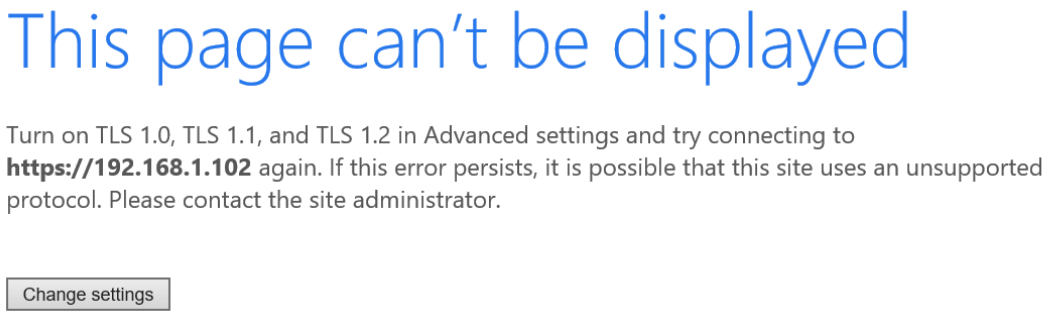
Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that, the handshake is not performed when the curve select in the cipher suite is *"secp192r1"*. The following picture shows the curve in the *"Client hello"* message.



The *"Server hello"* message uses the curve selected by the client as it can be appreciated in the next picture.



The client does not continue the handshake after receives the *"Server hello"* with the curve *"secp192r1"*. The browser response is shown in the following capture.



17.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

17.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.2.1.

18. FCS_TLSC_EXT.3

18.1. Assurance activity

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 Hash Algorithm enumeration within the signature_algorithms extension (for example, send a certificate with a SHA1 signature). The evaluator will verify that the OS disconnects after receiving the server's Certificate handshake message.

18.2. Documentation review activity

18.2.1. Findings

The section 6.2.2 of the *"Windows 10 Security Target"* document describes the signature algorithm used by default in the *"Client Hello"* message. This algorithms are:

- SHA1
- SHA256
- SHA384
- SHA512

Additionally, this section explains that the algorithm extension can be configured editing a registry key.

The section 4.1 of the *"Windows 10 and Windows Server 2012 R2 GP OS Operational Guidance"* document explains how modify the signature algorithm extension. In order to modify the signature algorithm the following registry key shall be editing:

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00010003"

In addition, the section 4.1 states that *"Remove the signature algorithm that should not be used. No additional algorithms other than the default set may be specified"*. Therefore, when the evaluated configuration is applied the SHA1 algorithm must be removed.

18.2.2. 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.

18.3. Test Activity

18.3.1. Test 1

18.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"createCertificate.sh"* script.

- certificate (RSA) with SHA-1.

The script listed above generates a certificate in *"pem"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Platforms listed in the ST)
- Support Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.101
- Client Machine, IP = 192.168.1.102
- Support Machine, IP = 192.168.1.109

The Web Server (IIS) must be installed and configured following the next steps:

- Open the Server Manager from the task bar.
- From Server Manager Dashboard select Add roles and Features.
- Select "Role-based or features-based" installation from the "Installation Type" screen, and click next.
- The current server is selected by default. Click next.
- From the "Server Roles" screen check a mark in the box "Web Server (IIS)". An additional pop-up screen must appear explain all the features required to install the Domain Services. Click "Add features". A new screen is shown and click next.
- On "Select features", Click Next.

- Review the information on the Web Server Role (IIS) tab and click Next.
- On Select Role Services. Click Next.
- Finally, click Install.

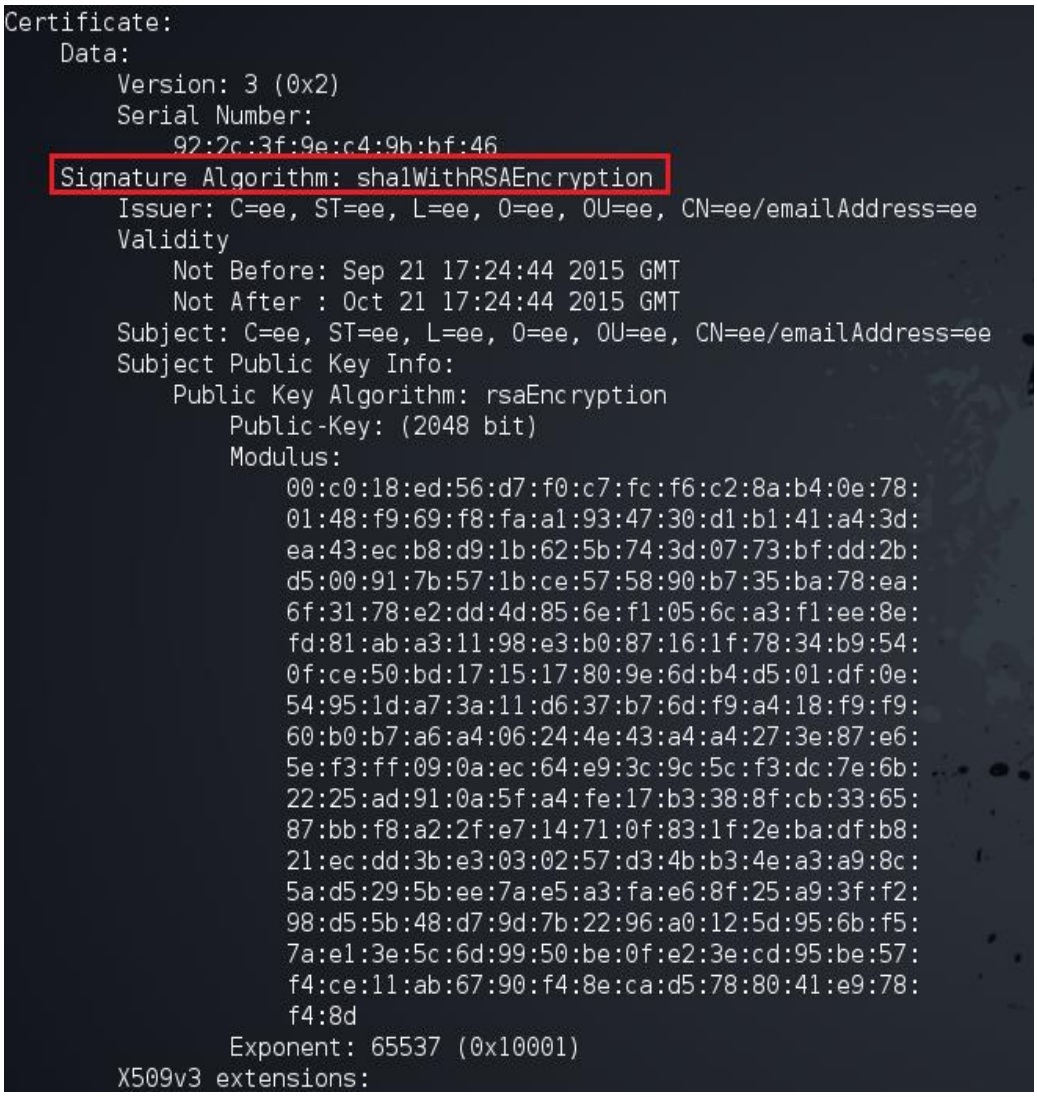
The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"createCertificate.sh"* script shall be copied in the Support Machine.

18.3.1.2. Procedure

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

10. Open a terminal in the Server Machine and checks that the certificate.pem contains the SHA-1 signature using *"openssl x509 -text -noout -in certificate.pem"* command, as it can be appreciated in the next picture.



11. Import the *"RSA"* certificate in *".pfx"* format created in the Client Machine.
    - Open *"Server Manager"* from the task bar.
    - Click *"IIS"*.
    - Right-click in the server name where the web service is installed and click *"Internet Information Services (IIS) Manager"*.
    - Double click in *"Server Certificates"*.
    - Click *"Import..."*, browse to folder where the certificate is stored and type the password (*ee*).
    - Click *"OK"*.
    - Expand *"Sites"* and click *"Default Web Site"*.
    - Click *"Bindings..."*.

12. Click *"Add"*, *"type: https"*, *"IP address: 192.168.1.101"* and in the "SSL Certificate" load the certificate previously loaded.
13. Open a *"Wireshark"* application in the Client Machine to verify that the handshake operation is not performed correctly.
14. Open the browser and attempt to navigate to the test web (https://192.168.1.101).

18.3.1.3. Results

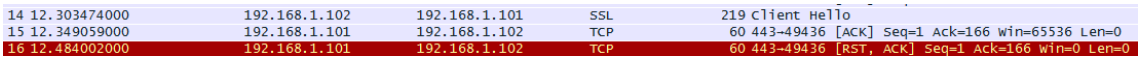
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

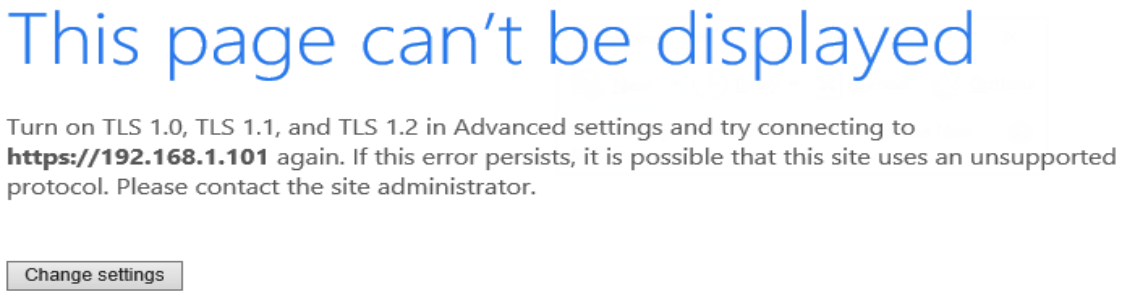
Taking into account the evidences obtained from the *"Wireshark"* application, the evaluator can determine that, the connection is not established correctly using a certificate with SHA-1 signature algorithm. The hashes contained in the extension *"signature_algorithms"* in the *"Client Hello"* message can be appreciate in the following picture.

```
⊟ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 156
    Version: TLS 1.2 (0x0303)
  ⊞ Random
    Session ID Length: 0
    Cipher Suites Length: 20
  ⊞ Cipher Suites (10 suites)
    Compression Methods Length: 1
  ⊞ Compression Methods (1 method)
    Extensions Length: 95
  ⊞ Extension: server_name
  ⊞ Extension: status_request
  ⊞ Extension: elliptic_curves
  ⊞ Extension: ec_point_formats
  ⊟ Extension: signature_algorithms
      Type: signature_algorithms (0x000d)
      Length: 14
      Signature Hash Algorithms Length: 12
    ⊟ Signature Hash Algorithms (6 algorithms)
      ⊟ Signature Hash Algorithm: 0x0401
            Signature Hash Algorithm Hash: SHA256 (4)
            Signature Hash Algorithm Signature: RSA (1)
      ⊟ Signature Hash Algorithm: 0x0501
            Signature Hash Algorithm Hash: SHA384 (5)
            Signature Hash Algorithm Signature: RSA (1)
      ⊟ Signature Hash Algorithm: 0x0403
            Signature Hash Algorithm Hash: SHA256 (4)
            Signature Hash Algorithm Signature: ECDSA (3)
      ⊟ Signature Hash Algorithm: 0x0503
            Signature Hash Algorithm Hash: SHA384 (5)
            Signature Hash Algorithm Signature: ECDSA (3)
      ⊟ Signature Hash Algorithm: 0x0601
            Signature Hash Algorithm Hash: SHA512 (6)
            Signature Hash Algorithm Signature: RSA (1)
      ⊟ Signature Hash Algorithm: 0x0603
            Signature Hash Algorithm Hash: SHA512 (6)
            Signature Hash Algorithm Signature: ECDSA (3)
  ⊞ Extension: SessionTicket TLS
```

When the server receives the "Client Hello" packet and checks the "signature algorithms" extension, this sends a "RST, ACK" packet, as it can be appreciated in the following picture.



In addition, the browser shows to the user the following screen when the Client Machine receives the reset packet.



18.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

18.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.3.1.

19. FCS_TLSC_EXT.4

19.1. Assurance activity

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:

Configure the server to require mutual authentication and then modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field must not be the CA used to sign the client's certificate. The evaluator will verify the connection is unsuccessful.

19.2. Documentation review activity

19.2.1. Findings

The section 6.2.2 of the *"Windows 10 Security Target"* document states that:

"Windows 10 implements TLS to enable a trusted network path that is used for client and server authentication, as well as HTTPS"

On the other hand, the section 7.2 of the *"Windows 10 and Windows Server 2012 R2 GP OS Operational Guidance"* document includes the following URL:

- https://technet.microsoft.com/en-us/library/cc754246.aspx

In this URL can be find the information about how the client can obtain and configure the client certificate.

19.2.2. 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.

19.3. Test Activity

19.3.1. Test 1

19.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1(i1)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine shall be installed the *"Cerberus FTP Server enterprise"* and *"Wireshark"* applications.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

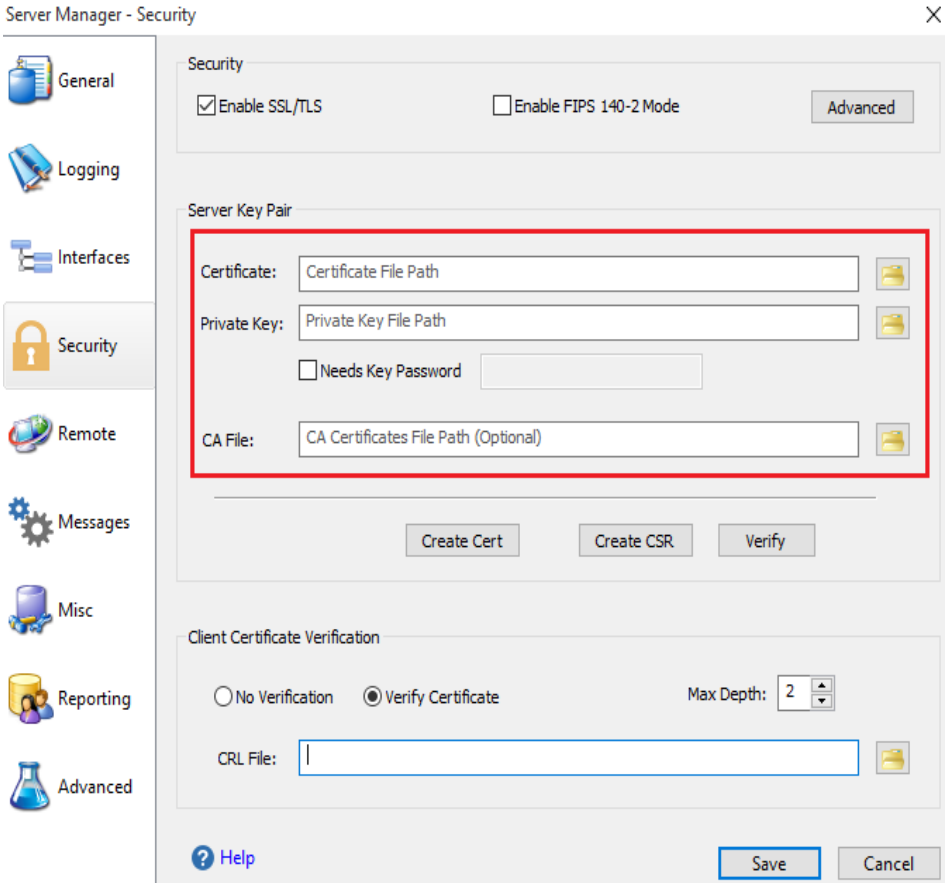
The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine

19.3.1.2. Procedure

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

13. Add the invalid certificates used for the test in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER".*
    - On the *"File"* menu, click *"Add/Remove Snap-in".*
    - In the Add standalone Snap-in dialog box, select *"Certificates".*
    - Press *"Add".*
    - Press *"OK".*
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK".*
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pfx"* is stored.
    - Select *"Personal".*
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"client.pfx"* is stored.
14. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"server.crt"*, the *"server.pem" and "chain.pem".* The *"chain.pem"* contains the *"RootCA"* and *"IntermediateCA1".*
    - In Client Certificate Verification select "*Verify Certificate*".
    - Press save.
    - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
    - Press save.



15. In the MITM Machine open a terminal and type the followings commands:
    a. *"cd Desktop/SSL_Proxy"*
    b. *"chmod 777 run_mitm"*
    c. *"./run_mitm"*
16. In the client machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
17. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
18. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

19.3.1.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.

- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_CA_request(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 58
        if packet[indexPacket] == '\x16':
            indexPacket += 1924
            packetLen = len(packet)
            if packetLen > 2000:
                if packet[indexPacket] == '\x16':
                    indexPacket += 5
                    if packet[indexPacket] == '\x0D':
                        indexPacket += 58
                        newPacket = packet[:indexPacket] + '\x30' + packet[indexPacket + 1:] #CountryName modified


    return newPacket
```

The above function changes the contryName from *"ca"* to *"c0"* in the *"RootCA"* certificate of the *"Certificate Request"* message.

The packets captured with the *"Wireshark"* application shows the handshaking process, whereby it can be appreciated that the client sends a *"Client Hello"* message and the server response with a *"Server Hello"* and *"Certificate Request"*.

When the client receives the *"Certificate Request"* modified, this continues the handshake process sending to the server the *"Certificate, Client Key Exchange, Change Cipher Spec and Encrypted Handshake"* messages. The modified packet can be appreciated in the next picture:

Finally when the server receives *"Certificate, Client Key Exchange, Change Cipher Spec and Encrypted Handshake"* messages, this send an *"Alert Description: Handshake Failure"* message. The following picture shows the browser response. The next picture shows the *"Alert"* message.

32	12.653820000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
33	12.655911000	192.168.1.102	192.168.1.120	TCP	60 443→2714 [ACK] Seq=1 Ack=170 Win=30720 Len=0
34	12.692887000	192.168.1.102	192.168.1.120	TCP	60 443→2713 [ACK] Seq=2139 Ack=171 Win=30720 Len=0
35	12.771834000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
36	12.771836000	192.168.1.102	192.168.1.120	TLSv1.2	732 Certificate
37	12.771945000	192.168.1.120	192.168.1.102	TCP	54 2714→443 [ACK] Seq=170 Ack=2139 Win=262144 Len=0
38	12.793722000	192.168.1.120	192.168.1.102	TLSv1.2	403 Certificate, Client Key Exchange, Change Cipher Spec,
39	12.797695000	192.168.1.102	192.168.1.120	TCP	60 443→2714 [ACK] Seq=2139 Ack=519 Win=31744 Len=0
40	12.798693000	192.168.1.102	192.168.1.120	TLSv1.2	61 Alert (Level: Fatal, Description: Handshake Failure)

This page can't be displayed

Turn on TLS 1.0, TLS 1.1, and TLS 1.2 in Advanced settings and try connecting to **https://www.test.com** again. If this error persists, it is possible that this site uses an unsupported protocol. Please contact the site administrator.

Change settings

19.3.1.4. Verdict

The evaluator considers that, the tests results obtained during the test activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the PASS verdict is assigned to the test activity.

19.4. Final Verdict

Due to all test activities 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 this FCS_TLSC_EXT.4.1.

20. FDP_ACF_EXT.1.1

20.1. Assurance activity

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.

20.2. Documentation review activity

20.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.3.1 Discretionary Access Control,** which is related to the access control policy and how this policies are applied to the 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 used algorithm to determine whether a user has access to one file or not. Information about which kind of permissions are checked in each steps is also included.

The fourth one includes information about which are the defaults access rules and how the inheritance and these rules are applied by default to all new object.

Finally, the following subsections are also included in the 6.3.1 Discretionary Access Control.

- 6.3.1.5 DAC Management
- 6.3.1.6 Reference Mediation

20.2.2. Verdict

The evaluator has reviewed the information provided in section 6.3.1 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 the folder and files in the operating system and the access control policy applied in any situation.

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

20.3. Test Activity

20.3.1. Test 1 to Test 6

20.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- User accounts with user names *user1* and *user2* shall exist. These users shall belong to the default *Users* group. Passwords for these accounts must be *p@ss1234*.
- Both user groups, *Users* and *Administrators*, shall have the same permissions (read, write and execute) over the folder where the scripts are stored.
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

20.3.1.2. Procedure

In order to perform these tests, the evaluator has followed the steps defined below:

1. Run as administrator, the script *FDP_ACF_EXT.1.ps1*. The script behavior is explained as follows:
    - FDP_ACF_EXT.1.ps1: It creates a new empty file in the *user1* home directory and after that it invokes the following scripts in order to perform each test. The script source code is as follows:

```
. .\common\UserAndGroupManagement.ps1

#Variables
$groupName = "Users"
$user1= "user1"
$user2 = "user2"
$pass = "p@ss1234"

$filePath = "Documents\file.txt"
$usersFolder = "$env:HOMEDRIVE\Users"
$user1HomePath = "$usersFolder\$user1"
$user2HomePath = "$usersFolder\$user2"
$fullFilePathUser1 = "$user1HomePath\$filePath"
$fullFilePathUser2 = "$user2HomePath\$filePath"

#Creating a new file in user1 HOME directory
Write-Host "Creating file in user1 home directory..."
$securepass = $pass | ConvertTo-SecureString -AsPlainText -Force
$user1Cred = New-Object System.Management.Automation.PSCredential("$env:COMPUTERNAME\$user1",$securepass)
$process = Start-Process powershell -ArgumentList "-Command New-Item $fullFilePathUser1 -type file; pause" -Credential $user1Cred -Wait
Write-Host "Done!`n"

#Test 1-3
Write-Host "Running Test 1 to 3 (view the new opened window)..."
$user2Cred = New-Object System.Management.Automation.PSCredential("$env:COMPUTERNAME\$user2",$securepass)
$process = Start-Process powershell -ArgumentList "-File Tests1-3.ps1 -path $fullFilePathUser1 " -Credential $user2Cred -Wait
Write-Host "Done!`n"

#Test 4-6
Write-Host "Running Test 4 to 6 (view the new opened window)..."
$process = Start-Process powershell -ArgumentList "-File Tests4-6.ps1 -pathUser1 $fullFilePathUser1 -pathUser2 $fullFilePathUser2" -Credential $user1Cred -Wait
Write-Host "Done!`n"
```

- Test1-3.ps1: It carries out the tests 1 to 3. All of these operations are performed using the *user2* credentials. The script source code is as follows:

```
param([string]$path)

#Test 1 - Read file
Write-Host "`n****TEST 1 ****"
Write-Host "`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 "`n****TEST 2 ****"
Write-Host "`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 "`n****TEST 3 ****"
Write-Host "`User2 attempts to delete a file in user1's home directory..."
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 *user1* credentials. The script source code is as follows:

```
param([string]$pathUser1, [string]$pathUser2)

#Test 4 - Create file
Write-Host "`n****TEST 4 ****"
Write-Host "`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 "`n****TEST 5 ****"
Write-Host "`User1 attempts to modify a file in user1's home directory..."
try{
    Add-Content $pathUser1 "Modified" -ErrorAction Stop
    Write-Host "Done!"
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}

#Test 6 - Delete a correct file
Write-Host "`n****TEST 6 ****"
Write-Host "`User1 attempts to delete a file in user1's home directory..."
try{
    Remove-Item $pathUser1 -ErrorAction Stop
    Write-Host "Done!"
} catch [Exception]{
    Write-Warning $Error[0].Exception.Message
}
pause
```

2. To execute the script, type the following command in a PowerShell terminal: ".*FDP_ACF_EXT1.ps1*".

3. Observe the results.

20.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same results for all the tested platforms. The following screenshots show the results obtained during the test execution:

```
****TEST 1 ****
User2 attempts to read a file in user1's home directory...
WARNING: Access is denied

****TEST 2 ****
User2 attempts to modify a file in user1's home directory...
WARNING: Access is denied

****TEST 3 ****
User2 attempts to delete a file in user1's home directory...
WARNING: Access is denied
Press Enter to continue...:
```

```
****TEST 4 ****
User1 attempts to create a new file in user2's home directory...
WARNING: Access to the path 'C:\Users\user2\Documents\file.txt' is denied.

****TEST 5 ****
User1 attempts to modify a file in user1's home directory...
Done!

****TEST 6 ****
User1 attempts to delete a file in user1's home directory...
Done!
Press Enter to continue...:
```

As it can be observed, user2 does not have access to files stored in the user1's home directory, and therefore user2 cannot read, modify or delete them. On the other hand, the user1 does not have permission to create a file in user2's home directory, but has permissions to modify or delete files stored in its home directory.

These behaviors match with the described one in the assurance activity section.

20.3.1.4. Verdict

The evaluator considers that, the tests results obtained during the test activity demonstrate the fulfillment of all test requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the test activity.

20.4. Final Verdict

Due to both documentation review activity and test activities 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 FDP_ACF_EXT.1.1.

21. FDP_IFC_EXT.1

21.1. Assurance activity

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).

21.2. Documentation review activity

21.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.3.2 VPN Client.** This section states that *"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"*

The TSS does not state conformance with any native VPN client, it only provides interfaces for application in order to establish a secure tunnel. Moreover the TSS has made a pointer to the MSDM describing all these available interfaces and further information about VPN configuration.

21.2.2. 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.

On the other hand, once the VPN channel is established the TSS states which traffic does not go through the VPN and which traffic does.

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

21.3. Test Activity

The assurance activity does not require testing activities. Therefore the **PASS** verdict is assigned.

21.4. Final Verdict

Due to all activities 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 this FDP_IFC_EXT.1 requirement.

22. FIA_AFL.1

22.1. Assurance activity

The assurance activity for FIA_AFL.1.1 requirement states 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.

The assurance activity for FIA_AFL.1.2 requirement states 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.*

22.2. Documentation review activity

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

22.3. Test Activity

22.3.1. Test 1 - User name and password

22.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- The audit function related to the logon, user account management and account lockouts must be enabled. To do that, the evaluator shall open a PowerShell terminal as administrator and type the following commands:

```
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
```

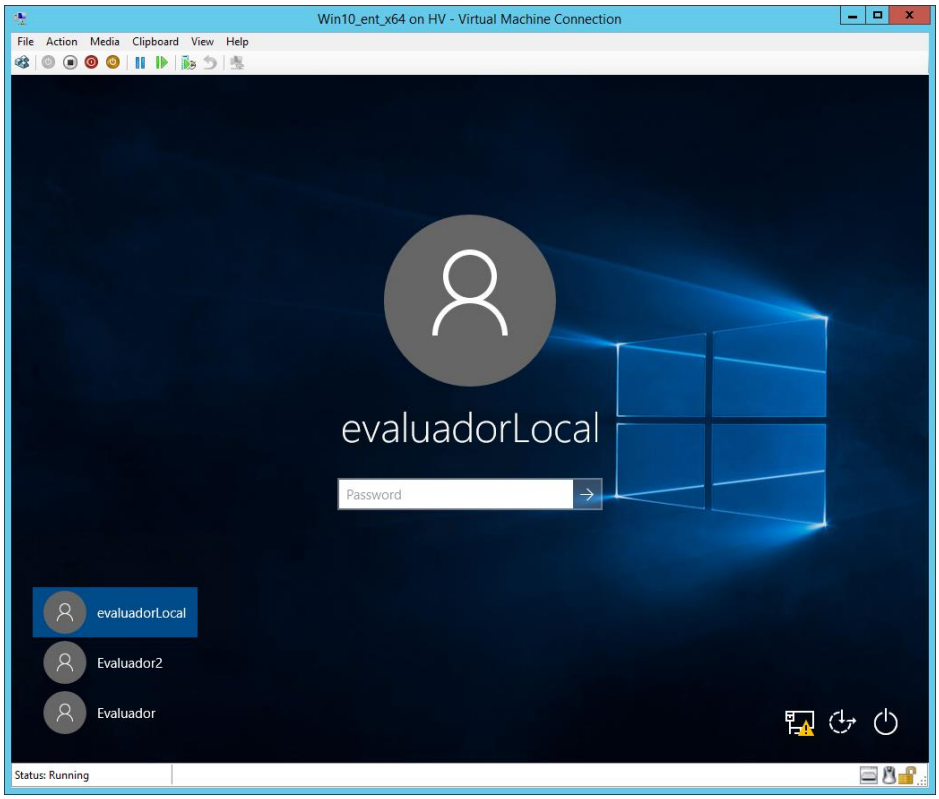
22.3.1.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. First of all, the evaluator shall configure the threshold for unsuccessful authentication attempts. To do that the evaluator shall carry out the following steps:
    a. Log in using a user with administrator rights.
    b. Open a PowerShell terminal as an administrator and execute the following command:

    net account /lockoutthreshold:3 /lockoutwindow:15 /lockoutduration:15

    c. The evaluator can set any value within the range defined in the security target (between 1 and 999). The evaluator has set the value at 3 for this test procedure.
2. Once the threshold has been configured, sign out the administrator session go back to the login screen.
3. The evaluator shall attempt to login using a user account without the administrator right and using an invalid password.

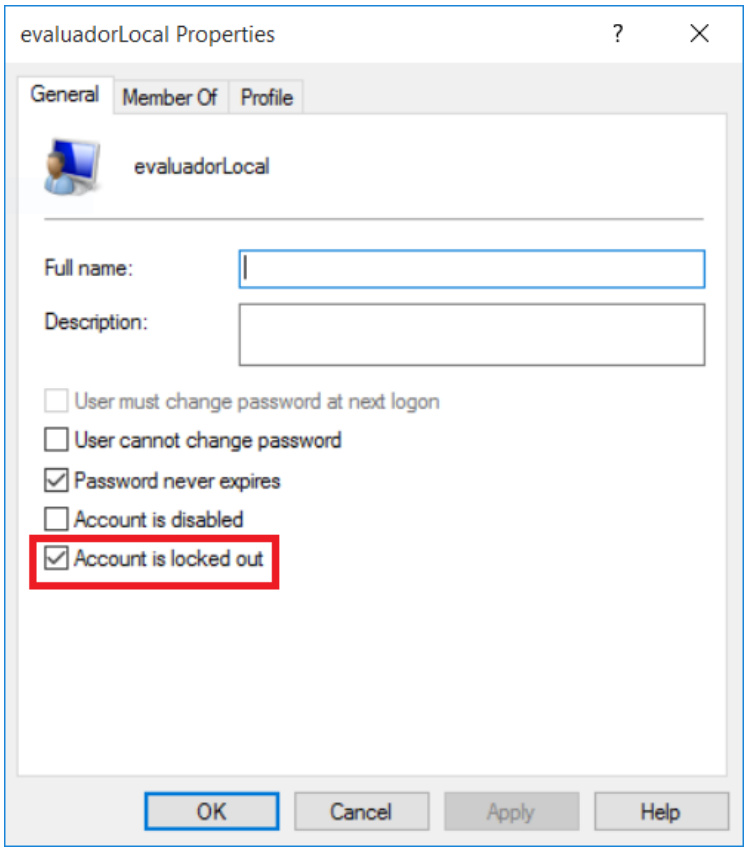


4. Repeat the step 3, until the configured threshold has been reached. Once this threshold is reached, an error message shall be shown.

5. After that, the evaluator shall perform a valid login attempt using a user account with administrator rights.

6. Right-click in the *System* button and open the Computer Management. Then, go to *System Tools->Local Users and Groups->Users*. Select the user account which has been locked.

7. Ensure that this account is locked. To do that, verify that the *Account is locked out* check box is checked.



8. Uncheck this checkbox in order to unlock the user account.

9. Finally, open a PowerShell terminal as administrator a type the following commands in order to obtain the generated audit events:

```
Get-EventLog Security -InstanceId 4740 -Newest 1 | fl *
Get-EventLog Security -InstanceId 4767 -Newest 1 | fl *
Get-EventLog Security -InstanceId 4625 -Newest 1 | fl *
```

10. Observe the result.

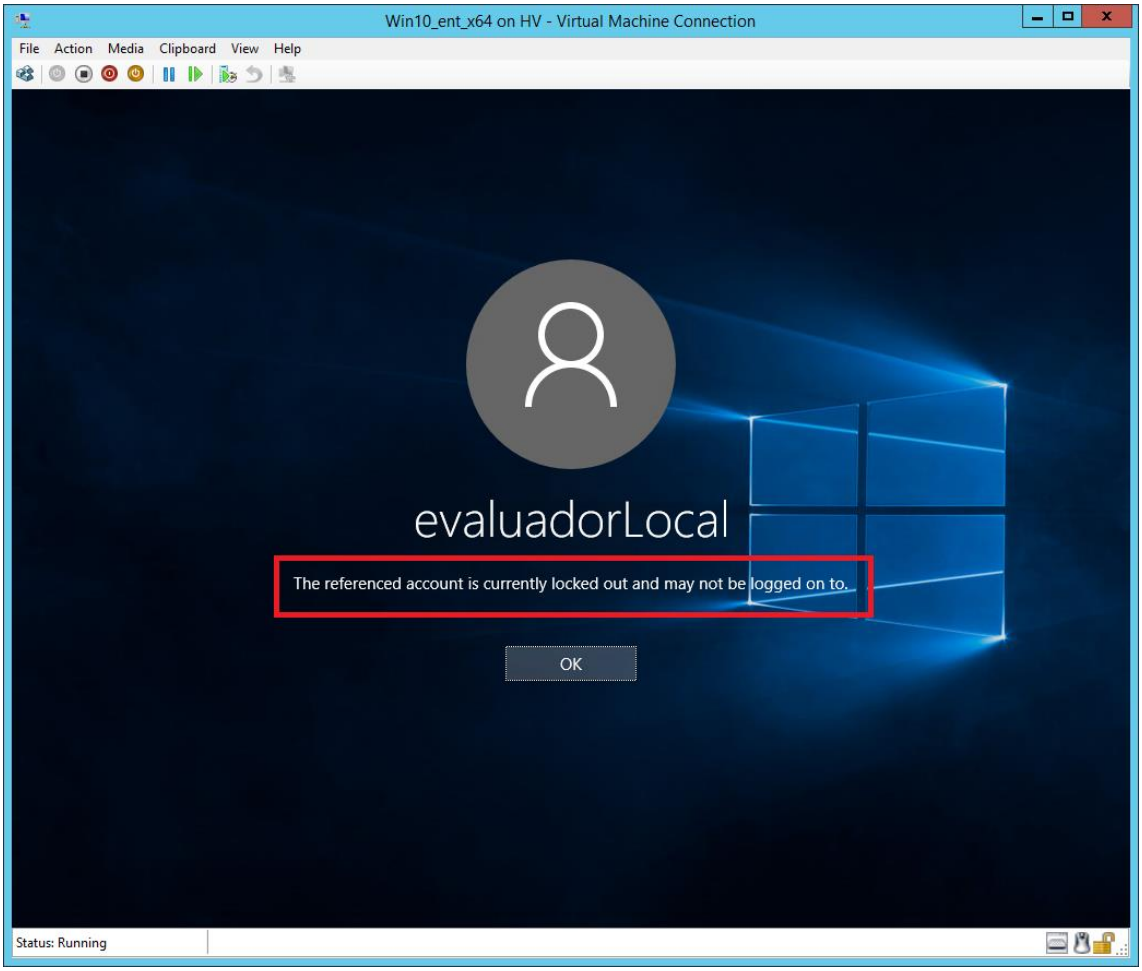
22.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results in all the tested platforms. The evaluator has carried out the steps defined above and has obtained the following results. After reaching the

configured threshold, the user account is locked and the following message is shown in the log in screen:



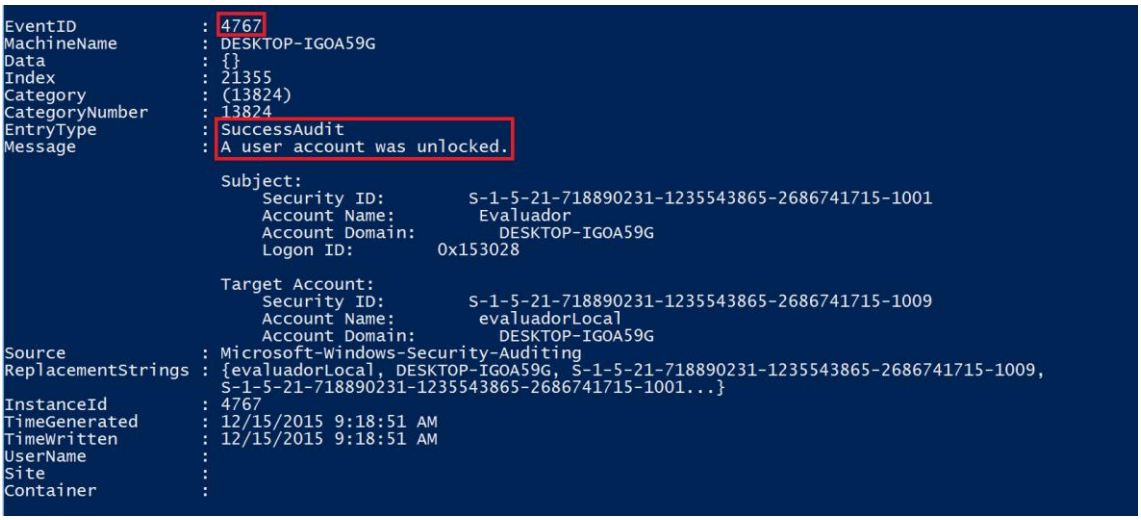
Once the user account has been locked, the evaluator has logged into the computer using a user account with administrator rights and has executed the commands defined in the step 9. For each unsuccessful authentication attempt the following audit event has been generated, indicating that there was an invalid authentication attempt:

```
EventID             : 4625
MachineName         : DESKTOP-IGOA59G
Data                : {}
Index               : 23124
Category            : (12546)
CategoryNumber      : 12546
EntryType           : FailureAudit
Message             : An account failed to log on.

                      Subject:
                          Security ID:        S-1-5-18
                          Account Name:        DESKTOP-IGOA59G$
                          Account Domain:        WORKGROUP
                          Logon ID:        0x3e7

                      Logon Type:            2

                      Account For Which Logon Failed:
                          Security ID:        S-1-0-0
                          Account Name:        evaluadorLocal
                          Account Domain:        DESKTOP-IGOA59G

                      Failure Information:
                          Failure Reason:        %%2307
                          Status:            0xc0000234
                          Sub Status:        0x0

                      Process Information:
                          Caller Process ID:    0x3e8
                          Caller Process Name:    C:\Windows\System32\svchost.exe

                      Network Information:
                          Workstation Name:    DESKTOP-IGOA59G
                          Source Network Address:    127.0.0.1
                          Source Port:        0

                      Detailed Authentication Information:
                          Logon Process:        User32
```

Once the configure threshold has been reached and the user account has been locked, the following audit event has been generated showing this fact:

```
EventID             : 4740
MachineName         : DESKTOP-IGOA59G
Data                : {}
Index               : 23074
Category            : (13824)
CategoryNumber      : 13824
EntryType           : SuccessAudit
Message             : A user account was locked out.

                      Subject:
                          Security ID:        S-1-5-18
                          Account Name:        DESKTOP-IGOA59G$
                          Account Domain:        WORKGROUP
                          Logon ID:        0x3e7

                      Account That Was Locked Out:
                          Security ID:        S-1-5-21-718890231-1235543865-2686741715-1009
                          Account Name:        evaluadorLocal

                      Additional Information:
                          Caller Computer Name:    DESKTOP-IGOA59G
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {evaluadorLocal, DESKTOP-IGOA59G, S-1-5-21-718890231-1235543865-2686741715-1009, S-1-5-18...}
InstanceId          : 4740
TimeGenerated       : 12/16/2015 9:37:49 AM
TimeWritten         : 12/16/2015 9:37:49 AM
UserName            :
Site                :
Container           :
```

Finally, the evaluator has unlocked the user account and the generated audit event showing this information is as follows:



22.3.1.4. Verdict

The evaluator has performed invalid authentication attempts using an invalid password 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 test results obtained during this test activity demonstrate the fulfillment of **Test 1** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 1**.

22.4. Final Verdict

Due to all applicable test activities 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 **FIA_AFL.1.1** and **FIA_AFL.1.2**

23. FIA_UAU.5.1

23.1. Assurance activity

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 on figure 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.*

23.2. Documentation review activity

23.2.1. Findings

Due to neither *authentication based on user name and a PIN that releases an asymmetric key stored in OE-protected storage* nor *authentication based on X.509 certificates* are selected in the final version of the security target, a documentation review activity is not necessary for this requirement.

23.3. Test Activity

23.3.1. User name and password - Test 1 & Test 2

23.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- The audit function related to the logon must be enabled. To do that, the evaluator shall open a PowerShell terminal as administrator and type the following commands to enable auditing:

  auditpol /set /subcategory:"logon" /success:enable /failure:enable
  wevtutil cl Security

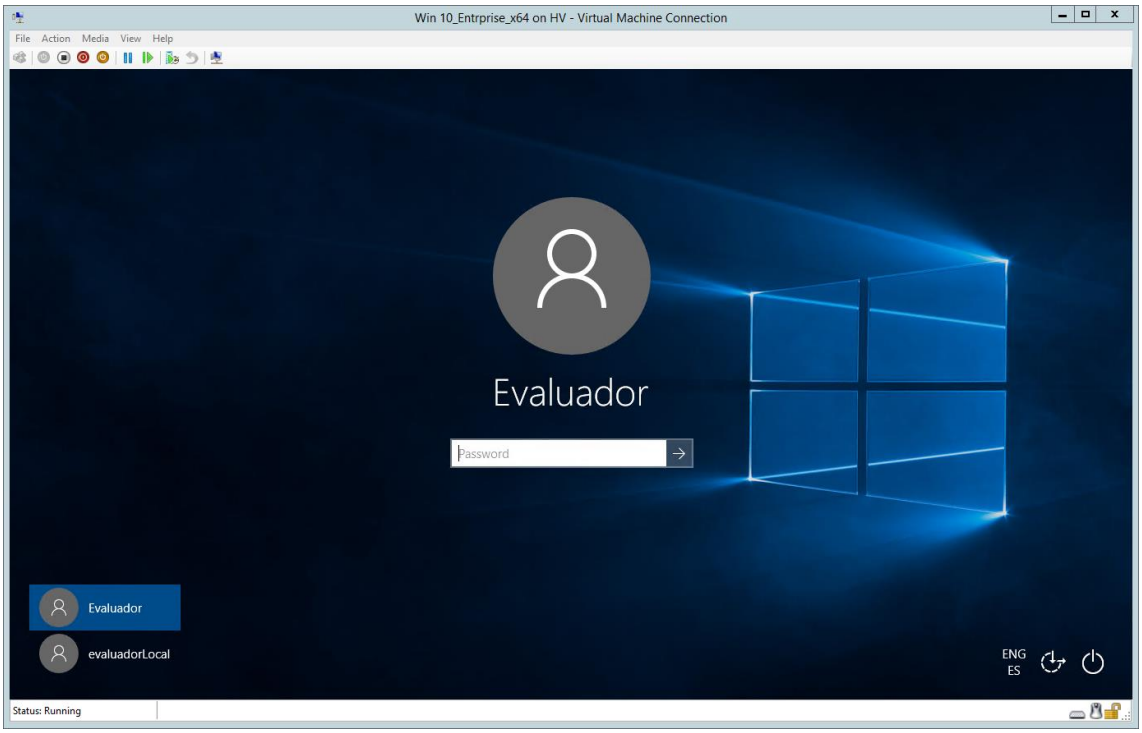
- An administrator account shall exist. This account shall belong to default *Administrators* group.

- A standard user account shall exist in the system, in order to perform valid and invalid authentication attempts.

23.3.1.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

4. Boot the TOE and select the user account used for this test. Enter the correct password for the account in the text box.



5. Observe that the user is authenticated successfully and its session is loaded. After that open a PowerShell terminal as administrator and type the following command in order to get the generated audit event: *Get-EventLog Security -InstanceID 4624 -Newest 1 | fl **

6. Observe the result of the event.

7. Sign out the current session and go back to the login screen. Select the same user and enter a wrong password in the text box.

8. Observe the result.

23.3.1.3. Results

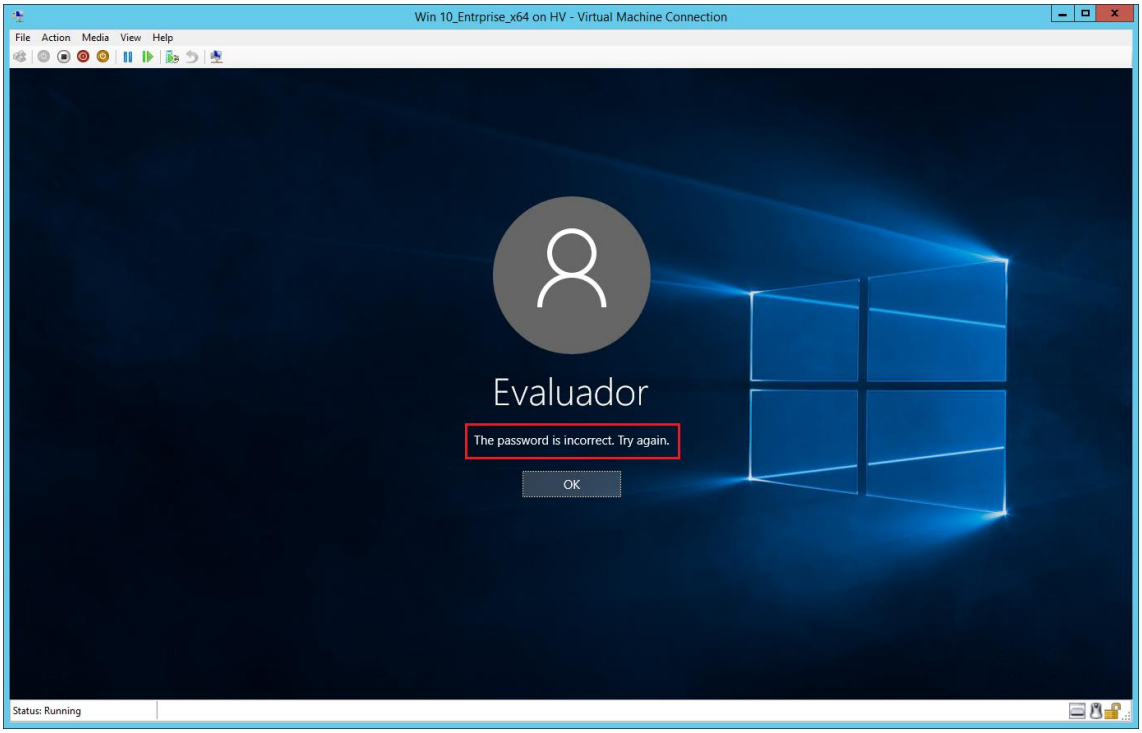
The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has been able to access to the user account using the correct password. Additionally, the evaluator has ensured that the related audit event has been generated:

```
EventID             : 4624
MachineName         : DESKTOP-DOIB9FD
Data                : {}
Index               : 2575
Category            : (12544)
CategoryNumber      : 12544
EntryType           : SuccessAudit
Message             : An account was successfully logged on.

                      Subject:
                          Security ID:        S-1-5-18
                          Account Name:         DESKTOP-DOIB9FD$
                          Account Domain:         WORKGROUP
                          Logon ID:         0x3e7

                      Logon Information:
                          Logon Type:         2
                          Restricted Admin Mode:    -
                          Virtual Account:        %%1843
                          Elevated Token:         %%1843

                      Impersonation Level:        %%1833

                      New Logon:
                          Security ID:        S-1-5-21-2653389645-2162558090-1581609492-1001
                          Account Name:         Evaluador
                          Account Domain:         DESKTOP-DOIB9FD
                          Logon ID:         0x11ffc2
                          Linked Logon ID:        0x11ffa6
                          Network Account Name:     -
                          Network Account Domain:     -
                          Logon GUID:         {00000000-0000-0000-0000-000000000000}

                      Process Information:
                          Process ID:         0x34c
                          Process Name:         C:\Windows\System32\svchost.exe

                      Network Information:
                          Workstation Name:     DESKTOP-DOIB9FD
                          Source Network Address:     127.0.0.1
                          Source Port:        0
```

On the other hand, the evaluator has performed an authentication attempt using an invalid password. The obtained result is as follows:



23.3.1.4. Verdict

The evaluator has performed an authentication attempt using the user name and the correct password and it has been completed successfully. The evaluator has performed another authentication attempt but this time using an invalid password. The evaluator has ensured that this authentication attempt failed.

Due to this, the evaluator considers that, the test results obtained during this test activity demonstrate the fulfillment of **User name and password - Test 1 and Test 2** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **User name and password - Test 1 and Test 2**.

23.4. Final Verdict

Due to all applicable test activities 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 **FIA_UAU.5.1.**

24. FIA_X509_EXT.1.1

24.1. Assurance activity

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 selfsigned 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.)*
- ***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.)*

24.2. Documentation review activity

24.2.1. Findings

The section 6.4.1 of the "Windows 10 Security Target" document includes a URL where describes the API used to check the validity of the X509 certificated. The URL *"https://msdn.microsoft.com/en-us/library/windows/desktop/aa380252(v=vs.85).aspx "* can be found the "Certificate and Certificate Store Functions" where explains the functions used to manage the X509 certificates. In addition, each function has the section *"Requirements"* where list the name of the component (Crypt32.lib and Wincrypt.h) and the *"Minimum operating system supported".*

On the other hand, the TSS provides information about the certification path algorithm, this is described in the *"RFC 5280"* section 5.1.4.3.

24.2.2. 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.

24.3. Test Activity

24.3.1. Test 1

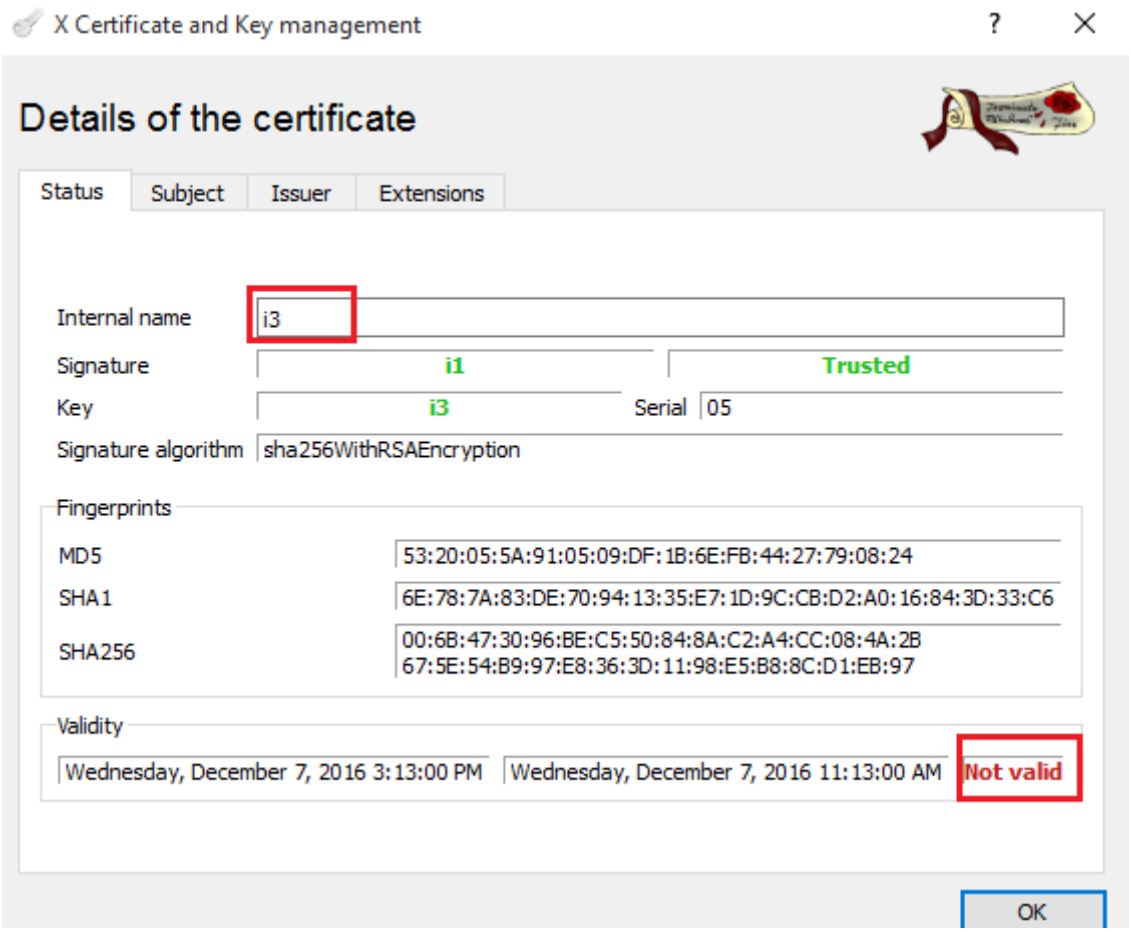
24.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA3, (i3) (This certificate is not valid)
- CN = www.test.com, (Server (s2))

- CN = www.test.com, (Client (c3))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA3 (chainTest1a).

The certificates listed above form an invalid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA3 -> (Server, Client)"*. Due to the certificate *"IntermediateCA3"* is not valid, as it can be appreciated in the next picture.



The following certificates also are necessary to fulfill the assurance activity for the Test 1.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).
- File that contains the RootCA and IntermediateCA2 (chaintest1c).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. The *"chainTest1c"* file contains an incomplete certification

path *"RootCA -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

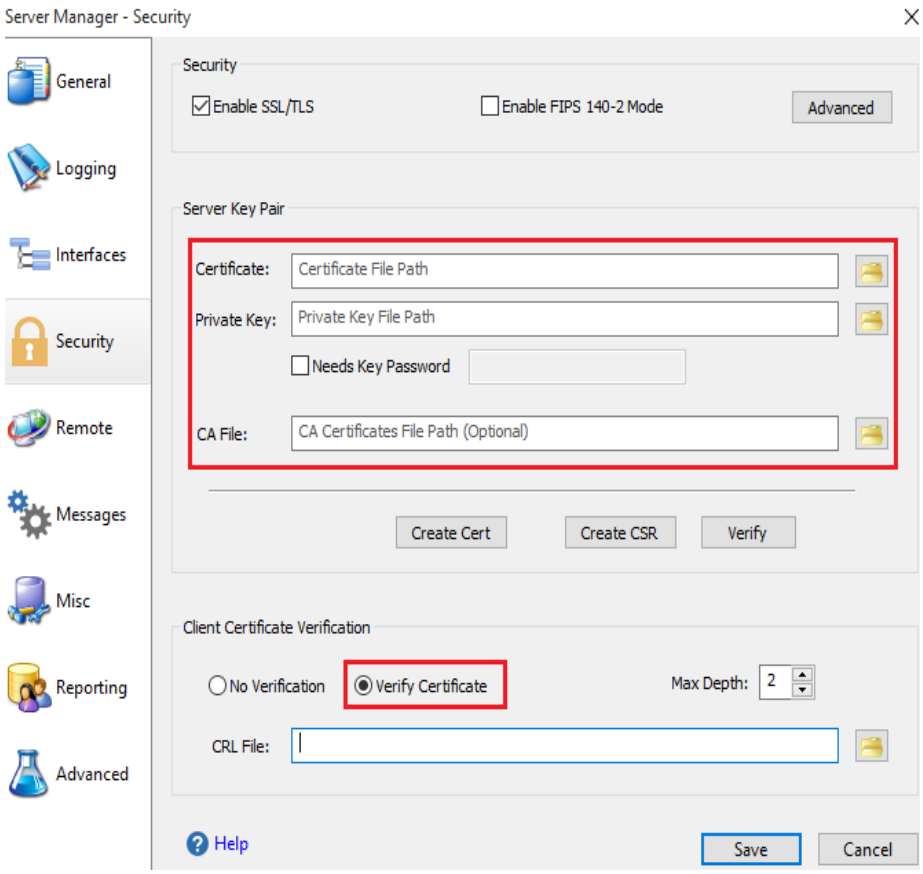
24.3.1.2. Procedure

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

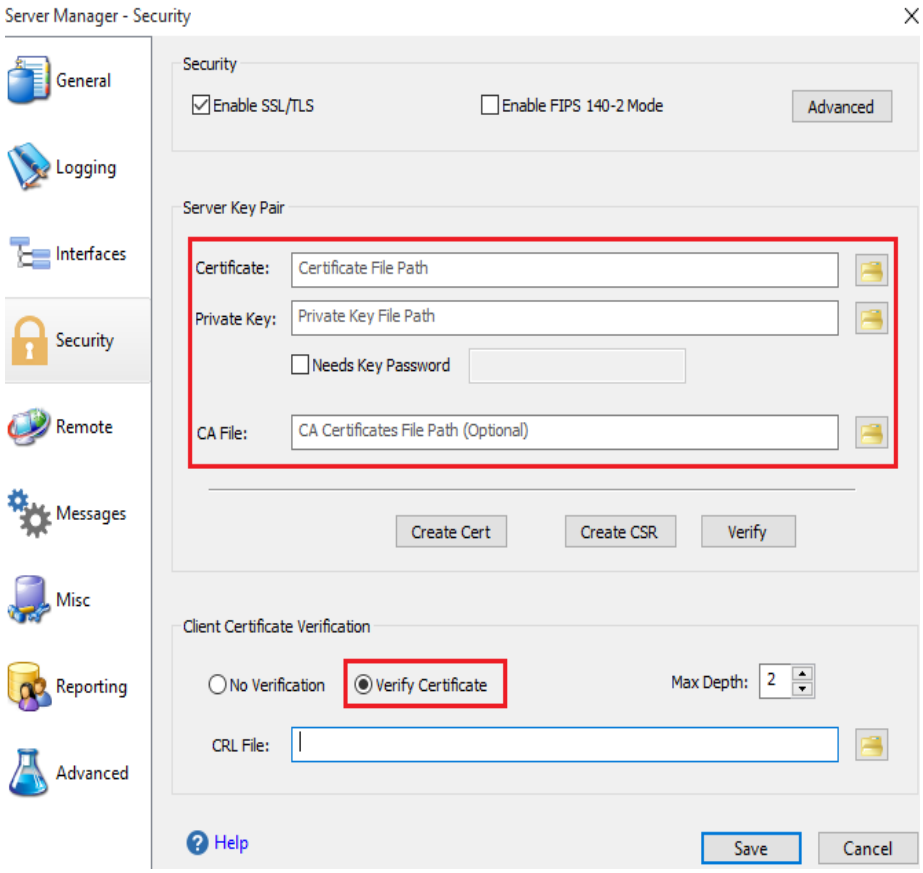
1. Add the *"RootCA.pkcs"* and *"c3.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
    - Select *"Personal"*.

- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c3.pkcs"* is stored and type the key of the *"pkcs"*.

2. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"s2.crt"*, the *"s2.pem" and "ca.crt"*. The *"chaintest1a.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA3"*.
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press *"save"*.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press *"save"*.

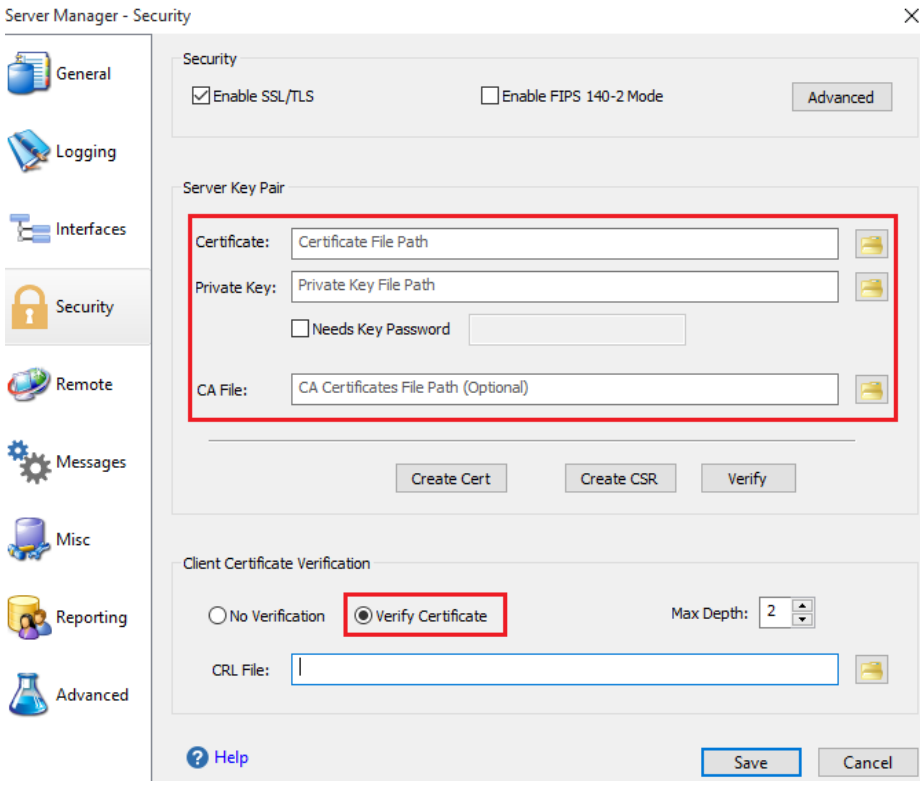


3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
4. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
5. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.
6. Delete from the Client Machine the *"c3.pkcs"* certificate using the *"mmc"* service.
7. Add the *"c1.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.
8. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chainTest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2"*.
    - In Client Certificate Verification select "*Verify Certificate*".
    - Press *"save"*.
    - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
    - Press *"save"*.



9. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
10. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
11. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.
12. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chainTest1c.pem"* contains the *"RootCA and IntermediateCA2"*.
    - In Client Certificate Verification select "*Verify Certificate*".
    - Press *"save"*.

- Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
- Press *"save"*.



13. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
14. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
15. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

24.3.1.3. Results

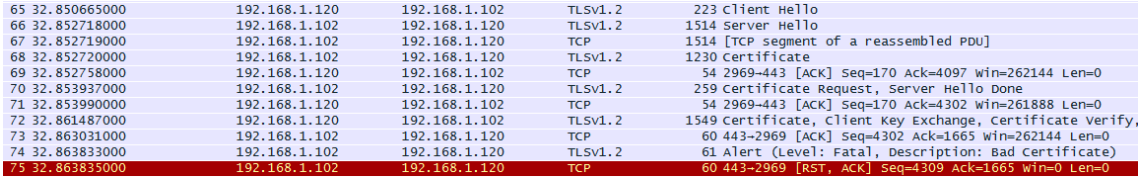
The test has been performed in the following platforms:

- Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.

- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured in the *"Wireshark"* application for the first part of the Test 1, It can be appreciated the *"Certificates"* message sent by the Server Machine. This message contains all certificates included in the *"chainTest1a"* file loaded in the "*Cerberus FTP server enterprise "*, as it can be appreciated in the following picture.

The Client Machine responses with a *"Certificate, Client Key Exchange, Certificate Verify"* message. The following picture shows all steps of the handshaking process.



The Server Machine cannot validate the client certificate due to the *"IntermediateCA3"* is not valid, the server sends an *"Alert"* message and *"RST, ACK"* packet to the client. When the client receives the *"Alert"* message and the *"RST, ACK"* packet, It does not continue the handshaking process. The browser shows to the user the following screen.

This page can't be displayed

- Make sure the web address https://www.test.com is correct.
- Look for the page with your search engine.
- Refresh the page in a few minutes.

Fix connection problems

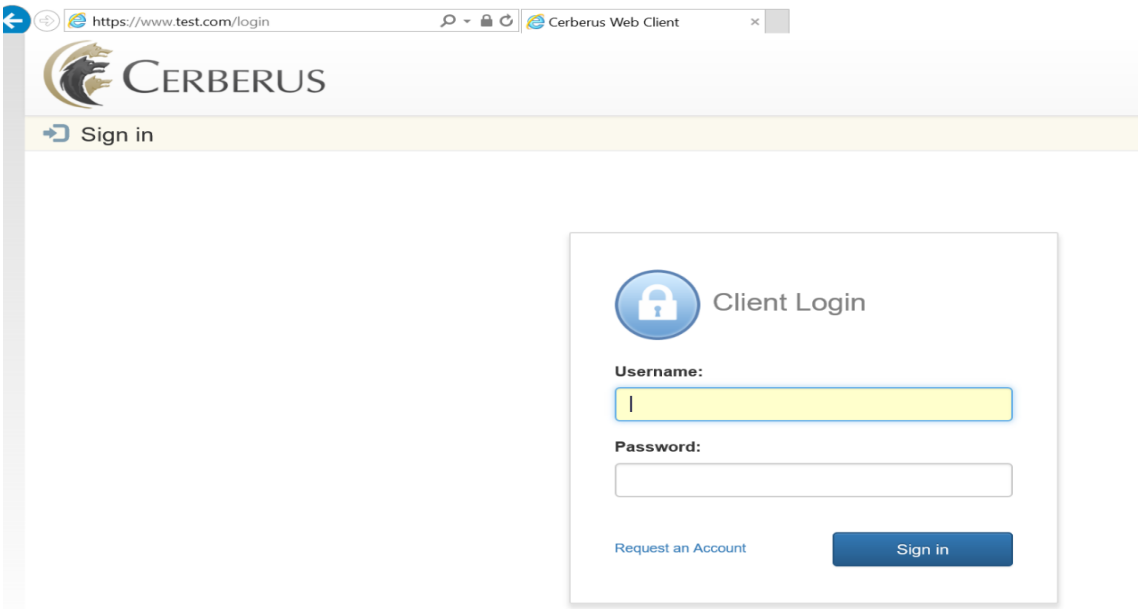
Analyzing the packets captured during the execution of the activities listed in the Protection Profile for the second part of the Test 1, it can be appreciated the *"Certificates"* message sent by the Server Machine. This message contains all certificates included in the *"chainTest1b"* file loaded in the *"Cerberus FTP server enterprise* ", as it can be appreciated in the following picture.

```
⊟ TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4033
  ⊟ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 4029
      Certificates Length: 4026
    ⊟ Certificates (4026 bytes)
        Certificate Length: 879
      ⊞ Certificate (pkcs-9-at-emailAddress=s1,id-at-commonName=www.test.com,id-at-organizationalUnitName=s1,id
        Certificate Length: 1054
      ⊞ Certificate (pkcs-9-at-emailAddress=i2,id-at-commonName=intermediateCA2,id-at-organizationalUnitName=i2
        Certificate Length: 1045
      ⊞ Certificate (pkcs-9-at-emailAddress=i1,id-at-commonName=intermediateCA1,id-at-organizationalUnitName=i1
        Certificate Length: 1036
      ⊞ Certificate (pkcs-9-at-emailAddress=ca,id-at-commonName=RootCA,id-at-organizationalUnitName=ca,id-at-or
```

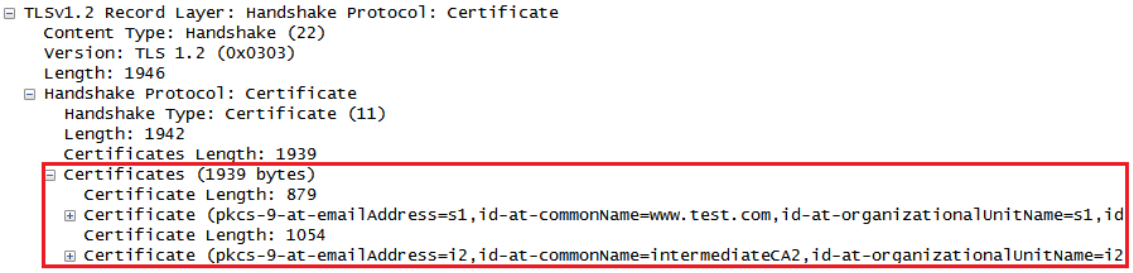
The Client Machine responses with a *"Certificate, Client Key Exchange, Certificate Verify"* message. The following picture shows all steps of the handshaking process.

19	0.851370000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
20	0.852982000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
21	0.852984000	192.168.1.102	192.168.1.120	TCP	1514 [TCP segment of a reassembled PDU]
22	0.852985000	192.168.1.102	192.168.1.120	TLSv1.2	1230 Certificate
23	0.853092000	192.168.1.120	192.168.1.102	TCP	54 2974→443 [ACK] Seq=170 Ack=4097 Win=262144 Len=0
24	0.854551000	192.168.1.102	192.168.1.120	TLSv1.2	438 Certificate Request, Server Hello Done
25	0.854612000	192.168.1.120	192.168.1.102	TCP	54 2974→443 [ACK] Seq=170 Ack=4481 Win=261632 Len=0
26	0.868078000	192.168.1.120	192.168.1.102	TLSv1.2	1549 Certificate, Client Key Exchange, Certificate Verify, Change Cipher
27	0.869748000	192.168.1.102	192.168.1.120	TCP	60 443→2974 [ACK] Seq=4481 Ack=1665 Win=262144 Len=0
28	0.896274000	192.168.1.102	192.168.1.120	TLSv1.2	1216 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
29	0.896322000	192.168.1.120	192.168.1.102	TCP	54 2974→443 [ACK] Seq=1665 Ack=5643 Win=262144 Len=0
30	0.903699000	192.168.1.120	192.168.1.102	TLSv1.2	363 Application Data

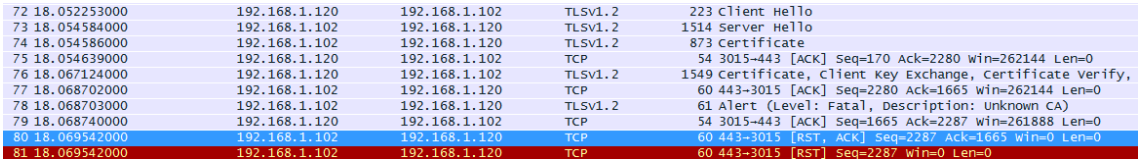
In the upper picture can be appreciated that the connection has been established successfully, because the client sends *"Application Data"* packets. In addition, the browser shows the login page of the "*Cerberus FTP server enterprise*", the following picture illustrates the browser response.



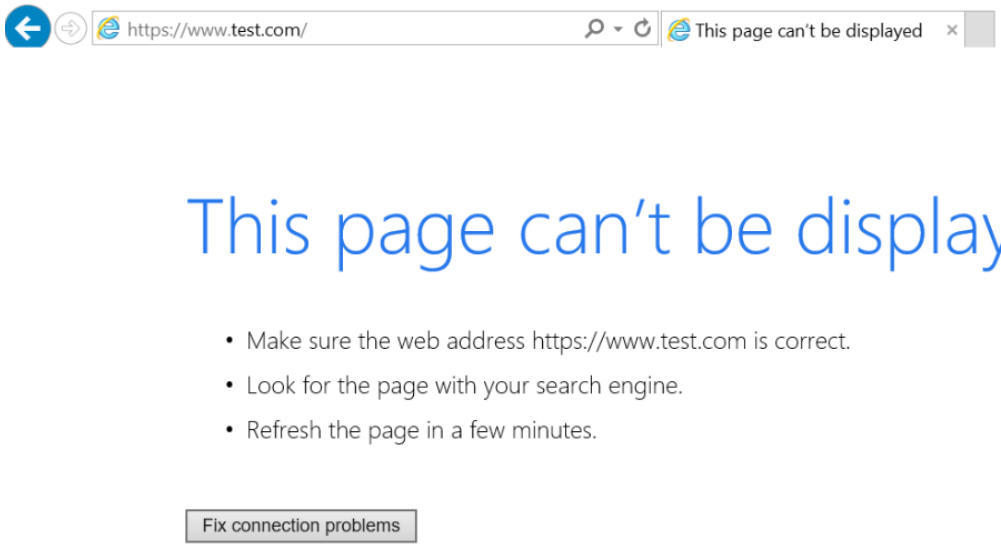
As it can be observed in the packets captured for the last part of the Test1, the "*Certificate*" message sent by the server does not contain the *"IntermediateCA1"*. The following picture shows the certificates in the *"Certificate"* message.



The Client Machine responses with a *"Certificate, Client Key Exchange, Certificate Verify"* message. The following picture shows all steps of the handshaking process.



When the server checks the certification path of the client's certificate, it cannot find the *"RootCA"* due to the *"IntermediateCA1"* being missing. The Server Machine sends an *"Alert"* message indicating that *"Unknown CA"* and a *"RST, ACK"* packet. In addition, the browser shows to the user the following screen:



24.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

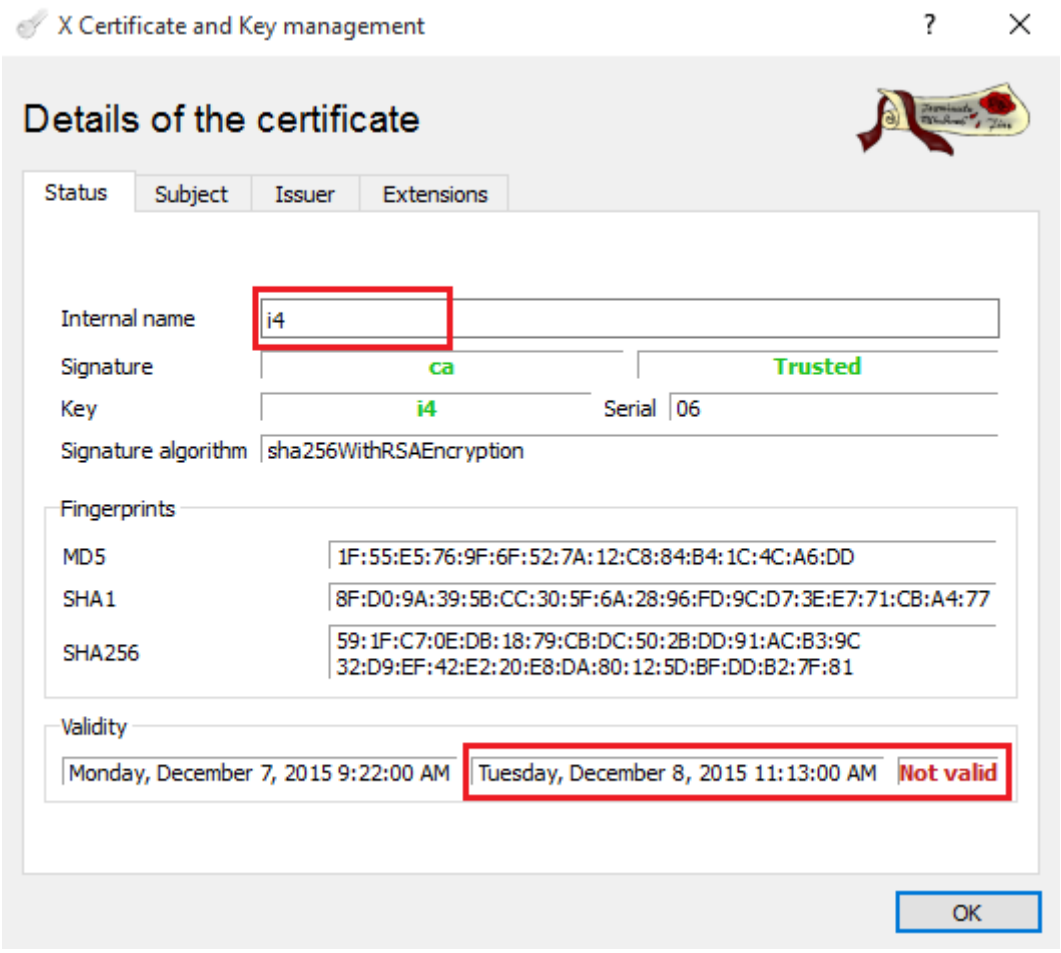
24.3.2. Test 2

24.3.2.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA4, (i4) (Expired certificate)
- CN = IntermediateCA5, (i5)
- CN = www.test.com, (Server (s3))
- CN = www.test.com, (Client (c4))
- File that contains the RootCA, IntermediateCA4 and IntermediateCA5 (chainTest2).

The certificates listed above, form an invalid certification path *"RootCA -> IntermediateCA4 -> IntermediateCA5 -> (Server, Client)"*. Due to the certificate *"IntermediateCA4"* is not valid as it can be appreciated in the next picture.



The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

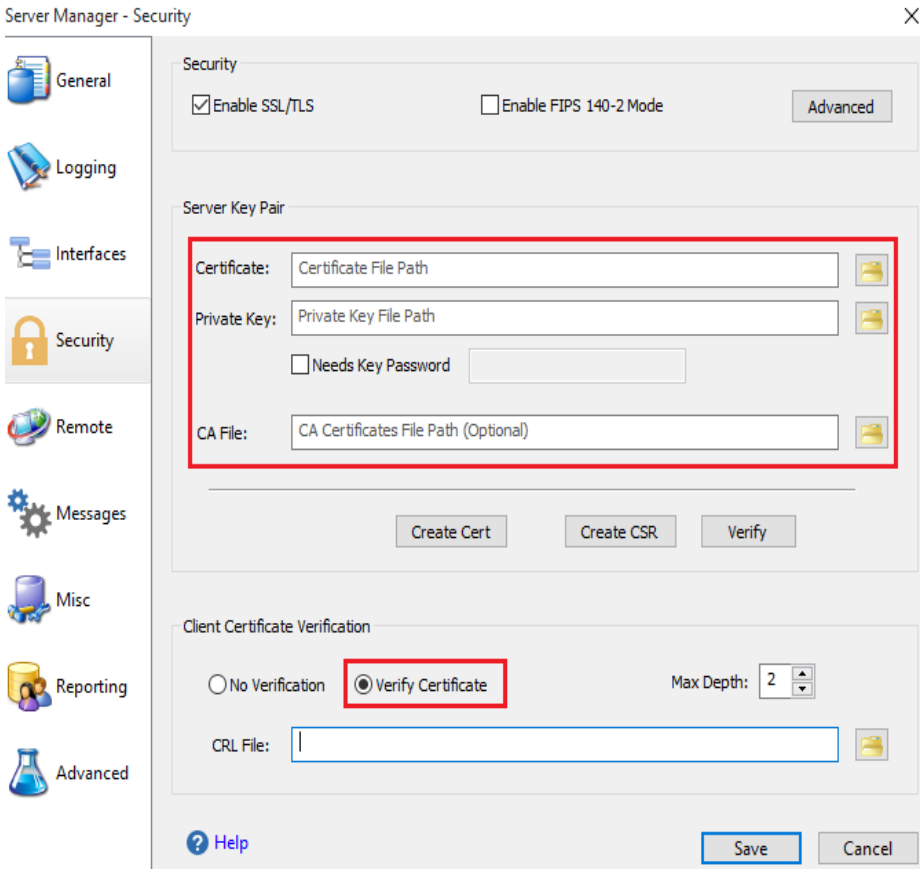
In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

24.3.2.2. Procedure

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

1. Add the *"RootCA.pkcs"* and *"c4.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c4.pkcs"* is stored and type the key of the *"pkcs"*.
2. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s3.crt"*, the *"s3.pem" and "ca.crt"*. The *"chainTest2.pem"* contains the *"RootCA, IntermediateCA4 and IntermediateCA5"*.
    - In Client Certificate Verification select "*Verify Certificate*".
    - Press *"save"*.
    - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
    - Press *"save"*.



3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
4. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
5. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

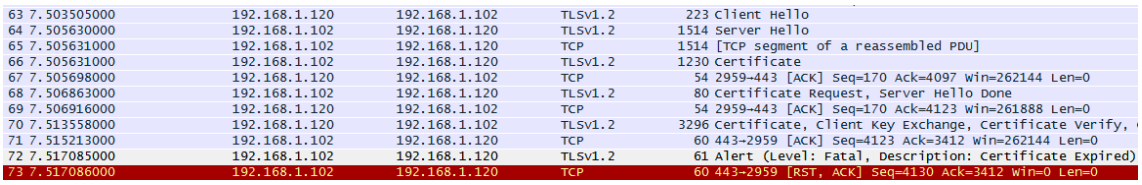
24.3.2.3. Results

The test has been performed in the following platforms:

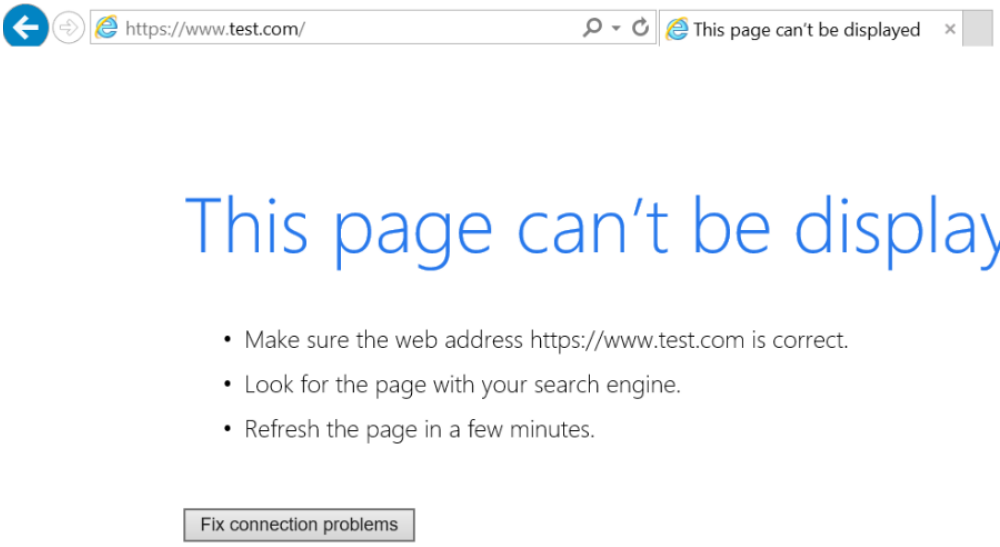
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured in the *"Wireshark"*, it can be appreciated the *"Certificates"* message sent by the Server Machine. This message contains all certificates included in the *"chainTest2"* file loaded in the "*Cerberus FTP server enterprise* ", as it can be appreciated in the following picture.

The Client Machine responses with a *"Certificate, Client Key Exchange, Certificate Verify"* message. The following picture shows all steps of the handshaking process.



When the server checks the certification path of the client's certificate, it cannot build a valid path to the *"RootCA"* due to the *"IntermediateCA4"* is expired. The Server Machine sends an *"Alert"* message indicating that *"Certificate Expired"* and a *"RST, ACK"* packet. . In addition, the browser shows to the user the following screen:



24.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.3.3. Test 3

24.3.3.1. Setup

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Root CA Server Machine (Windows Server 2012 R2).
- Intermediate CA1 Server Machine (Server 2012 R2).
- Intermediate CA2 Server Machine (Server 2012 R2).
- Client Machine (Windows 10 x64 Enterprise Edition).

These four machines are in the same network with the following configuration:.

- Root CA Server Machine IP = 192.168.248.126.
- Intermediate CA1 Server Machine IP = 192.168.248.130.
- Intermediate CA2 Server Machine IP = 192.168.248.133.
- Client Machine IP = 192.168.248.135.

In all servers machines shall be installed a *"Wireshark"* application to verify the OCSP server performance.

We shall have a chain of at least four certificates: the node certificate to be tested which resides in the Client Machine, two intermediates CAs (Intermediate CA1 and CA2 server machines) and a self-signed Root CA.

In all machines shall add the next line *"192.168.248.126 <name_of_domain_server_root_CA>"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc.*

In first place, the Root CA Server Machine shall has installed and configured the next roles and features:

- Active Directory Domain Services.
- Active Directory Certificate Services.
- Web Server(IIS).
- OCSP server.

The Active Directory Domain Services shall be installed following the next steps:

1. Open the Server Manager from the task bar.

2. From Server Manager Dashboard select Add roles and Features.
3. Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
4. The current server is selected by default. Click next.
5. From the *"Server Roles"* screen check a mark in the box *"Active Directory Domain Services"*. An additional pop-up screen must appear, explaining all the features required to install the Domain Services.
6. Click *"Add features"*.
7. A new screen is shown and click next.
8. On *"Select features"*, click Next.
9. Review the information on the Active Directory Domain Services tab and click Next.
10. Finally, click Install.

Once the Active Directory Domain Services role is installed, it can be configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select the *"Notification Icon"* from the top of the Server Manager. A subsection appears and then click *"Promote this server to a domain controller"*.
3. From the Deployment Configuration tab select *"Add a new forest"*. Insert your root domain name into the Root Domain name field (For example, winserver.org). And, click Next.
4. Select a Forest and Domain functional level. In this case, both must be *"Windows Server 2012"*. The Domain Name System(DNS) is selected by default.
5. Write a password in the DSRM password field and then click Next.
6. A warning pop-up screen appears about DNS options. Click Next.
7. Confirm the NetBIOS name (For example, WINSERVER). And click Next.
8. In the Configure the location of SYSVOL, maintain the default configuration.
9. In Review Options Screen click Next.
10. The system will check to ensure all prerequisites are installed on the system for several minutes. If the system passes these checks, click Install.

The Active Directory Certificate Services shall be installed and configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select Add roles and Features.
3. Select *"Role-based or features-based installation"* from the *"Installation Type"* screen, and click Next.
4. Click in the radio button *"Select a server from the server pool"*. The current server is selected by default. Click next.
5. From the *"Server Roles"* screen check a mark in the box *"Active Directory Certificate Services"*. An additional pop-up screen shall appear explaining all the features required

to install the Domain Services. Click *"Add features"*. A new screen is shown and click next.

6. On *"Select features"*, click Next.
7. Review the information on the Active Directory Certificate Services tab and click Next.
8. From the *"Server role services"* screen check a mark in the box *"Certification Authority"*. Click Next.
9. Finally, click Install.
10. When the installation is finished, click *"Configure Active Directory Certificate Services on the destination server"* type in blue color.
11. On Credentials, write the name of the administrator in Credentials field. Click Next.
12. From the *"Role services"* screen check a mark in the box *"Certification Authority"*. Click Next.
13. On the Setup Type screen, select Enterprise CA. Click Next.
14. On the CA Type screen, select RootCA. Click Next.
15. On Private Key screen, select create a new private key. Click Next.
16. On Cryptography for CA, select SHA256. Others attributes are selected by default. Click Next.
17. On CA Name, all fields are selected by default. Click next.
18. On validity Period select 25 years. Click Next.
19. On CA Database, the database locations are selected by default. Click Next.
20. Review the information on the Confirmation tab and click Configure.
21. On Results page, click Close.

The Web Server (IIS) shall be installed and configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select Add roles and Features.
3. Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
4. The current server is selected by default. Click next.
5. From the *"Server Roles"* screen check a mark in the box *"Web Server (IIS)"*. An additional pop-up screen must appear explain all the features required to install the Domain Services. Click "Add features". A new screen is shown and click next.
6. On "Select features", Click Next.
7. Review the information on the Web Server Role (IIS) tab and click Next.
8. On Select Role Services. Click Next.
9. Finally, click Install.

The OCSP server shall be installed following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select Add roles and Features.
3. Select "Role-based or features-based" installation from the "Installation Type" screen, and click next.

4. The current server is selected by default. Click next.
5. From the "Server Roles" select the box "Active Directory Certificate Services". And then check a mark in the box "Online Responders". An additional pop-up screen must appear explaining all the features required to install the online responder. Click "*Add features".* A new screen is shown and click next.
6. On "Select features" screen, click Next.
7. Click Install.
8. When the installation is finished, click "Configure Active Directory Certificate Services on the destination server" typed in blue.
9. On "Credentials" screen, click Next.
10. On "Role services" screen, select Online Responder and then click next.
11. Finally, click Configure.

The OCSP server shall be configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select "AD CS" from the left panel.
3. Right-Click on the server. And select "Certification Authority".
4. In the left panel, Right-Click in your server and click properties.
5. Go to the extensions tab, and from the Select Extension drop-down box choose Authority Information Access (AIA).
6. Click the "Add" button and type the FQDN address for the OSCP responder in the form http://w2012.winserver.org/ocsp. Click OK.
7. Select the OCSP http://w2012.winserver.org/ocsp and check the box "Include in the online certificate status protocol(OCSP) extension" and click Apply. A pop-up appear, you must choose Yes.
8. From the select extension drop-down box choose CRL Distribution Point(CDP). Click Add.
9. Type the FQDN address where resides the CRL revocation list. In this case http:// w2012.winserver.org /revocation_list.crl. Click OK.
10. Select the address from the list then check the box "Include in the CDP extension of issued certificates".
11. Click OK then YES to restart the certificate services.
12. Copy the file revocation list file (revocation_list.crl) from C:\Windows\system32\CertSrv\CertEnroll into C:\Windows\inetpup\wwwroot.
13. Open the Server Manager from the task bar.
14. From Server Manager Dashboard select "AD CS" from the left panel.
15. Right-Click on the server. And select "Certification Authority".
16. In the left panel, Right-Click in "Certificate Templates" and click Manage.
17. Locate the OCSP Response Signing template, right-click it and choose properties.
18. Go to Security tab. And give to OCSP server the Enroll permission to the OCSP certificate template.

19. Back to the Certification Authority. Right-Click in Certificate Templates and choose New -> Certificate Template to Issue.
20. From the list select the OCSP Response Signing template and click OK.
21. Go to the Server Manager..
22. From Server Manager Dashboard select "AD CS" from the left panel.
23. Right-Click on the server. And select "Online Responder Management".
24. Right-click the Revocation Configuration object and choose Add Revocation Configuration.
25. On the Getting started page click Next to continue.
26. On the Revocation Configuration page, give a friendly name. Click Next.
27. On the Select CA Certificate Location page, select "Select a certificate for an existing enterprise CA". Click Next.
28. On Choose CA certificate page, browse the Certification Authority certificate.
29. Leave the "Auto-Enroll for an OCSP signing certificate box enabled. The certificate template is selected by default. Click Next.
30. Finally, click Finish.
31. After few seconds, the OCSP Responder should be working just fine.

In second place, the Intermediate CA1 and CA2 Server Machine shall have installed and configured the next roles and features:

- Active Directory Domain Services.
- Active Directory Certificate Services.

The Active Directory Domain Services is installed and configured as in the Root CA server machine.

In order to install the Active Directory Certificate Services in the intermediate CA1 and CA2 Machines, the steps listed for the Root CA Machine case shall be followed except:

- On the CA type screen we shall select *"Subordinate CA"* instead of *"Root CA"*.
- On the Certificate request screen, a file type .req is downloaded in the path show in "File name" field.

The Intermediate CA1 request shall be submitted to the Root CA certification authority follows the next steps:

1. Copy the file .req into the Root CA server machine.
2. Open the Server Manager from the task bar.
3. From Server Manager Dashboard select "AD CS" from the left panel.
4. Right-Click on the server. And select "Certification Authority".
5. In the left panel, Right-Click in your server, select "All tasks" and then click "Submit new Request...".
6. Browse the file .req. And click Next.
7. Save the signed request in somewhere location.

8. Copy the certificate into the Intermediate CA1.
9. In the Intermediate CA1, go to "Certification Authority".
10. In the left panel, right-click in the server. Select "All tasks" and then click "Install CA Certificate..."
11. Select the signed certificate. Click next.
12. Finally, right-click in the server, select "All tasks" and then click "Start Service".

The intermediate CA2 request shall be submitted to the Intermediate CA1 certification authority follow the same steps as the intermediate CA1.

1. Copy the file .req into the Intermediate CA2 server machine.
2. Open the Server Manager from the task bar.
3. From Server Manager Dashboard select "AD CS" from the left panel.
4. Right-Click on the server. And select "Certification Authority".
5. In the left panel, Right-Click in your server, select "All tasks" and then click "Submit new Request...".
6. Browse the file .req. And click Next.
7. Save the signed request in somewhere location.
8. Copy the certificate into the Intermediate CA2.
9. In the Intermediate CA2, go to "Certification Authority".
10. In the left panel, right-click in the server. Select "All tasks" and then click "Install CA Certificate..."
11. Select the signed certificate. Click next.
12. Finally, right-click in the server, select "All tasks" and then click "Start Service".

In third place, the client machine does shall sends an OCSP request to validate his certificate (<certificate>.cer). This certificate shall be created by the intermediate CA2 follow the next steps.

1. In the Intermediate CA2, Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
2. At the command prompt, type *"mmc"* and press *"ENTER".*
3. On the *"File"* menu, click *"Add/Remove Snap-in".*
4. In the Add standalone Snap-in dialog box, select *"Certificates"*.
5. Press *"Add".*
6. In the Certificates Snap-in dialog box, select *"Computer Account"* and click next.
7. Select Local Computer and Press *"Finish".*
8. *Press "OK".*
9. In the console tree, right-click the Personal folder. Select "All tasks" and then click "Request new certificate...",

10. Click Next in the "Certificate Enrollment" screen. And then click Next.
11. Check a mark in the box "Domain Controller". Click Enroll.
12. Click Finish.
13. In the folder "Personal -> Certificates". Right-Click in the certificate created, select "All tasks" and then click export.
14. Click next in all steps and then save the certificate in the Desktop, this certificate shall be imported into the client machine.

We shall obtain the file.cer for each CA server:

- Root CA.- <CA_Root_Certficate>.cer
- Intermediate CA1.- <Intermediate_CA1_certificate>.cer
- Intermediate CA2.-<Intermediate_CA2_certificate>.cer

The certificates Root CA, Intermediate CA1 and Intermediate CA2 shall be exported in "der" format following the next steps.

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select "AD CS" from the left panel.
3. Right-Click on the server. And select "Certification Authority".
4. In the left panel, Right-Click in your server. And click "Properties".
5. In the tab General, select the CA certificate. And click "View certificate".
6. On the Certificate screen, select tab "Details".
7. Click Next in the wizard screen. And then select "DER encoded binary X.509 (.CER)". Click Next.
8. Select somewhere location to the file. Click Next.
9. Click finish.

In client machine, shall be introduce the Root CA, Intermediate CA1 and Intermediate CA2 certificate in the " Trusted Root Certification Authorities " following the next steps:

1. Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
2. At the command prompt, type *"mmc"* and press *"ENTER"*.
3. On the *"File"* menu, click *"Add/Remove Snap-in"*.
4. In the Add standalone Snap-in dialog box, select *"Certificates"*.
5. Press *"Add"*.
6. In the Certificates Snap-in dialog box, select *"Computer Account"* and click next.
7. Select Local Computer and Press *"Finish"*.
8. *Press "OK"*.

9. In the console tree, right-click the Trusted Root Certification Authorities folder. Select "All tasks" and then click "Import...",
10. Click Next in the "Certificate Import Wizard" screen. And then click Next.
11. Browse the Certificate. Click Next.
12. Click Next. And then Click Finish

These steps should be followed for each Certificate (Root CA, Intermediate CA1 and Intermediate CA2).

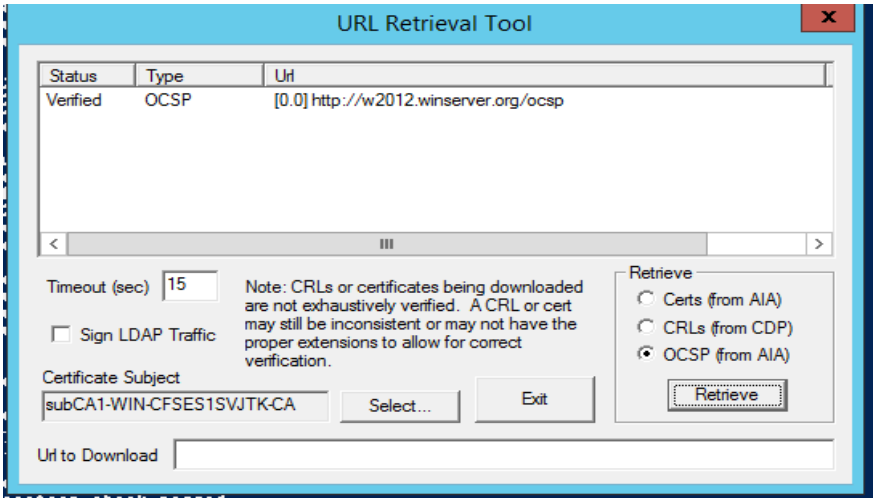
In Intermediate CA2 machine, it should have in the "Trusted Root Certification Authorities" the Root CA and Intermediate CA1 certificates to introduce these to the "Trusted Root Certification Authorities" following the same steps as the client machine.

In Intermediate CA1 machine, it should have in the "Trusted Root Certification Authorities" the Root CA certificate. To introduce them to the" Trusted Root Certification Authorities" follow the same steps as the client machine.

24.3.3.2. Procedure

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

1. In the three machines (intermediateCA1, intermediateCA2 and Client), run the following commands with their respective certificates.
    - Certutil -url <certificate>.cer
    - In the screen opened, check that the status is "Verified" to the "CRLs (from CDP)", Certs(from AIA) and "OCSP (from AIA)"



2. In the client machine, open Command prompt line.
3. Run the next command with the client certificate.
    - Certutil -verify <certificate>.cer
4. In the intermediate CA1 run the next command in a Command prompt line:
    - Certutil -verify <Intermediate_CA1_certificate>.cer
5. In the intermediate CA2 run the next command in a Command prompt line:

- Certutil -verify <Intermediate_CA2_certificate>.cer

6. In the Intermediate CA2 machine, revoke the client machine certificate following the next steps.
    - Open the Server Manager from the task bar.
    - From Server Manager Dashboard select "AD CS" from the left panel.
    - Right-Click on the server. And select "Certification Authority".
    - In the left panel, select "Issued Certificates".
    - Right-click in the client machine certificate. Select "All tasks" and click "revoke certificate".
    - Select "Certificate Hold" in the "Reason Code" drop-down box. Click Yes.
    - In the left panel, right-click Revoked Certificates. Select "All tasks" and then click "Publish".
    - Select "New CRL". Click OK.
7. In the client machine, open a command prompt line and then run the followings commands:
    - Certutil -urlcache * delete
    - Certutil -verify <certificate>.cer
8. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.
9. Unrevoke the client machine certificate following the next steps:
    - Open the "Certification Authority".
    - In the left panel, select "Revoked Certificates".
    - Right-click the client machine certificate, select "All tasks" and then click "Unrevoke Certificate.
10. In the Intermediate CA1, revoke the Intermediate CA2 certificate follow the steps listed in 6.
11. In the intermediate CA2, open a command prompt line and then run the following commands:
    - Certutil -urlcache * delete
    - Certutil -verify <intermediate_CA2_certificate>.cer
12. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.
13. In the client machine, open a command prompt line and then run the following commands:
    - Certutil -urlcache * delete
    - Certutil -verify <intermediate_CA2_certificate>.cer
14. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.
15. In the intermediate CA1 Machine, unrevoke the intermediate CA2 certificate following the next steps listed in 9.
16. In the Root CA, revoke the Intermediate CA1 certificate follow the steps listed in 6.
17. In the intermediate CA1, open a command prompt line and then run the following commands:
    - Certutil -urlcache * delete
    - Certutil -verify <intermediate_CA1_certificate>.cer

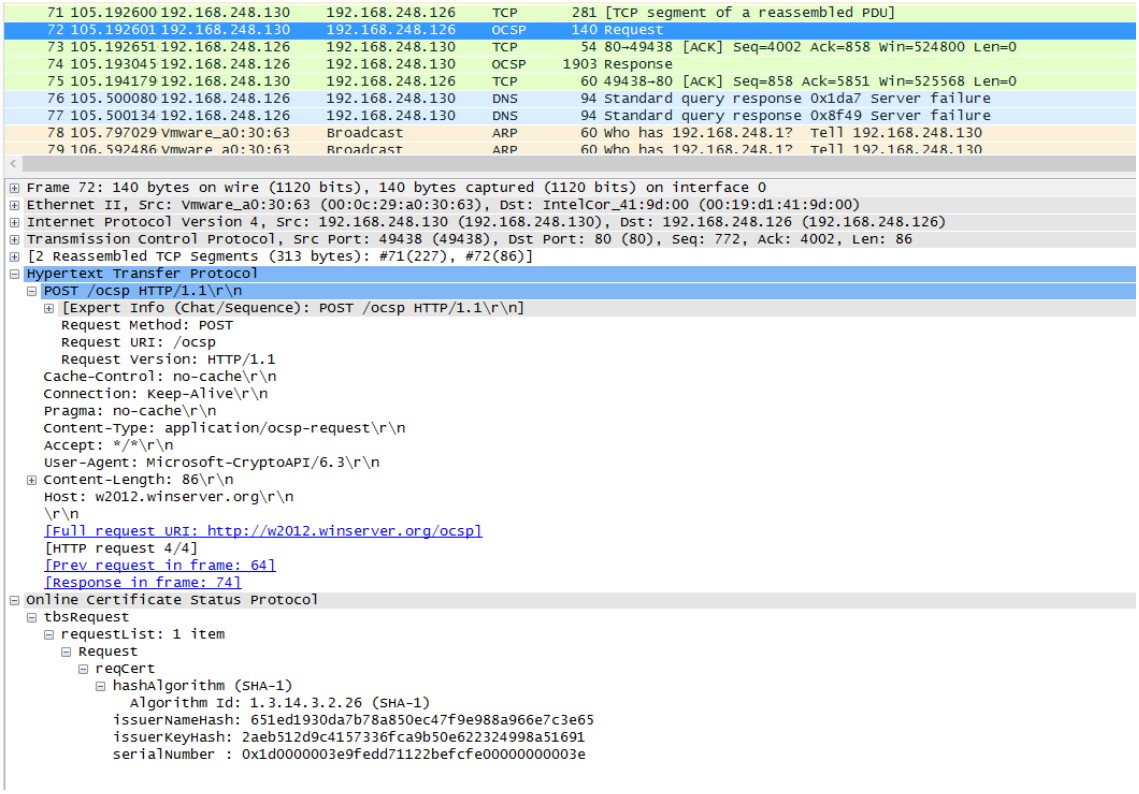
18. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.
19. In the intermediate CA2, open a command prompt line and then run the following commands:
    - Certutil -urlcache * delete
    - Certutil -verify <intermediate_CA2_certificate>.cer
20. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.
21. In the client machine, open a command prompt line and then run the following commands:
    - Certutil -urlcache * delete
    - Certutil -verify <certificate>.cer
22. The output of the "Certutil -verifiy <certificate>.cer" must be revoked.

24.3.3.3. Results

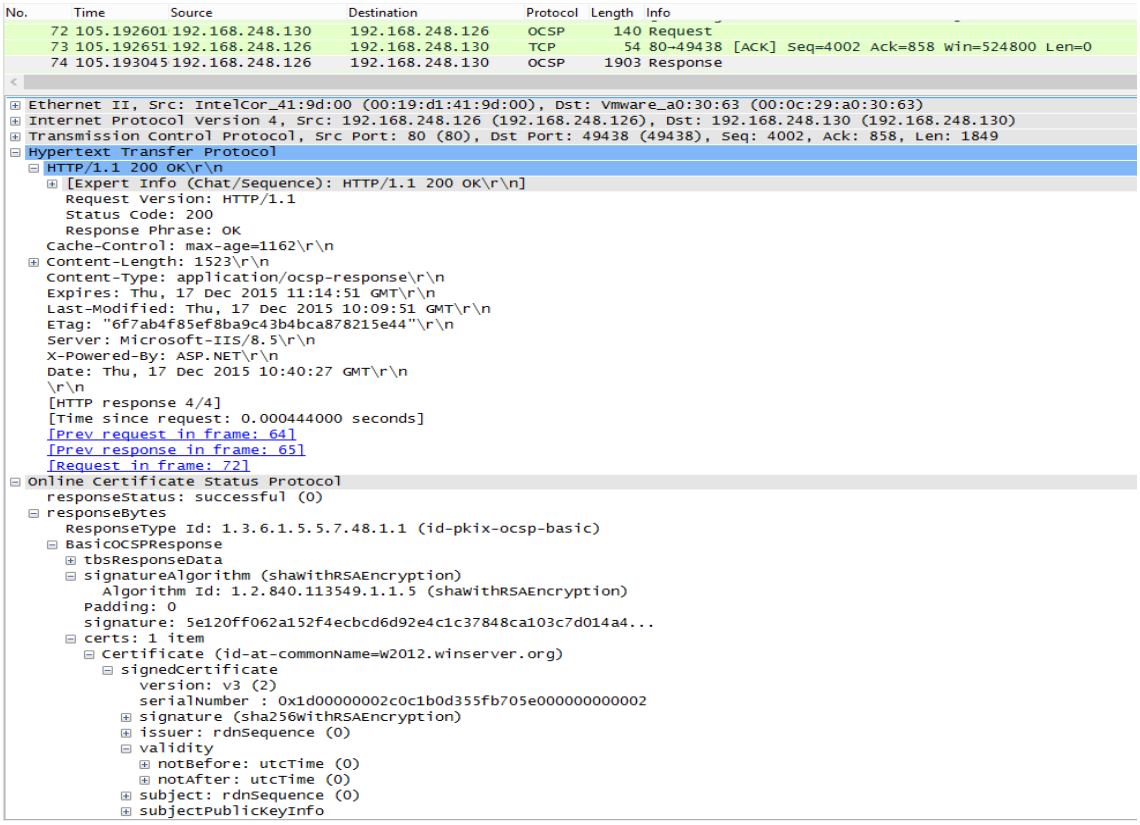
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

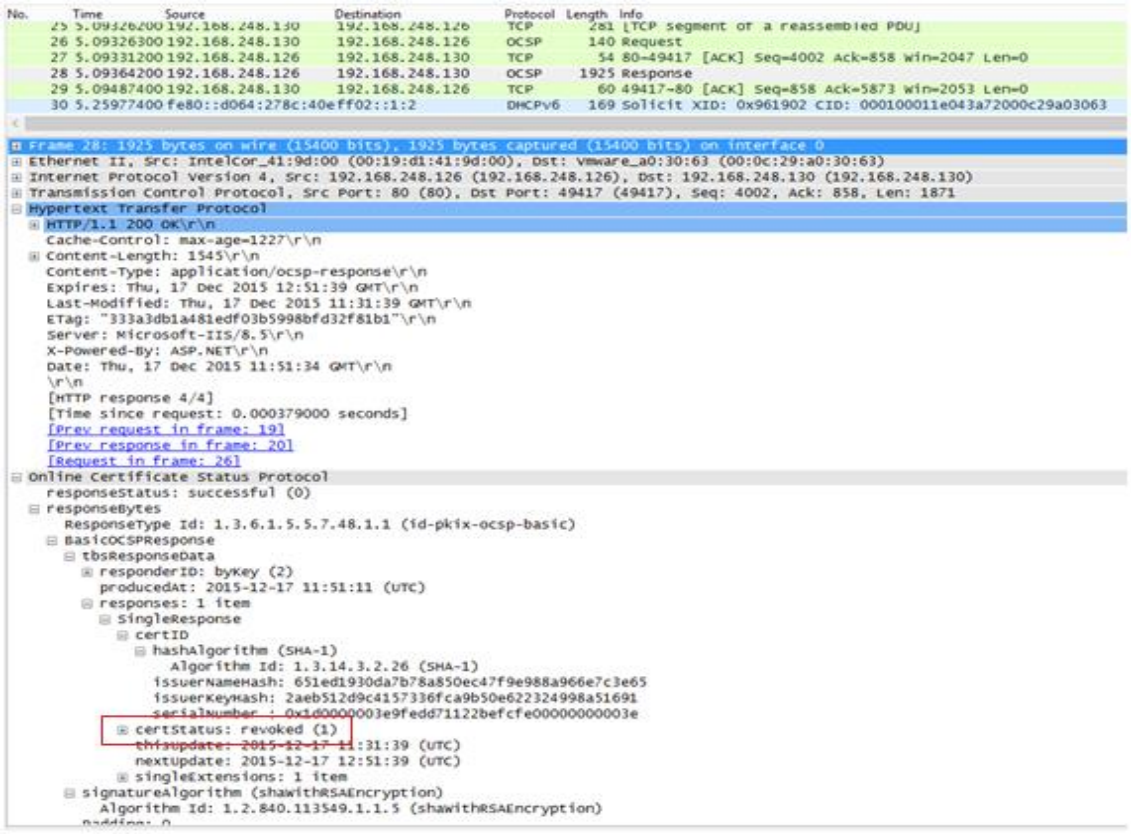
Analyzing the packets captured in the *"Wireshark"* when the command *"certutil -verify <certificate>.cer"* is performed in the Client Machine, it can be appreciated that the OCSP request is sent correctly to the CA server. In the following picture can be appreciated the *"OCSP Request"*.



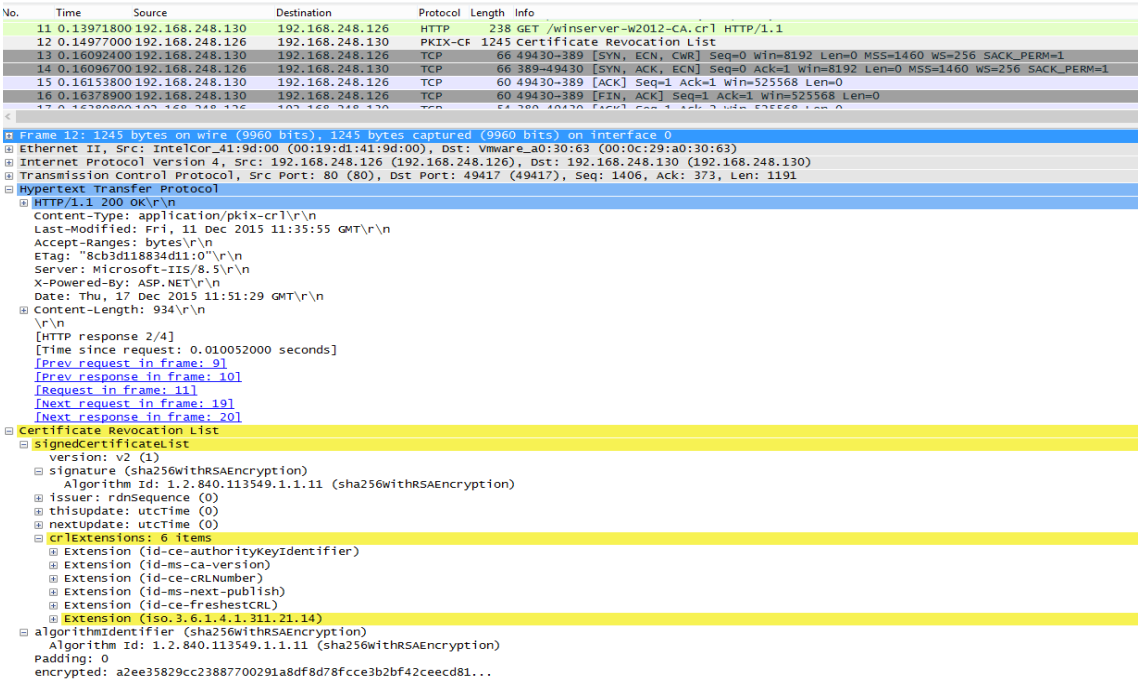
When the OCSP Server receives the Request, this verifies if the certificate is valid. The next capture shows the *"OCSP Response"* when the certificate is not revoked.



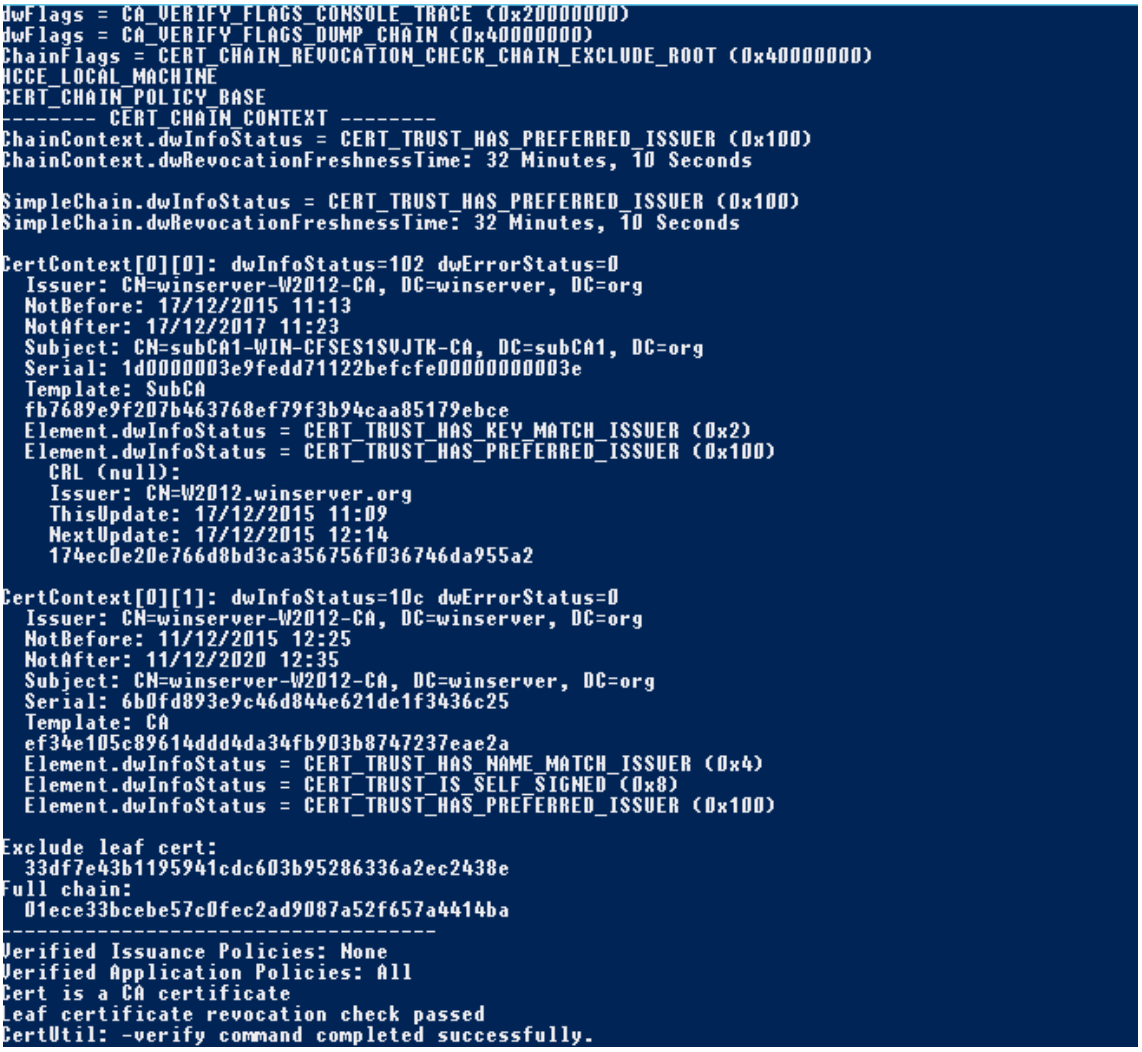
In case that the certificate is revoked, the "*OCSP Response*" message indicates clearly the status of the Client certificate, as it can be appreciated in the following picture.



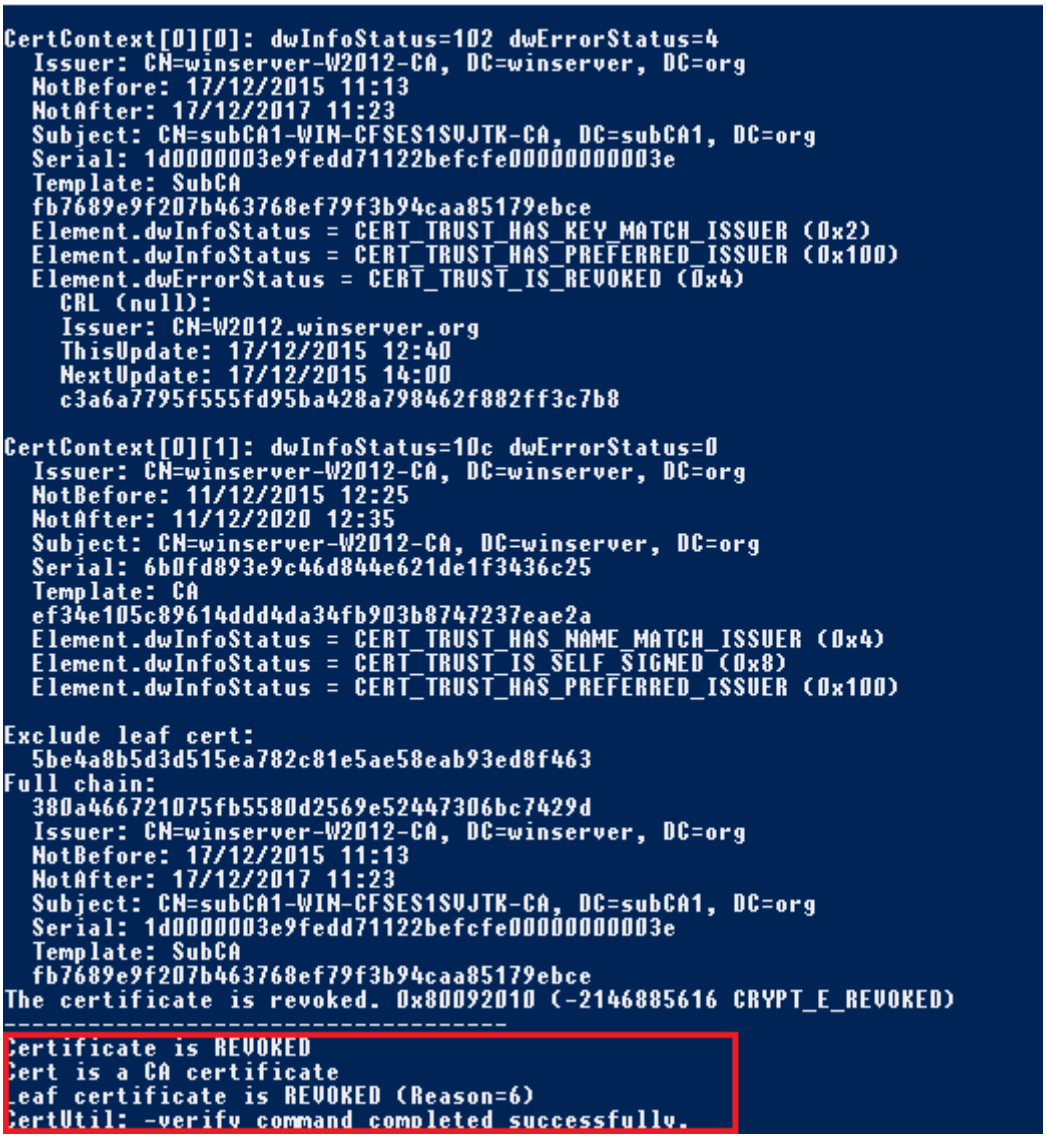
When the *"certutil -verify <certificate>.cer"* is performed, this download the CRL in the Client Machine, as it can be appreciated in the next picture.



In the following picture can be appreciated the output for the *"certutil"* tool when the Client certificate is not revoked.



In case that the certificate is revoked, the *"certutil"* output indicates clearly the status of the Client certificate, as it can be appreciate in the following picture.



24.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.3.4. Test 4

24.3.4.1. Setup

The scenario to perform the assurance activities according to the Protection Profile is composed by the following machine:

- Client and Server Machine (Windows 10 x64 Enterprise Edition).

This machine uses the IP 127.0.0.1 *"localhost"* to install the OCSP responder server. In the client machine, unzip tool *"OCSP-Test4.rar"* in one folder.

Uncompressed files in the folder must be:

- OcspServer.exe, regressaddon.dll, commonlib.dll. - Listens for Ocsp requests, generates and signs(using a designated key) response.
- OcspSigner.pfx. - A key with the OCSP signing purpose used with OcspServer.exe to demonstrate the OCSP response succeeds.
- ServerAuthSigner.pfx. - A key with the Server Authentication purpose used with OcspServer to demonstrate the OCSP response fails.
- cc-CC-CA1-CA.cer. - The public key of the root CA used to issue the above two signing keys.
- OcspTest.cer. - A test certificate for which to request certificate status using the certutil.exe utility. The certificate includes an AIA URL as “http://localhost:2025/ocsp”.

24.3.4.2. Procedure

In order to perform this test, the evaluator has followed the steps listed below:

1. Import the cc-CC-CA1.CA.cer into the Trusted Root Certification Authorities local machine store:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER".*
    - On the *"File"* menu, click *"Add/Remove Snap-in".*
    - In the Add standalone Snap-in dialog box, 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, right-click the Trusted Root Certification Authorities.
    - Select "All tasks" and then click "Import" to import the cc-CC-CA1.CA.cer. Click Next.
    - Browse the file cc-CC-CA1.CA.cer in the unzip folder. Click Next.
    - Click Next and then click "Finish".
2. Create a firewall exception to allow UDP access on the port 2025 for the OcspServer.exe program.
    - Click *"Start"*, click *"Run"*, type *"control panel"* and then click *"OK"*.

- Select Windows Firewall.
- In the left panel, select "Advanced settings".
- In the left panel, click Inbound Rules.
- In the right panel, click "New Rule...".
- On "Rule type" screen, select "Port". Click Next.
- On Protocol and Ports select "UDP" and type the port 2025 in "Specific local ports". Click Next.
- On "Action" screen select "Allow the connection". Click Next.
- On "Profile" screen, select all options and click Next.
- Type the rule name "UDP exception" in "Name" field.
- Click Finish.

3. Open a command window and start the OCPS listener server with the following command (in this case, use the OcspSigner.pfx (a certificate with the OCSP signing purpose)):
   - *"OcspServer.exe -name CN=OcspSigner -signer OcspSigner.pfx -port 2025 -responder regressaddon.dll"*
4. Open another command windows and request certificate status on the test certificate status with the following command:
   - *"CertUtil.exe -verify OcspTest.cer"*
5. See the information displayed by the *"Certutil"* tool and *"OcspServer"* tool, in both command windows.
6. The *"OcspServer"* and *"CertUtil"* display a success response.
7. Stop the *"OcspServer"* tool by issuing a *"Crtl-c"* break command.
8. Start the OCSP listener server with the following command(in this case, ServerAuthSigner.pfx (A certificate with the Server Authentication purpose)):
   - *"OcspServer.exe -name CN=ServerAuthSigner -signer ServerAuthSigner.pfx -port 2025 -responder regressaddon.dll"*
9. Flush the cache of certificate revocation check response by issuing the following command:
   - *"CertUtil -UrlCache * delete"*
10. The *"CertUtil"* outputs a list of cached results that are deleted.
11. Re-Request certificate status with the following command:
    - *"CertUtil.exe -verify OcspTest.cer"*
12. Both of tools displays some information:
    - *"OcspServer"* tool: display the request information and the encoded response.

- *"CertUtil"* tool: display a failed response indicating that *"The revocation function was unable to check revocation because the certificate revocation server is offline"*.

24.3.4.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

The following certificates have been used OCSP server:

- OcspSigner.pfx. - A certificate with the OCSP signing purpose.
- ServerAuthSigner.pfx. - A certificate with the Server Authentication purpose.

When a OCSP request is sent to the OCSP Server (running with the OcspSigner.pfx certificate) using the *"certutil"* tool, this responses if the certificate is *"good"* or *"revoked"*. The following picture shows the output of the *"certutil"* for a valid certificate.

```
C:\Users\LGS\Desktop\OSCP\OCSP-Test4>CertUtil.exe -verify OcspTest.cer
Issuer:
    CN=cc-CC-CA1-CA
    DC=cc
    DC=com
  Name Hash(sha1): 7a829013878b260f80d54906ade49ce52175d6ab
  Name Hash(md5): 8c14397b7c40ee502ef199bf5d1135c7
Subject:
    CN=OcspTest
  Name Hash(sha1): ecc1bba2fc86513dc618c4c491d7890ef5ff7e28
  Name Hash(md5): b975a72c983763725bf764b5d596f075
Cert Serial Number: 142935828eb8801e00000000031c

dwFlags = CA_VERIFY_FLAGS_CONSOLE_TRACE (0x20000000)
dwFlags = CA_VERIFY_FLAGS_DUMP_CHAIN (0x40000000)
ChainFlags = CERT_CHAIN_REVOCATION_CHECK_CHAIN_EXCLUDE_ROOT (0x40000000)
HCCE_LOCAL_MACHINE
CERT_CHAIN_POLICY_BASE
-------- CERT_CHAIN_CONTEXT --------
ChainContext.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
ChainContext.dwRevocationFreshnessTime: 0s

SimpleChain.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
SimpleChain.dwRevocationFreshnessTime: 0s

CertContext[0][0]: dwInfoStatus=102 dwErrorStatus=0
  Issuer: CN=cc-CC-CA1-CA, DC=cc, DC=com
  NotBefore: 19/10/2015 22:03
  NotAfter: 18/10/2017 22:03
  Subject: CN=OcspTest
  Serial: 142935828eb8801e00000000031c
  Template: WebServer
  Cert: 5125686669212b21e2e1f6e310ca632512175b72
  Element.dwInfoStatus = CERT_TRUST_HAS_KEY_MATCH_ISSUER (0x2)
  Element.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
    CRL (null):
    Issuer: CN=OcspSigner
    ThisUpdate: 07/12/2015 12:11
    NextUpdate: 08/12/2015 12:11
    CRL: 1ebb20453d4616489079e3722d43e6fd6f83e0d6
  Application[0] = 1.3.6.1.5.5.7.3.1 Server Authentication

CertContext[0][1]: dwInfoStatus=10c dwErrorStatus=0
  Issuer: CN=cc-CC-CA1-CA, DC=cc, DC=com
  NotBefore: 25/04/2014 12:45
  NotAfter: 25/04/2019 12:55
  Subject: CN=cc-CC-CA1-CA, DC=cc, DC=com
  Serial: 2cc3198a5aa92a894ded62939e534a3f
  Cert: ca06b4e07dd7e90f1eebc30c14631f5095f9aecb
  Element.dwInfoStatus = CERT_TRUST_HAS_NAME_MATCH_ISSUER (0x4)
  Element.dwInfoStatus = CERT_TRUST_IS_SELF_SIGNED (0x8)
  Element.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)

Exclude leaf cert:
  Chain: 0e2f02f19852630a1c5f90f692360a57434cab0a
Full chain:
  Chain: 87a2b6cb3c3ec73bcc720dfb6941cf231014744e
----------------------------------
Verified Issuance Policies: None
Verified Application Policies:
    1.3.6.1.5.5.7.3.1 Server Authentication
Leaf certificate revocation check passed
CertUtil: -verify command completed successfully.
```

The output for the OCSP Server when receives a request from the *"certutil"* command is shown in the next picture.

```
--------------OCSP Request Start---------------
Version: 0
RequesterName: NULL (is not present in the request)
# of Request Entries: 1
RequestEntry: 0
AlgorithmId is: 1.3.14.3.2.26
Parameters:
05 00
IssuerNameHash:
7a 82 90 13 87 8b 26 0f 80 d5 49 06 ad e4 9c e5
21 75 d6 ab
IssuerKeyHash:
6e 54 95 75 78 7e 3a c7 76 c8 d4 63 0a 84 8c 56
3e 51 9a d9
SerialNumber:
1c 03 00 00 00 00 1e 80 b8 8e 82 35 29 14
# of RequestEntry Extensions: 0

# of Extensions: 0
--------------OCSP Request End---------------
48 54 54 50 2f 31 2e 31 20 32 30 30 20 0d 0a 0d
0a 30 82 04 98 0a 01 00 a0 82 04 91 30 82 04 8d
```

```
89 dd bc 99 f1 17 cc fe a5 cd bb 48 fe a1 ee 04
00 64 ab 81 30 54 df cb 64 37 37 8e 24 86 8e eb
50 c6 fb 6c 4c d2 08 ff e3 24 2a cb 35
OCSPServerResponse::Returning 1
```

The request is validated because the OCSP server has loaded a certificate with the OCSP signing purpose.

On the other hand, if the OCSP server is started with the ServerAuthSigner.pfx certificate, the OCSP server cannot response the OCSP request sent by the certutil.exe tool.

The certutil.exe output shows a bad response, Due to the OCSP request cannot detect the server. The OCSP server without the OCSP Signing Purpose is similar to an offline server as it can be appreciated in the next picture.

```
Full chain:
  Chain: 2165460626ca4f03948acdf8806ad546d5acedff
  Issuer: CN=cc-CC-CA1-CA, DC=cc, DC=com
  NotBefore: 19/10/2015 22:03
  NotAfter: 18/10/2017 22:03
  Subject: CN=OcspTest
  Serial: 142935828eb8801e000000000031c
  Template: WebServer
  Cert: 5125686669212b21e2e1f6e310ca632512175b72
The revocation function was unable to check revocation because the revocation server was offline. 0x80092013 (-2146885613 CRYPT_E_REVOCATION_OFFLINE)
------------------------------------
Revocation check skipped -- server offline

ERROR: Verifying leaf certificate revocation status returned The revocation function was unable to check revocation because the revocation server was offline. 0x8009201
3 (-2146885613 CRYPT_E_REVOCATION_OFFLINE)
CertUtil: The revocation function was unable to check revocation because the revocation server was offline.
```

24.3.4.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.3.5. Test 5

24.3.5.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine shall be installed the *"Cerberus FTP Server enterprise"* and *"Wireshark"* applications.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

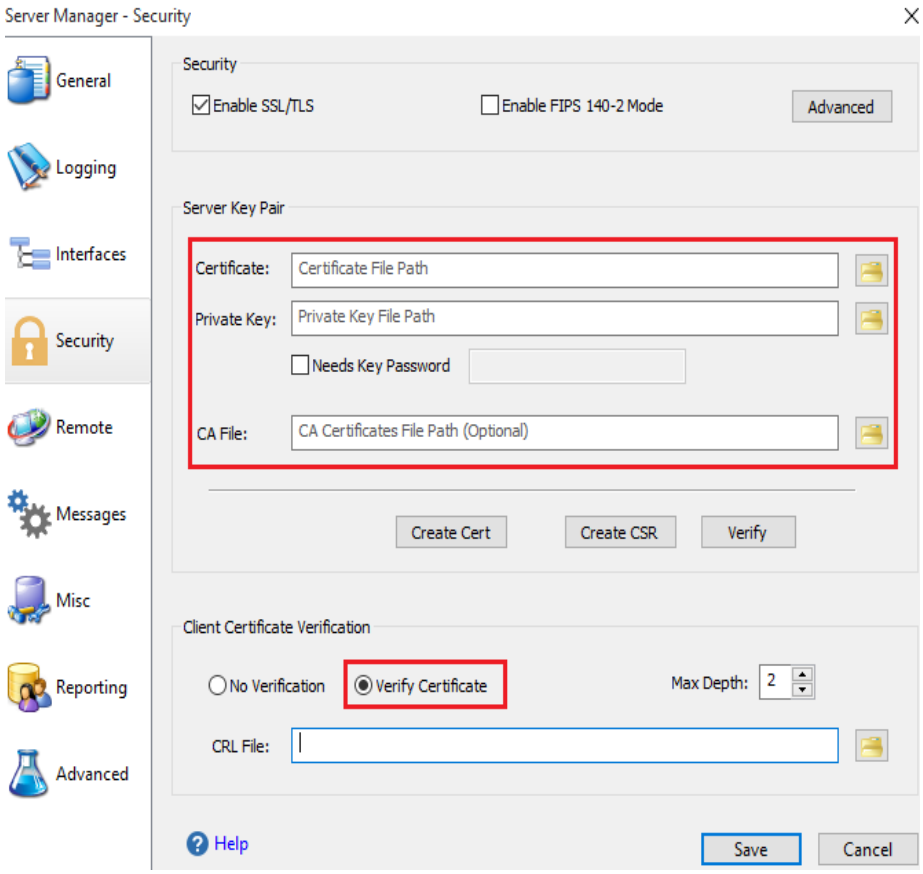
The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

24.3.5.2. Procedure

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

12. Add the *"RootCA.pkcs"* and *"c1.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.
13. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chainTest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2"*.

- In Client Certificate Verification select "*Verify Certificate*".
- Press *"save"*.
- Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
- Press *"save"*.



14. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
15. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
16. In the Server and Client Machine, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.

17. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

24.3.5.3. Results

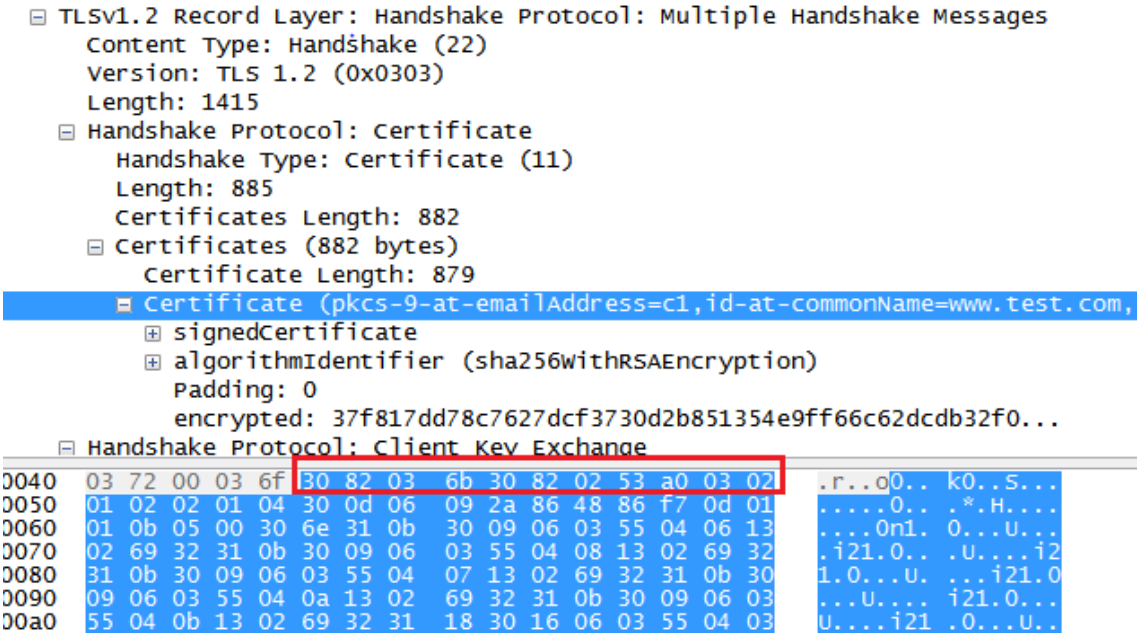
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

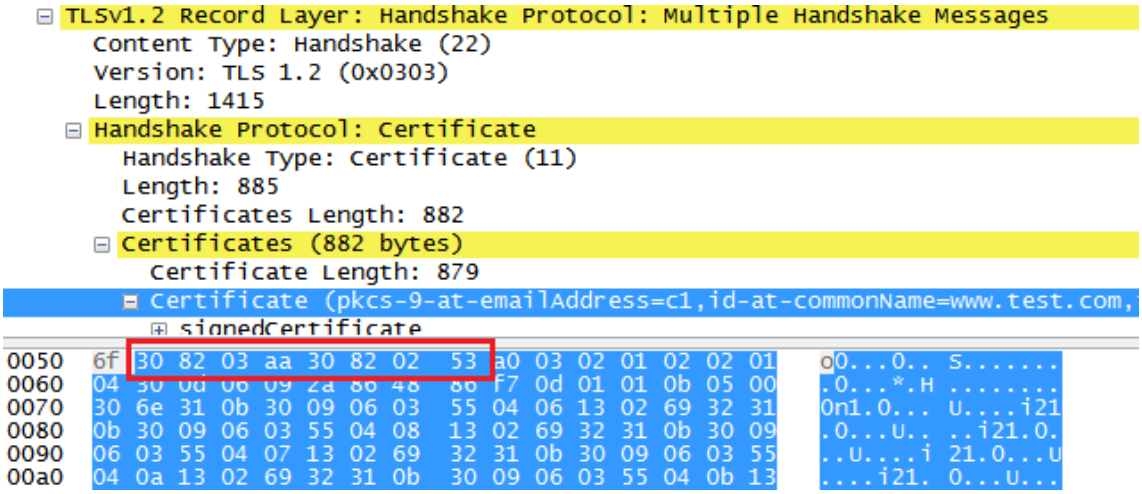
Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def modify_first_bytes(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 5
        if packet[indexPacket] == '\x0B':
            indexPacket += 13
            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

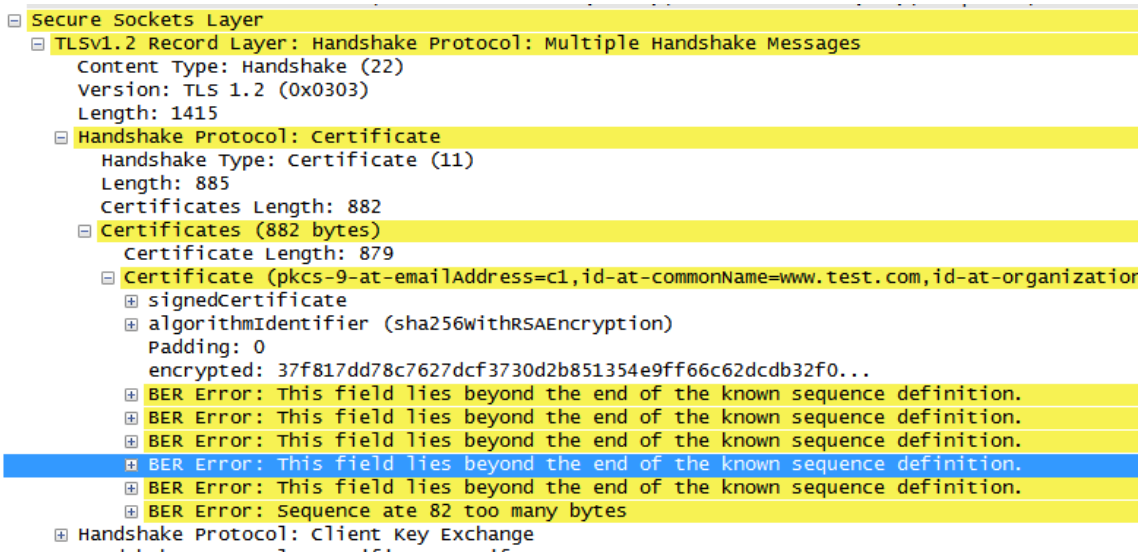
The Client Machine sends the *"Certificate"* message that contains the client certificate. In the following picture can be appreciated the first bytes of the client certificate.



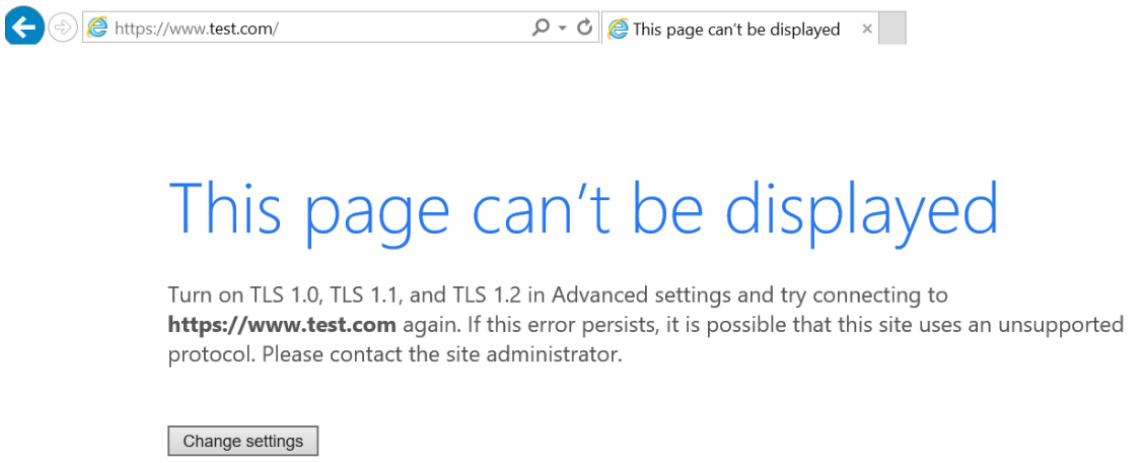
The MITM Machine overwrites the fourth byte *"6b"* by *"aa"* in the client certificate, as it can be appreciated in the next picture.



After the Server Machine receives the modified packet, it sends a *"FIN ACK"* to the Client Machine, due to the server detects and error while parse the certificate. Therefore the connection cannot be established, as it can be appreciated in the following picture.



In addition, the browser shows to the user the following screen.



24.3.5.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.3.6. Test 6

24.3.6.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine shall be installed the *"Cerberus FTP Server enterprise"* and *"Wireshark"* applications.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

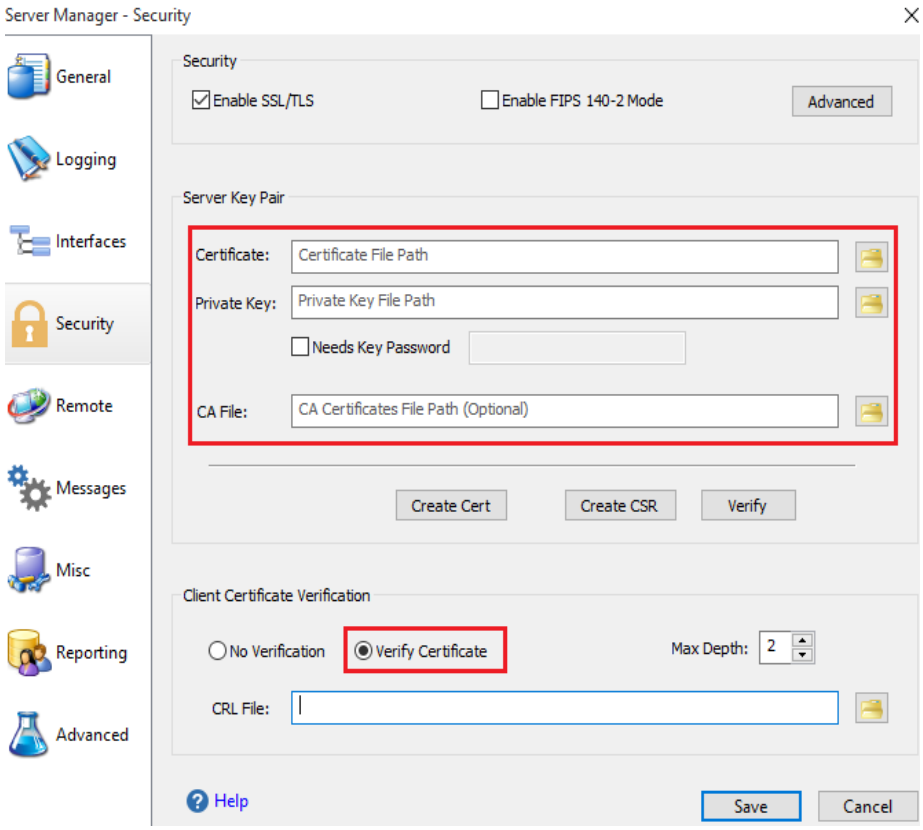
24.3.6.2. Procedure

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

1. Add the *"RootCA.pkcs"* and *"c1.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.

- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
- Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.

2. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt".* The *"chainTest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press *"save"*.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press *"save"*.



3. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
4. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
5. In the Server and Client Machine, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
6. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

24.3.6.3. Results

The test has been performed in the following platforms:

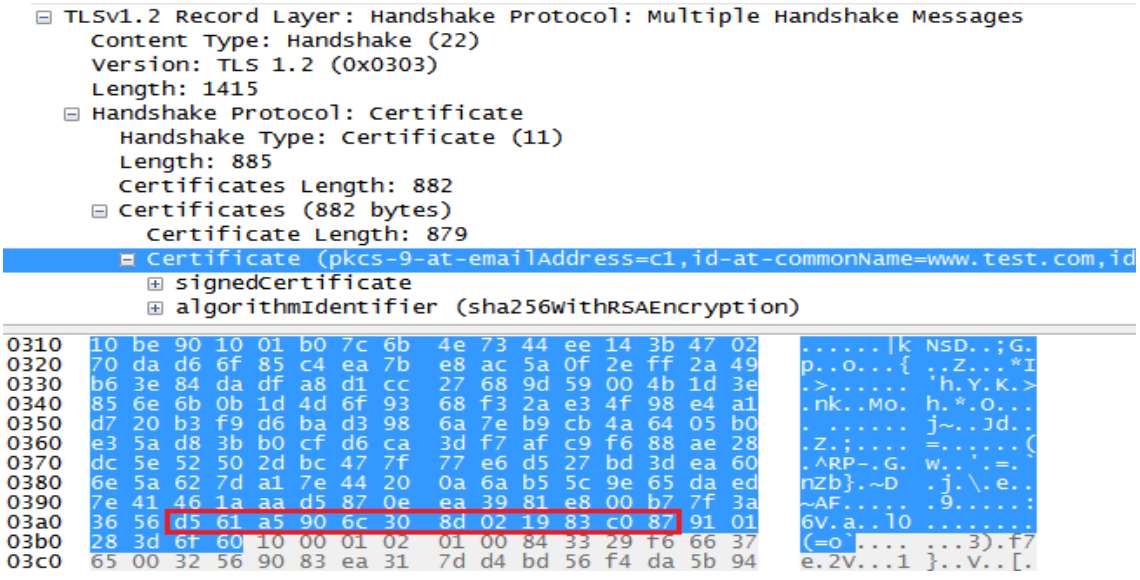
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.

- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

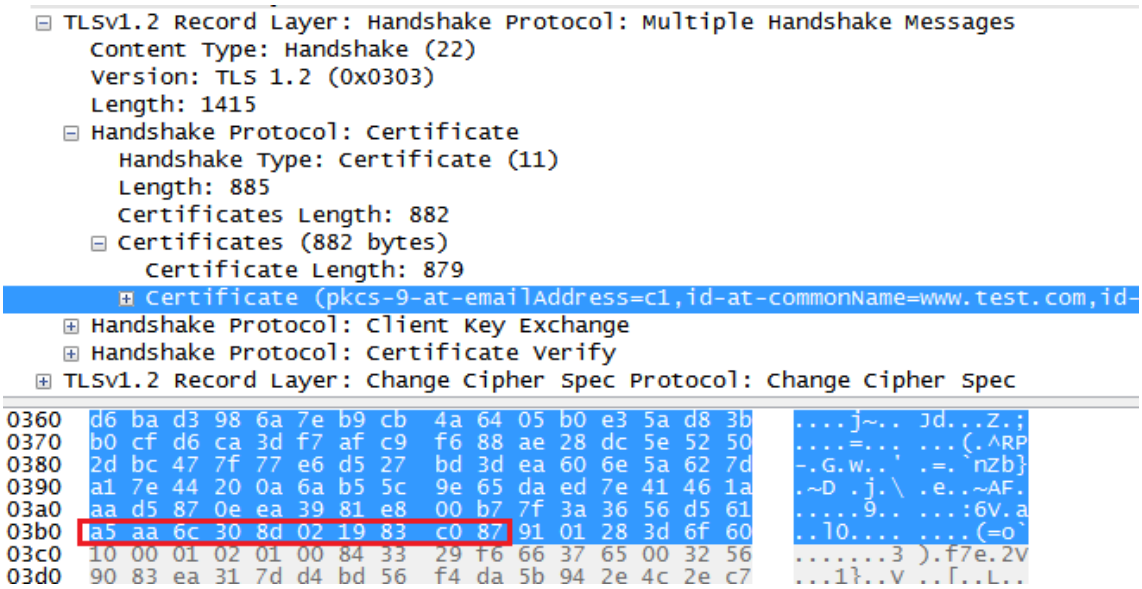
Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the certificate can be appreciated in the next picture.

```
def modify_last_byte_certificate(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 5
        if packet[indexPacket] == '\x0B':
            hexCertificateLen = packet[13:15].encode("hex")
            decCertificateLen = int(hexCertificateLen, 16)
            indexPacket += decCertificateLen - 5

            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

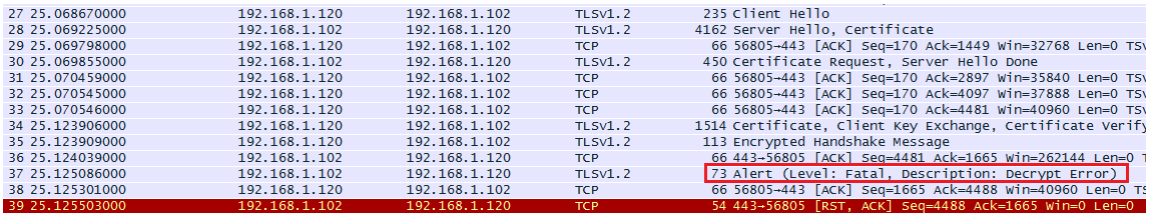
The Client Machine sends the “Certificate” message that contains the client certificate. In the following picture can be appreciated the last bytes of the client certificate.



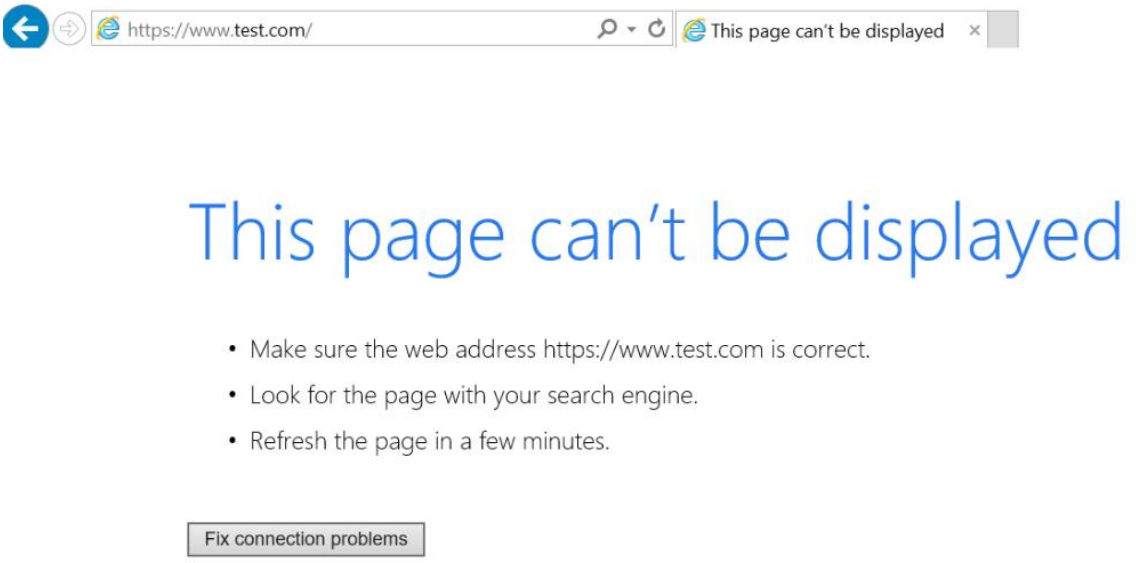
The MITM Machine overwrites one of the last bytes (e.g. *"90"* by *"aa"*) in the client certificate, as it can be appreciated in the next picture.



After the Server Machine receives the modified packet, it sends an *"Alert"* message and *"RST, ACK"* to the Client Machine, due to a decrypt error. Therefore the connection cannot be established, as it can be appreciated in the following picture.



In addition, the browser shows to the user the following screen.



24.3.6.4. Results

24.3.6.5. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.3.7. Test 7

24.3.7.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine shall be installed the *"Cerberus FTP Server enterprise"* and *"Wireshark"* applications.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

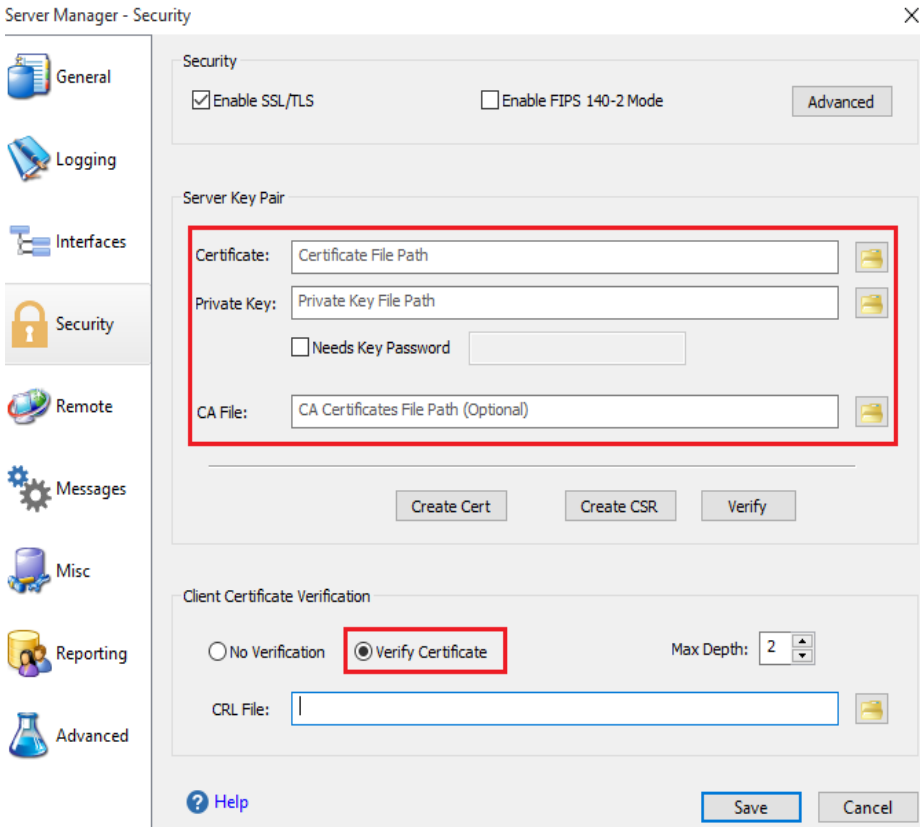
The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

24.3.7.2. Procedure

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

1. Add the *"RootCA.pkcs"* and *"c1.pkcs"* certificates in the Client Machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.
2. Load the server certificate and the chain file in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chainTest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2"*.

- In Client Certificate Verification select "*Verify Certificate*".
- Press *"save"*.
- Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
- Press *"save"*.



3. In the MITM Machine open a terminal and type the followings commands:
   - *"cd Desktop/SSL_Proxy"*
   - *"chmod 777 run_mitm"*
   - *"./run_mitm"*
4. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
5. In the Server and Client Machine, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.

6. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

24.3.7.3. Results

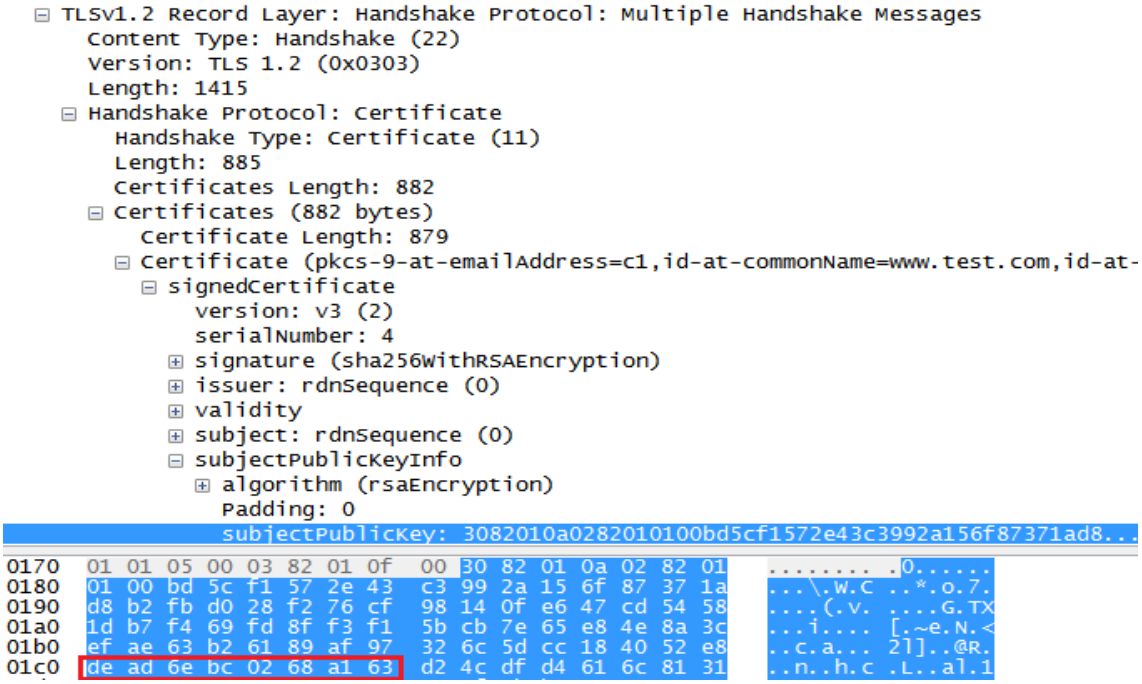
The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

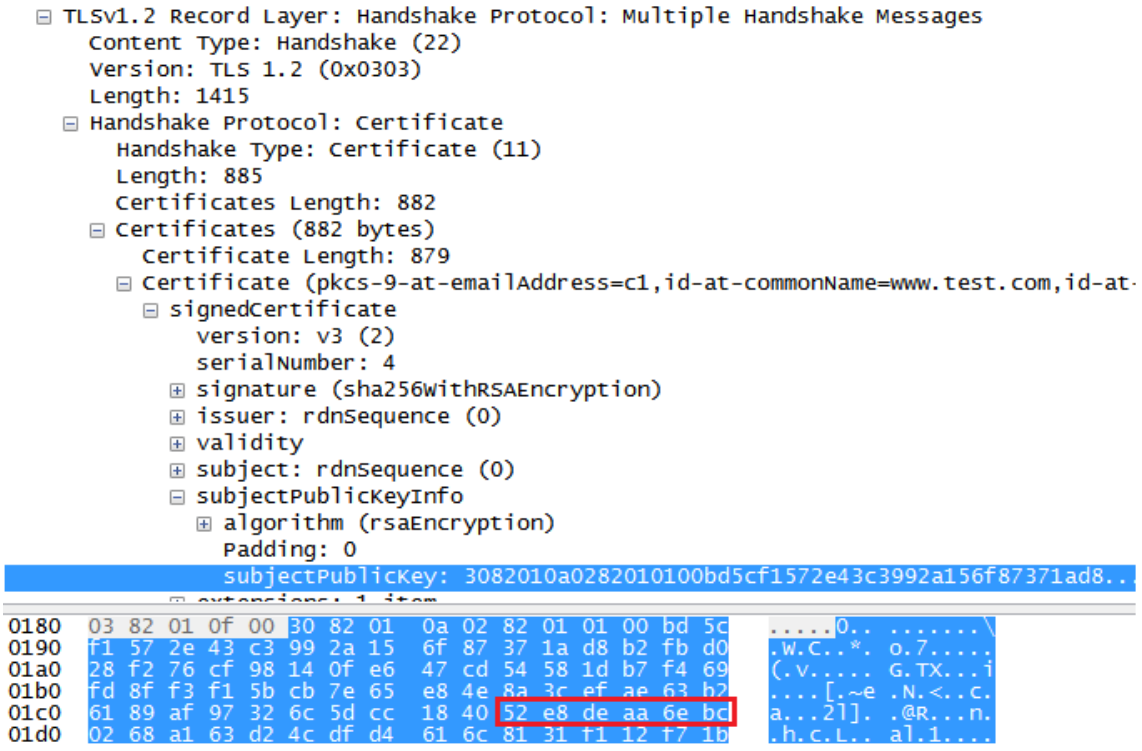
Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the certificate can be appreciated in the next picture.

```
def modify_public_key(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 5
        if packet[indexPacket] == '\x0B':
            indexPacket += 390

            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]

    return newPacket
```

The Client Machine sends the “Certificate” message that contains the client certificate. In the following picture can be appreciated a part of the public key into the client certificate.



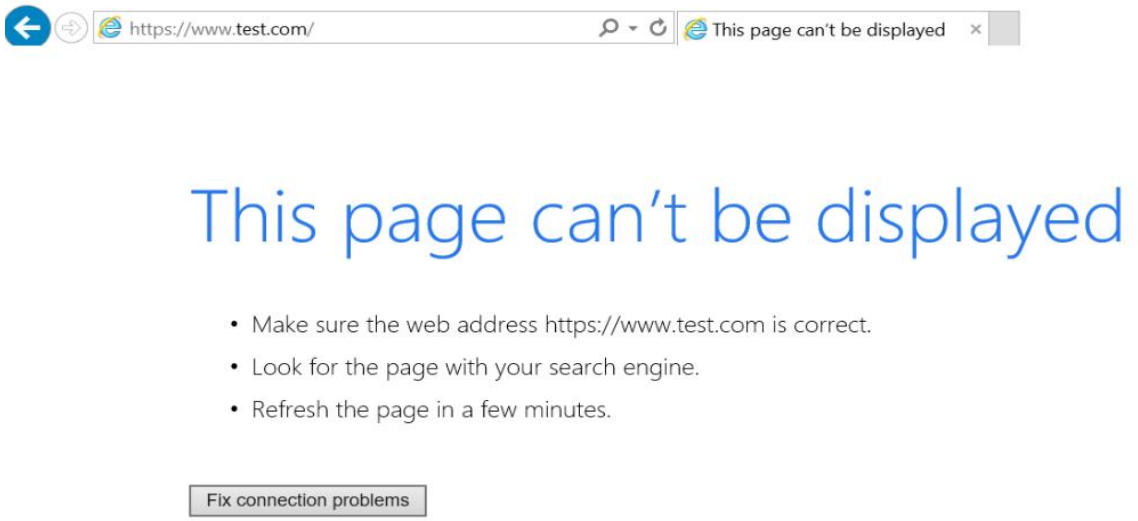
The MITM Machine overwrites the byte *"ad"* by *"aa"* in the client certificate, as it can be appreciated in the next picture.



After the Server Machine receives the modified packet, it sends an *"Alert"* message and *"RST, ACK"* packet to the Client Machine, due to a decrypt error. Therefore the connection cannot be established, as it can be appreciated in the following picture.

30	32.040991000	192.168.1.120	192.168.1.102	TLSv1.2	235 Client Hello
31	32.041457000	192.168.1.102	192.168.1.120	TLSv1.2	4162 Server Hello, Certificate
32	32.055302000	192.168.1.120	192.168.1.102	TCP	66 56845→443 [ACK] Seq=170 Ack=1449 Win=32768 Len=0 TSva
33	32.055303000	192.168.1.120	192.168.1.102	TCP	66 56845→443 [ACK] Seq=170 Ack=2897 Win=35840 Len=0 TSva
34	32.055304000	192.168.1.120	192.168.1.102	TCP	66 56845→443 [ACK] Seq=170 Ack=4097 Win=37888 Len=0 TSva
35	32.055359000	192.168.1.102	192.168.1.120	TLSv1.2	450 Certificate Request, Server Hello Done
36	32.063333000	192.168.1.120	192.168.1.102	TCP	66 56845→443 [ACK] Seq=170 Ack=4481 Win=40960 Len=0 TSva
37	32.079771000	192.168.1.120	192.168.1.102	TLSv1.2	1514 Certificate, Client Key Exchange, Certificate Verify,
38	32.079773000	192.168.1.120	192.168.1.102	TLSv1.2	113 Encrypted Handshake Message
39	32.079850000	192.168.1.102	192.168.1.120	TCP	66 443→56845 [ACK] Seq=4481 Ack=1665 Win=262144 Len=0 TS
40	32.081378000	192.168.1.102	192.168.1.120	TLSv1.2	73 Alert (Level: Fatal, Description: Decrypt Error)
41	32.081913000	192.168.1.102	192.168.1.120	TCP	54 443→56845 [RST, ACK] Seq=4488 Ack=1665 Win=0 Len=0

In addition, the browser shows to the user the following screen.



24.3.7.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

24.4. Final Verdict

Due to all test activities 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 this FIA_X509_EXT.1.1.

25. FIA_X509_EXT.1.2

25.1. Assurance activity

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 selfsigned 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.*

25.2. Documentation review activity

25.2.1. Findings

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

25.2.2. Verdict

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

25.3. Test Activity

25.3.1. Test 1

25.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)

- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine must be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

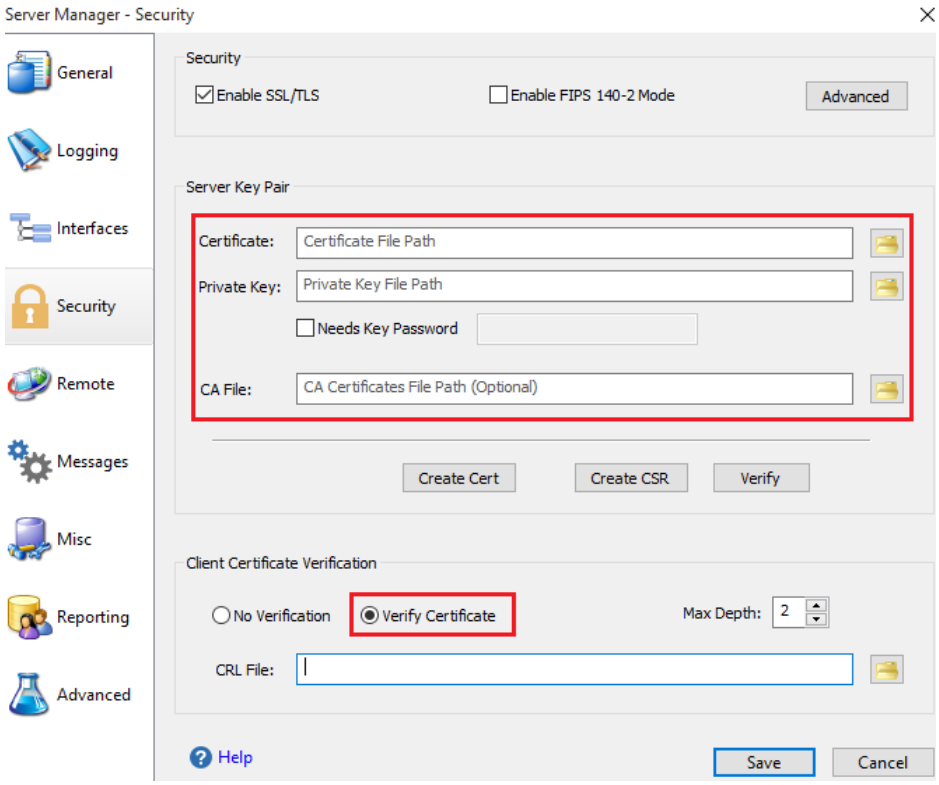
25.3.1.2. Procedure

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

1. Add the invalid certificates used for the Test 1 in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.

- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
- Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.

2. Load the server certificate and the chain in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chaintest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2"*.
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press *"save"*.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press *"save"*.



3. In the MITM Machine open a terminal and type the followings commands:
    a. *"cd Desktop/SSL_Proxy"*
    b. *"chmod 777 run_mitm"*
    c. *"./run_mitm"*
4. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
5. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
6. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

25.3.1.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.

- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def delete_basic_constraints(self, data, offset):
    packet = data
    newPacket = data
    newPacket5 = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 58

        if packet[indexPacket] == '\x16':
            indexPacket += 5

            if packet[indexPacket] == '\x0B':
                constrain = '\x30' + '\x0C' + '\x06' + '\x03' + '\x55'
                result = data.find(constrain)

                if result > 0 :
                    len1 = int(packet[indexPacket - 2:indexPacket].encode("hex"),16)
                    len1 -= 14

                    newPacket = packet[:indexPacket - 2] + struct.pack(">h", len1) + packet[indexPacket:]

                    len2 = int(newPacket[indexPacket + 1:indexPacket + 4].encode("hex"),16)
                    len2 -= 14

                    newPacket2 = newPacket[:(indexPacket + 1)] + '\x00' + struct.pack(">h", len2) + newPacket[(indexPacket + 4):]

                    len3 = int(newPacket2[indexPacket + 4:indexPacket + 7].encode("hex"),16)
                    len3 -= 14

                    newPacket3 = newPacket2[:indexPacket + 4] + '\x00' + struct.pack(">h", len3) + newPacket2[indexPacket + 7:]

                    len4 = int(newPacket3[indexPacket + 2994:indexPacket + 2997].encode("hex"),16)
                    len4 -= 14

                    newPacket4 = newPacket3[:indexPacket + 2994] + '\x00' + struct.pack(">h", len4) + newPacket3[indexPacket + 2997:]

                    newPacket5 = newPacket4[:indexPacket + 3566] + newPacket4[indexPacket + 3580:]  #basic constraints erased

    return newPacket5
```

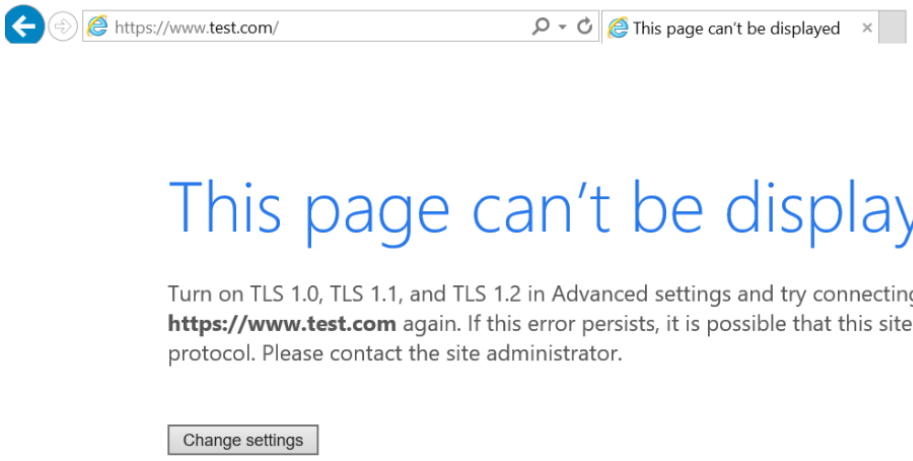
When the Client Machine sends the *"Client Hello"*, the Server Machine responses with a *"Server Hello"* and "Certificate" messages. The "Certificate" message contains all certificates of the certification path, included the *"RootCA"*. The *"RootCA"* has the extension *"basicConstratins"* as it can be appreciated in the following picture.

```
⊟ Certificates (4026 bytes)
    Certificate Length: 879
  ⊞ Certificate (pkcs-9-at-emailAddress=s1,id-at-commonName=www.test.com,id-at-organizationalUnitName=s1
    Certificate Length: 1054
  ⊞ Certificate (pkcs-9-at-emailAddress=i2,id-at-commonName=intermediateCA2,id-at-organizationalUnitName
    Certificate Length: 1045
  ⊞ Certificate (pkcs-9-at-emailAddress=i1,id-at-commonName=intermediateCA1,id-at-organizationalUnitName
    Certificate Length: 1036
  ⊟ Certificate (pkcs-9-at-emailAddress=ca,id-at-commonName=RootCA,id-at-organizationalUnitName=ca,id-at
    ⊟ signedCertificate
        version: v3 (2)
        serialNumber: 1
      ⊞ signature (sha256WithRSAEncryption)
      ⊞ issuer: rdnSequence (0)
      ⊞ validity
      ⊞ subject: rdnSequence (0)
      ⊞ subjectPublicKeyInfo
      ⊟ extensions: 3 items
        ⊞ Extension (id-ce-basicConstraints)
        ⊞ Extension (id-ce-subjectKeyIdentifier)
        ⊞ Extension (id-ce-authorityKeyIdentifier)
```

The MITM Machine erases the extension *"basicConstrains"* and recalculates the certificate length as it can be appreciated in the next picture.

```
Certificates (4012 bytes)
  Certificate Length: 879
  Certificate (pkcs-9-at-emailAddress=s1,id-at-commonName=www.test.com,id-at-organizationalUnitNa
  Certificate Length: 1054
  Certificate (pkcs-9-at-emailAddress=i2,id-at-commonName=intermediateCA2,id-at-organizationalUni
  Certificate Length: 1045
  Certificate (pkcs-9-at-emailAddress=i1,id-at-commonName=intermediateCA1,id-at-organizationalUni
  Certificate Length: 1022
  Certificate (pkcs-9-at-emailAddress=ca,id-at-commonName=RootCA,id-at-organizationalUnitName=ca,
    signedCertificate
      version: v3 (2)
      serialNumber: 1
    signature (sha256WithRSAEncryption)
    issuer: rdnSequence (0)
    validity
    subject: rdnSequence (0)
    subjectPublicKeyInfo
    extensions: 3 items
      Extension (id-ce-subjectKeyIdentifier)
      Extension (id-ce-authorityKeyIdentifier)
```

After the Client Machine receives the modified packet, it sends a *"FIN ACK"* to the Server Machine. Therefore the handshake process is not performed. In addition the browser shows to the user the following screen.



25.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

25.3.2. Test 2

25.3.2.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120
- MITM Machine, IP = 192.168.1.100

In the Server Machine must be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

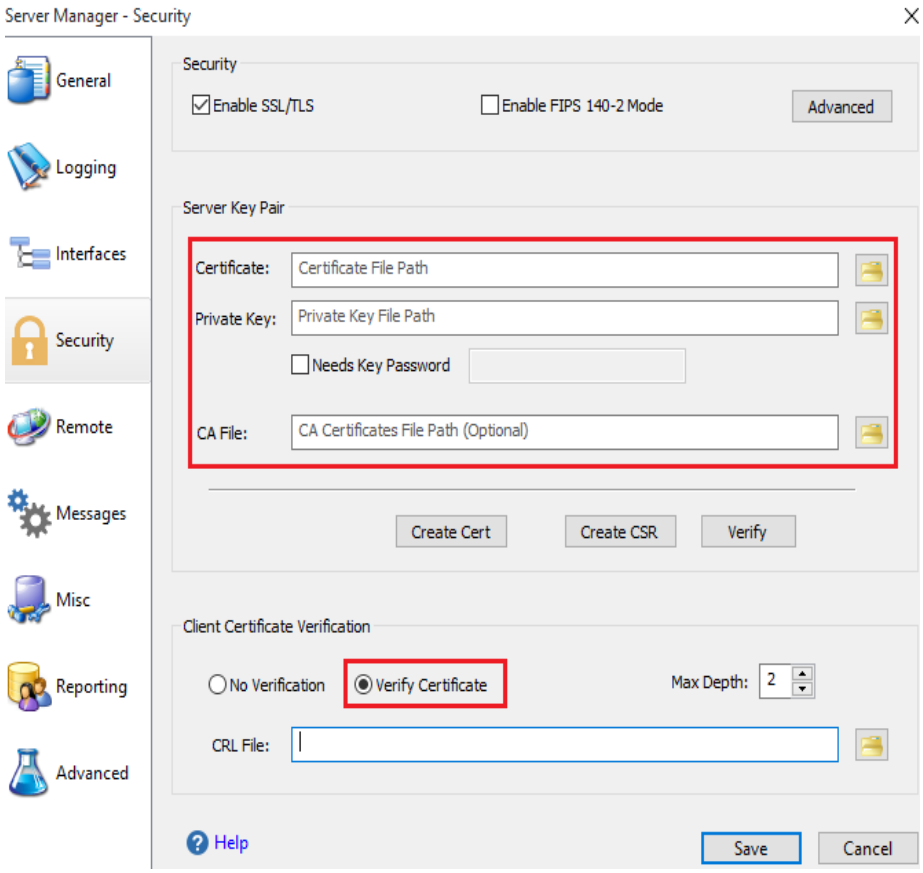
25.3.2.2. Procedure

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

1. Add the invalid certificates used for the Test 1 in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.

- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
- Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.

2. Load the server certificate and the chain in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chaintest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press *"save"*.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press *"save"*.



3. In the MITM Machine open a terminal and type the followings commands:
   - *"cd Desktop/SSL_Proxy"*
   - *"chmod 777 run_mitm"*
   - *"./run_mitm"*
4. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
5. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.
6. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

25.3.2.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.

- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured in the *"Wireshark"* application, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the packet can be appreciated in the next picture.

```
def flag_not_set(self, data, offset):
    packet = data
    newPacket = data
    newPacket5 = data
    indexPacket = offset

    if packet[0] == '\x16':
        indexPacket += 58

        if packet[indexPacket] == '\x16':
            indexPacket += 5
            if packet[indexPacket] == '\x0B':
                constrain = '\x30' + '\x0C' + '\x06' + '\x03' + '\x55'
                result = data.find(constrain)

                if result > 0 :
                    newPacket5 = packet[:indexPacket + 3579] + '\x00' + packet[indexPacket + 3580:]  # CA: TRUE(FF) -> 00

    return newPacket5
```

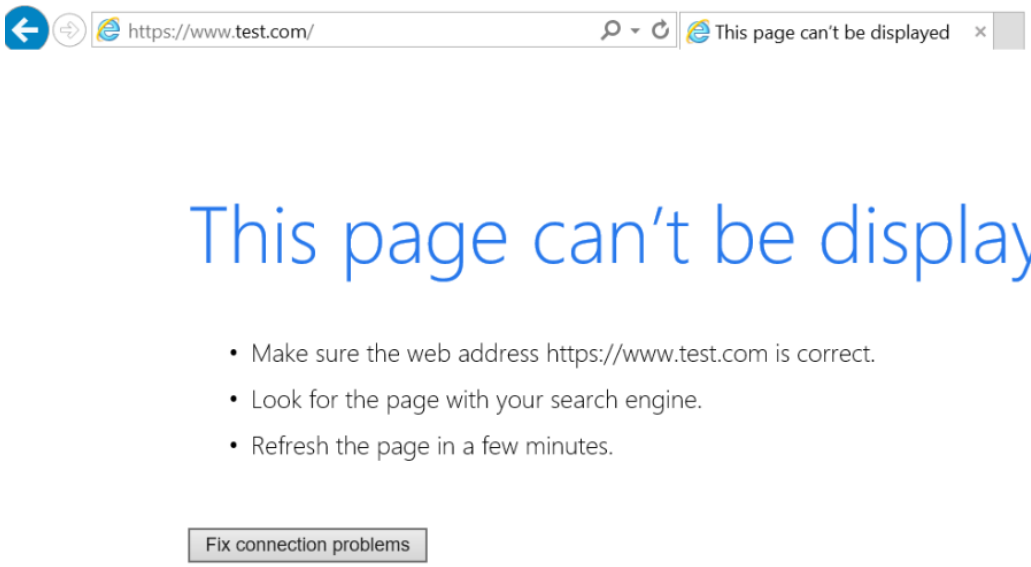
When the Client Machine sends the *"Client Hello"*, the Server Machine responses with a *"Server Hello"* and *"Certificate"* messages. The *"Certificate"* message contains all certificates of the certification path, included the *"RootCA"*. The *"RootCA"* has the flag in the *"basicConstraints"* set with the *"TRUE"* value, as it can be appreciated in the following picture.

```
⊟ Certificates (4026 bytes)
    Certificate Length: 879
  ⊞ Certificate (pkcs-9-at-emailAddress=s1,id-at-commonName=www.test.com,id-at-organizationalUnitName=
    Certificate Length: 1054
  ⊞ Certificate (pkcs-9-at-emailAddress=i2,id-at-commonName=intermediateCA2,id-at-organizationalUnitNa
    Certificate Length: 1045
  ⊞ Certificate (pkcs-9-at-emailAddress=i1,id-at-commonName=intermediateCA1,id-at-organizationalUnitNa
    Certificate Length: 1036
  ⊟ Certificate (pkcs-9-at-emailAddress=ca,id-at-commonName=RootCA,id-at-organizationalUnitName=ca,id-
    ⊟ signedCertificate
        version: v3 (2)
        serialNumber: 1
      ⊞ signature (sha256WithRSAEncryption)
      ⊞ issuer: rdnSequence (0)
      ⊞ validity
      ⊞ subject: rdnSequence (0)
      ⊞ subjectPublicKeyInfo
      ⊟ extensions: 3 items
        ⊟ Extension (id-ce-basicConstraints)
            Extension Id: 2.5.29.19 (id-ce-basicConstraints)
          ⊟ BasicConstraintsSyntax
              cA: True
```

The MITM Machine changes the flag in the *"basicConstraints"* from *"FF (True)"* to *"00 (False)"* as it can be appreciated in the next picture.

```
Certificates (4026 bytes)
  Certificate Length: 879
  Certificate (pkcs-9-at-emailAddress=s1,id-at-commonName=www.test.com,id-at-organizationalUnitName=s1,
  Certificate Length: 1054
  Certificate (pkcs-9-at-emailAddress=i2,id-at-commonName=intermediateCA2,id-at-organizationalUnitName=
  Certificate Length: 1045
  Certificate (pkcs-9-at-emailAddress=i1,id-at-commonName=intermediateCA1,id-at-organizationalUnitName=
  Certificate Length: 1036
  Certificate (pkcs-9-at-emailAddress=ca,id-at-commonName=RootCA,id-at-organizationalUnitName=ca,id-at
    signedCertificate
      version: v3 (2)
      serialNumber: 1
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
      validity
      subject: rdnSequence (0)
      subjectPublicKeyInfo
      extensions: 3 items
        Extension (id-ce-basicConstraints)
          Extension Id: 2.5.29.19 (id-ce-basicConstraints)
          BasicConstraintsSyntax
            cA: False
```

After the Client Machine receives the modified packet, it sends a *"FIN ACK"* to the Server Machine. Therefore the handshake process is not performed, in addition the browser shows to the user the following screen:



25.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

25.3.3. Test 3

25.3.3.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below, have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = IntermediateCA1, (i1)
- CN = IntermediateCA2, (i2)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))
- File that contains the RootCA, IntermediateCA1 and IntermediateCA2 (chainTest1b).

The certificates listed above form a valid certification path *"RootCA -> IntermediateCA1 -> IntermediateCA2 -> (Server, Client)"*. All certificates and the chain file should be exported in *"pem"*, *"crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.102
- Client Machine, IP = 192.168.1.120

In the Server Machine must be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

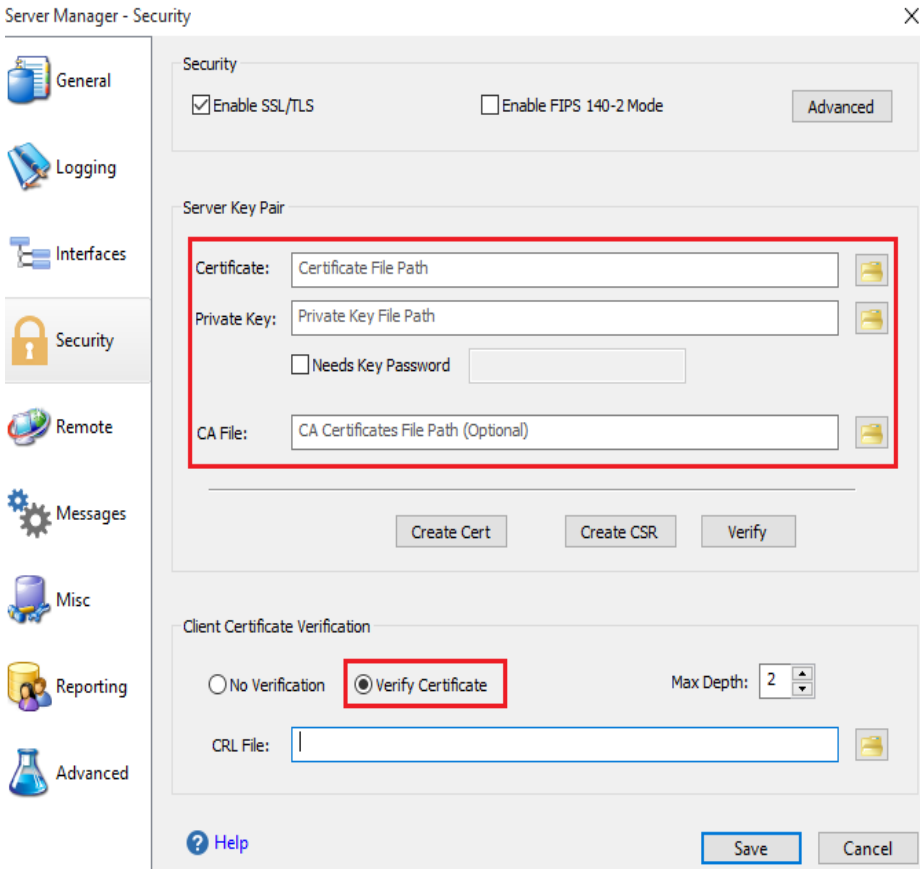
25.3.3.2. Procedure

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

1. Add the invalid certificates used for the Test 1 in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.

- On the *"File"* menu, click *"Add/Remove Snap-in"*.
- In the Add standalone Snap-in dialog box, select *"Certificates"*.
- Press *"Add"*.
- Press *"OK"*.
- In the Certificates Snap-in dialog box, select *"My user account"* and click next.
- Press *"OK"*.
- Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
- Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
- Select *"Personal"*.
- Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.

2. Load the server certificate and the chain in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
   - Launch the application *"Cerberus FTP server enterprise"*.
   - Open Configure tag and click in the Security option.
   - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*. The *"chaintest1b.pem"* contains the *"RootCA, IntermediateCA1 and IntermediateCA2".*
   - In Client Certificate Verification select "*Verify Certificate*".
   - Press *"save"*.
   - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
   - Press *"save"*.



3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
4. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
5. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

25.3.3.3. Results

The test has been performed in the following platforms:

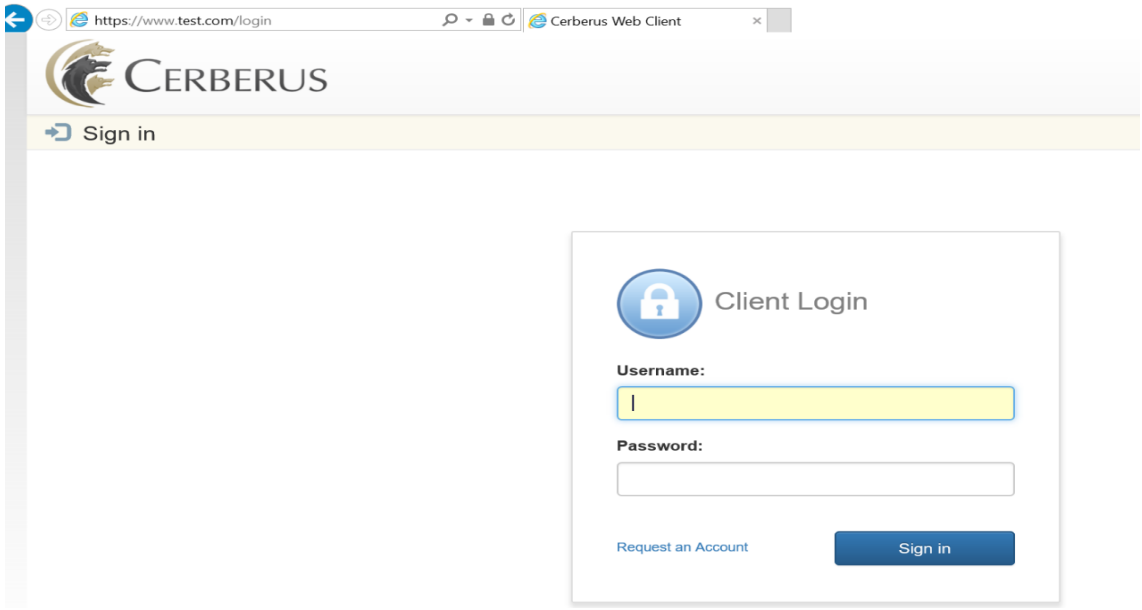
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured in the *"Wireshark"* application, it can be appreciated the handshaking process. The following picture shows all steps involved in this process.

36	3.941567000	192.168.1.120	192.168.1.102	TLSv1.2	223 Client Hello
37	3.943072000	192.168.1.102	192.168.1.120	TLSv1.2	1514 Server Hello
38	3.943073000	192.168.1.102	192.168.1.120	TCP	1514 [TCP segment of a reassembled PDU]
39	3.943074000	192.168.1.102	192.168.1.120	TLSv1.2	1230 Certificate
40	3.943144000	192.168.1.120	192.168.1.102	TCP	54 3783→443 [ACK] Seq=170 Ack=4097 Win=262144 Len=0
41	3.944562000	192.168.1.102	192.168.1.120	TLSv1.2	438 Certificate Request, Server Hello Done
42	3.944600000	192.168.1.120	192.168.1.102	TCP	54 3783→443 [ACK] Seq=170 Ack=4481 Win=261632 Len=0
43	3.958160000	192.168.1.120	192.168.1.102	TLSv1.2	1549 Certificate, Client Key Exchange, Certificate Verify,
44	3.959756000	192.168.1.102	192.168.1.120	TCP	60 443→3783 [ACK] Seq=4481 Ack=1665 Win=262144 Len=0
45	3.979561000	192.168.1.102	192.168.1.120	TLSv1.2	1216 New Session Ticket, Change Cipher Spec, Encrypted Han
46	3.979609000	192.168.1.120	192.168.1.102	TCP	54 3783→443 [ACK] Seq=1665 Ack=5643 Win=262144 Len=0
47	3.981100000	192.168.1.120	192.168.1.102	TLSv1.2	363 Application Data

Once the secure channel has been established, the Client and Server machines exchange application data packets as it can be appreciated in the *"packet 47"*.

When the connection is successful, the browser shows the login page of the "*Cerberus FTP server enterprise*", the following picture illustrates the browser response.



25.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

25.4. Final Verdict

Due to all test activities 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 this FIA_X509_EXT.1.2.

26. FIA_X509_EXT.2.1

26.1. Assurance activity

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.

26.2. Documentation review activity

26.2.1. Findings

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

26.2.2. Verdict

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

26.3. Test Activity

26.3.1. Test 1

26.3.1.1. Setup

The following certificates shall be used to perform the assurance activities listed in the Protection Profile. The certificates listed below have been created using the *"X Certificate and Key management"* tool.

- CN = RootCA, (ca)
- CN = www.test.com, (Server (s1))
- CN = www.test.com, (Client (c1))

All certificates should be exported in *"pem", "crt"* and *"pkcs12"* formats.

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows 10 Pro x86)
- Client Machine (Platforms listed in the ST)

These two machines are in the same network with the following configuration:.

- Server Machine, IP = 192.168.1.102

- Client Machine, IP = 192.168.1.120

In the Server Machine must be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

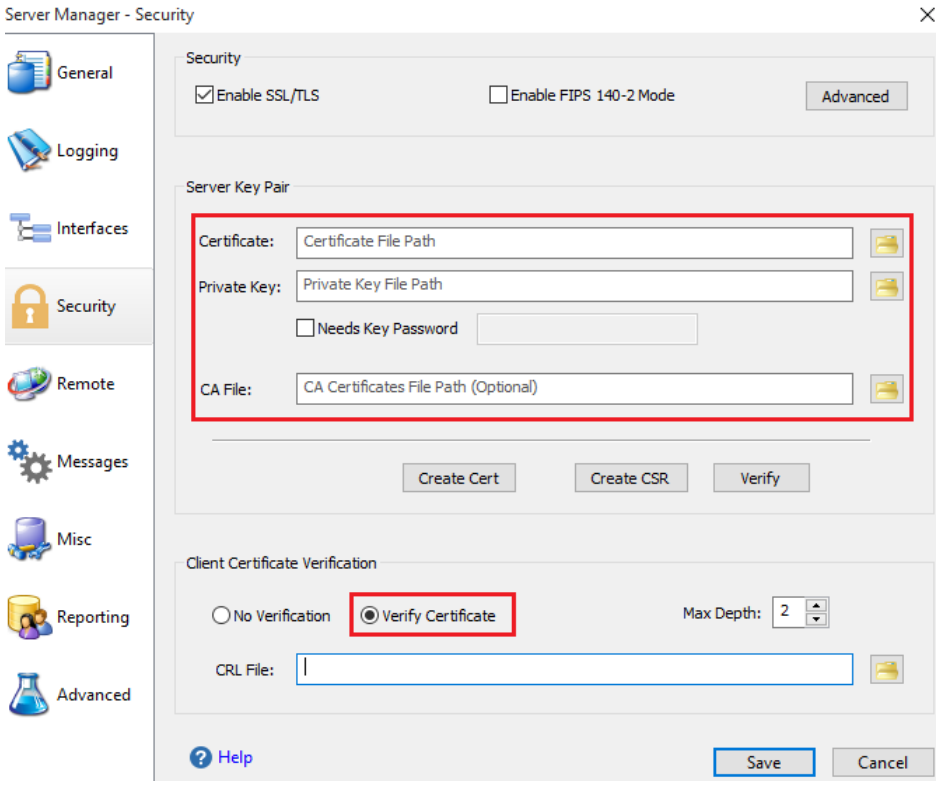
The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

26.3.1.2. Procedure

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

1. Add the invalid certificates used for the Test 1 in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pkcs"* is stored and type the key of the *"pkcs"*.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"c1.pkcs"* is stored and type the key of the *"pkcs"*.
2. Load the server certificate and the chain in the application *"Cerberus FTP server enterprise"*, the following steps shall be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"s1.crt"*, the *"s1.pem" and "ca.crt"*.
    - In Client Certificate Verification select "*Verify Certificate*".

- Press *"save"*.
- Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
- Press *"save"*.



3. In the Client Machine add the next line *"192.168.1.102 www.test.com"* to the hosts file located in the folder *"C:\Windows\System32\drivers\etc"* and reboot the client machine.
4. In the Server and Client Machines, open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
5. In the Client Machine, open the browser and attempt to navigate to the test web *"https://www.test.com"*.

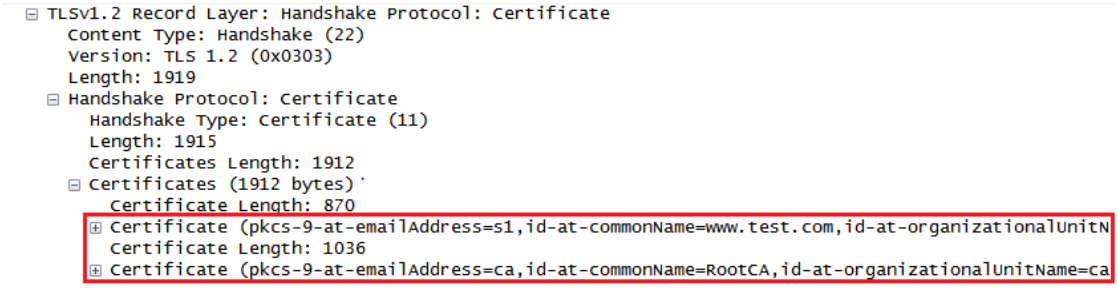
26.3.1.3. Results

The test has been performed in the following platforms:

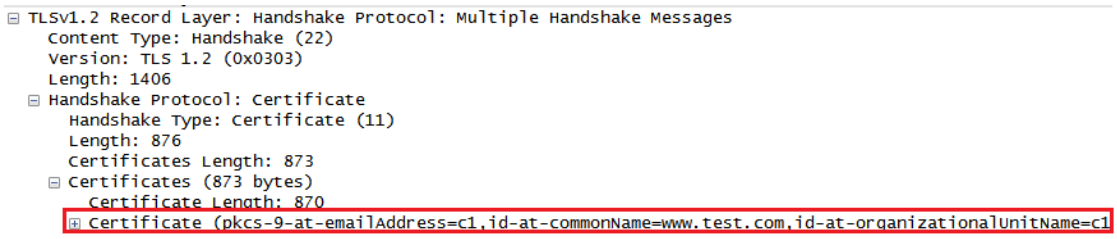
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.

- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

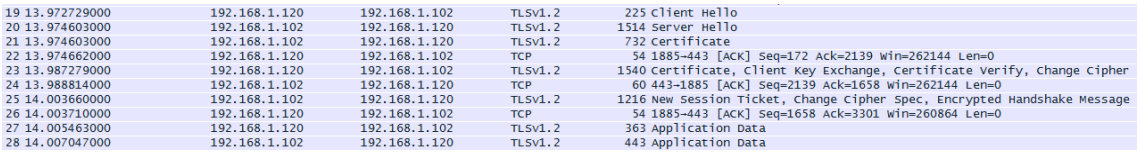
Analyzing the packets captured by *"Wireshark"*, it can be appreciated in the certificate provided for the server in the *"Server Hello"* message together with the Certification Authority. The server certificate is used by the Client Machine to authenticate the Server Machine. The following picture shows the *"Certificate"* message.



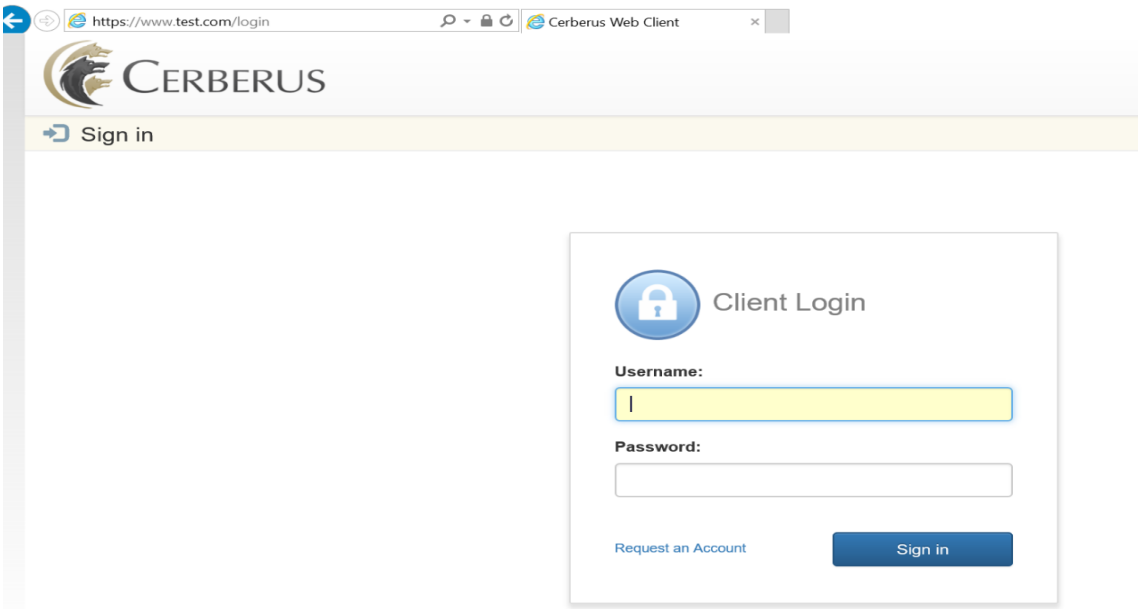
On the other hand, the Server Machine sends a *"Certificate Request"* message to the Client Machine asking for the client certificate, which will be used by the Server Machine to authenticate the Client Machine.



After the Server Machine verifies the client certificate, the connection is established using the certificates exchanged between the Client and Server Machines. In the following picture can be appreciated all steps in the handshaking process.



When the handshaking process is established correctly, the browser shows to the user the following screen:



26.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

26.4. Final Verdict

Due to all test activities 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 this FIA_X509_EXT.2.1.

27. FMT_MOF_EXT.1.1

27.1. Assurance activity

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.

27.2. Documentation review activity

27.2.1. Findings

The evaluator has reviewed the security target in order to identify all the management functions defined by the vendor.

#	Management Function	Administrator	User
1	configure minimum password length	√	
~~2~~	~~configure minimum number of special characters in password~~		
~~3~~	~~configure minimum number of numeric characters in password~~		
4	~~configure minimum number of uppercase characters in password~~		
~~5~~	~~configure minimum number of lowercase characters in password~~		
6	enable/disable screen lock	√	√
7	configure screen lock inactivity timeout	√	√
8	configure remote connection inactivity timeout	√	
9	~~enable/disable unauthenticated logon~~		
10	configure lockout policy for unsuccessful authentication attempts through [**selection**: *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[17]	√	
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 local audit storage capacity	√	
16	configure audit rules	√	
17	configure name/address of network time server	√	
18	enable/disable automatic software update	√	
19	configure Wi-Fi interface	√	
20	enable/disable Bluetooth interface	√	
21	configure USB interfaces	√	
22	enable/disable [**local area network interface**]	√	
23	[**none**]		

On the other hand, the evaluator has reviewed the operational guidance, which includes in its section **2 Management Functions** the following table:

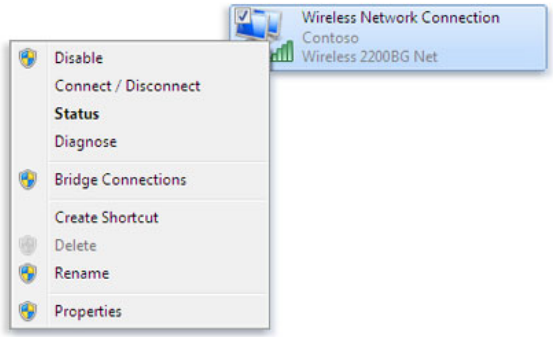
	Activity	Section
1	configure minimum password length	6
2	~~configure minimum number of special characters in password~~	-
3	~~configure minimum number of numeric characters in password~~	-
4	~~configure minimum number of uppercase characters in password~~	-
5	~~configure minimum number of lowercase characters in password~~	-
6	enable/disable screen lock	8
7	configure screen lock inactivity timeout	8
8	configure remote connection inactivitytimeout	8
9	enable/disable unauthenticated logon	6
10	configure lockout policy for unsuccessful authentication attempts through [selection: timeouts between attempts, limiting number of attempts during a time period]	5
11	configure host-based firewall	13
12	configure name/address of directory serverto bind with	14
13	configure name/address of remotemanagement server from which to receivemanagement settings	14
14	~~configure name/address of audit/logging server to which to send audit/logging records~~	-
15	configure local audit storage capacity	3
16	configure audit rules	3
17	configure name/address of network timeserver	15
18	enable/disable automatic software update	12
19	configure WiFi interface	16
20	enable/disable Bluetooth interface	10
21	configure USB interfaces	11

This table allows the evaluator identify where the information about each management function is provided in the operational guidance.

For each management function, the operational guidance provides links to the vendor support webpage, where the information is located. For instance, the information provided in the support webpage about how to configure Wi-Fi interface is as follows:

To enable or disable a network adapter

1. Open Network Connections by clicking the **Start** button, and then clicking **Control Panel**. In the search box, type **adapter**, and then, under Network and Sharing Center, click **View network connections**.
2. Right-click the network adapter, and then do one of the following:
   - To disable the network adapter, click **Disable**. If you're prompted for an administrator password or confirmation, type the password or provide confirmation.
   - To enable the network adapter, click **Enable**. If you're prompted for an administrator password or confirmation, type the password or provide confirmation.



The network adapter menu

If you disable the adapter, you have to enable it again to connect to a network.

Finally, additional information is provided for those management functions that can be performed as a standard user or as administrator, including information about how to operate with each one.

27.2.2. Verdict

The evaluator considers that all the security management functions captured in the security target are properly defined in the operational guidance, providing enough information to allow the evaluator perform each management function.

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.

27.3. Test Activity

27.3.1. Configure minimum password length

27.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- A user account without administrator right shall exist. This user shall belong to the default *Users* group.
- An administrator account shall exist. This account shall belong to default *Administrators* group.
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

27.3.1.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *test1ConfigureMinimumPassword.ps1*. The main command in the script is the following, which allow the evaluator set the minimum password length:

```
#change minium pasword length
net accounts /minpwlen:7
```

3. Observe the result. An audit entry shall be shown.
4. Log out, and log in again, but this time using the user account without administrator rights.
5. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

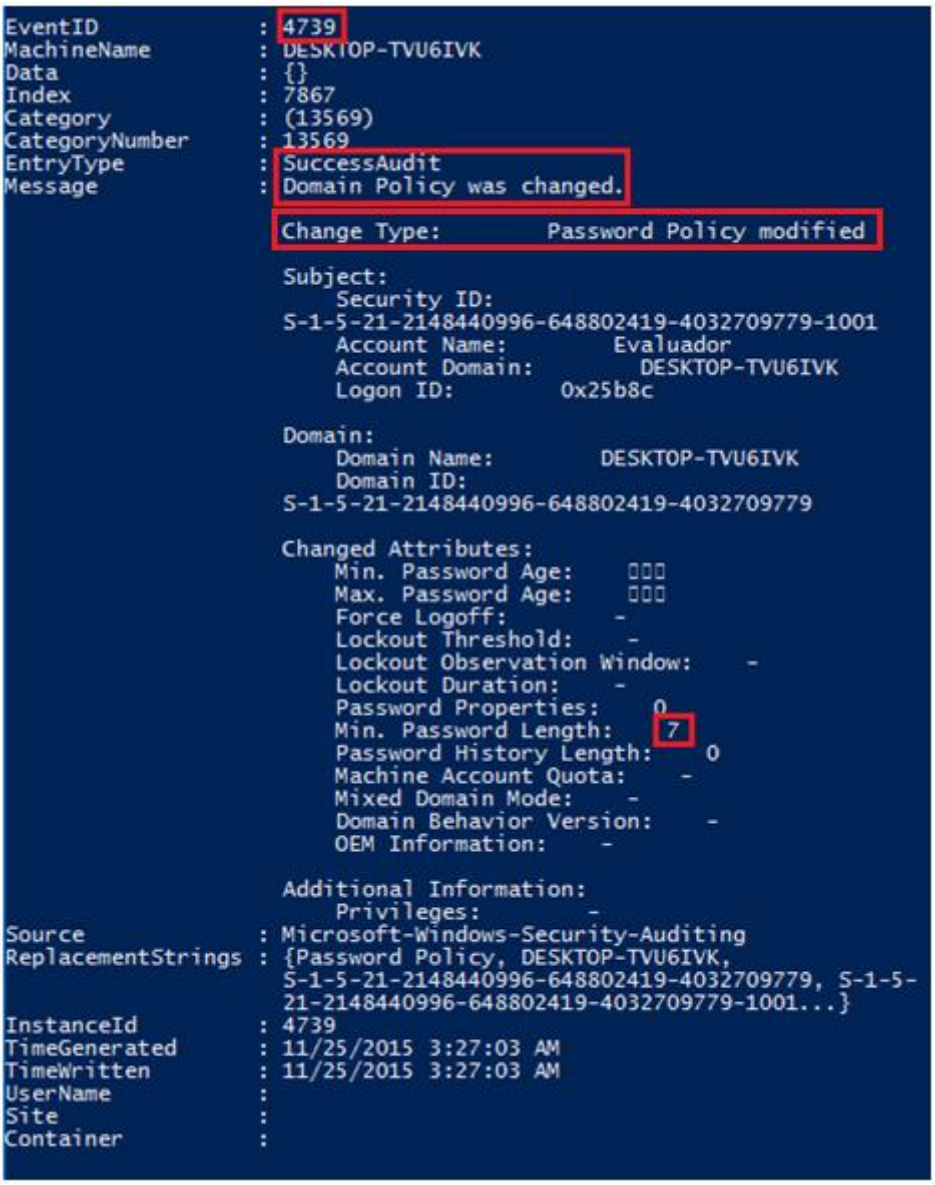
27.3.1.3. Results

The evaluator has performed the test in the following evaluated platforms:

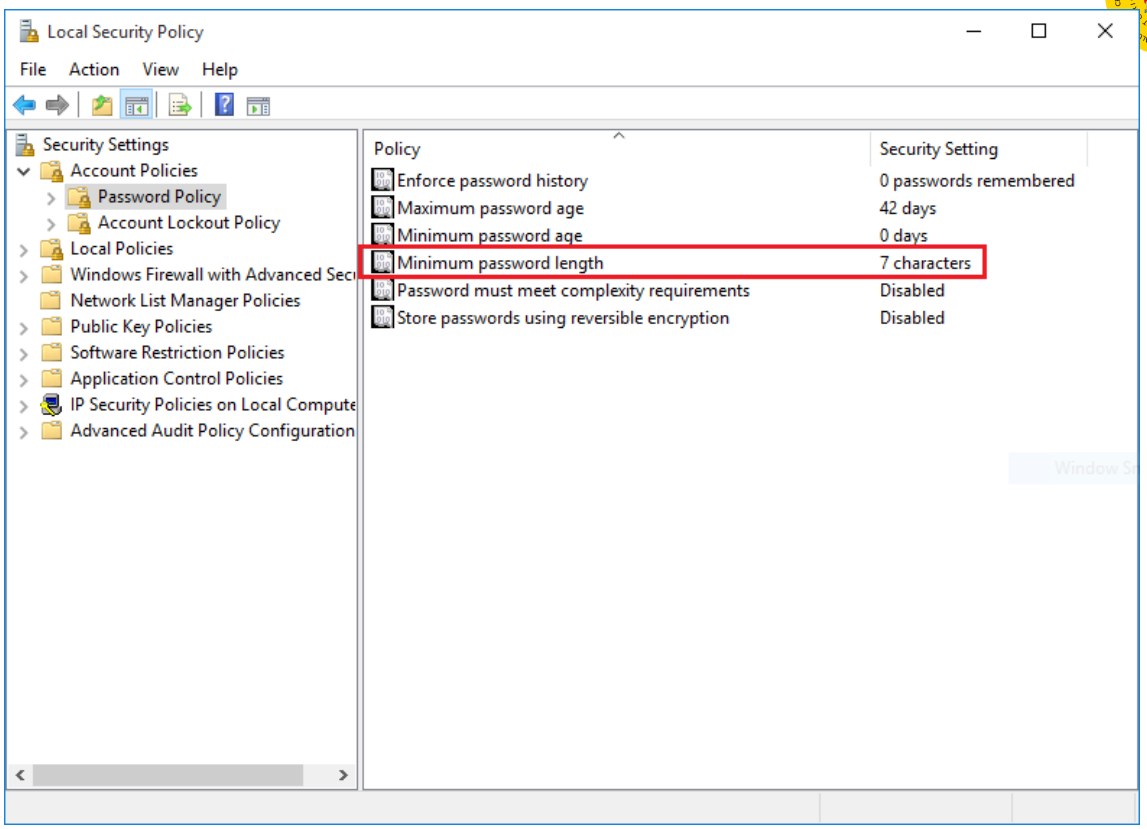
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both user account and administrator account, and the obtained result is as follows:

- Administrator account: The script has been executed properly and the following audit entry has been shown.



As it can be observed in the image above, the change has been applied successfully and the new value is shown. Additionally, the evaluator has checked that the minimum password length has been modified. To do this, the evaluator has opened the *Local Security Policy.*



- User account: The evaluator has obtained the following result when has attempted to execute the script:

```
***Showing information about configure minium password event***
System error 5 has occurred.

Access is denied.
```

27.3.1.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure minimum password length** management function.

27.3.2. Enable/Disable screen lock

27.3.2.1. Setup

The applicable setup for this test is the same as the defined one for *Configure minimum password length test.*

27.3.2.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *test6EnableDisableScreenLock.ps1*. The main command in the script is the following, which allow the evaluator lock the workstation:

```
#lock account
rundll32.exe user32.dll,LockWorkStation
```

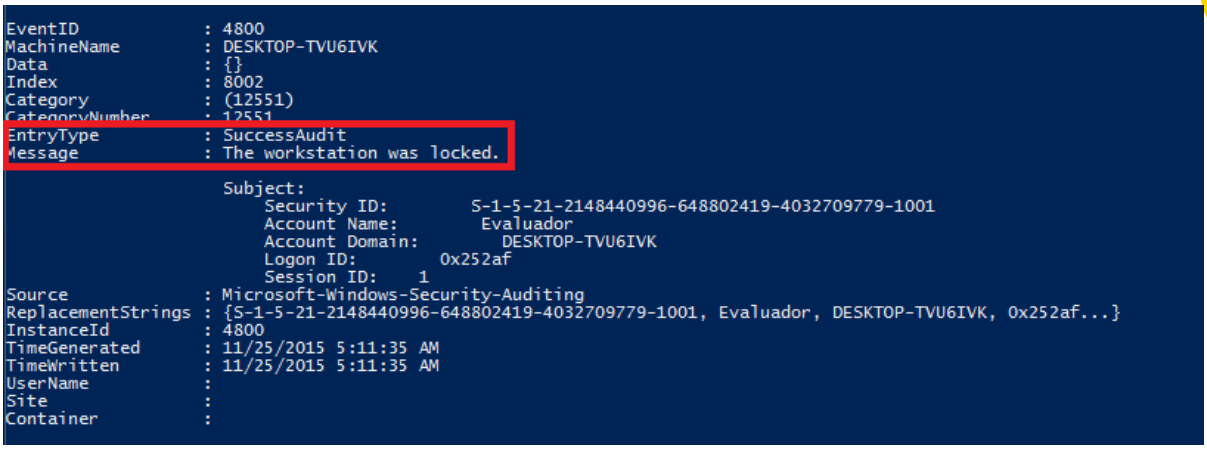
3. Enter the correct password in order to unlock the workstation and observe the results. Two audit entries shall be shown.
4. Observe the result and log out as Administrator.
5. Log out, and log in again, but this time using the user account without administrator rights.
6. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

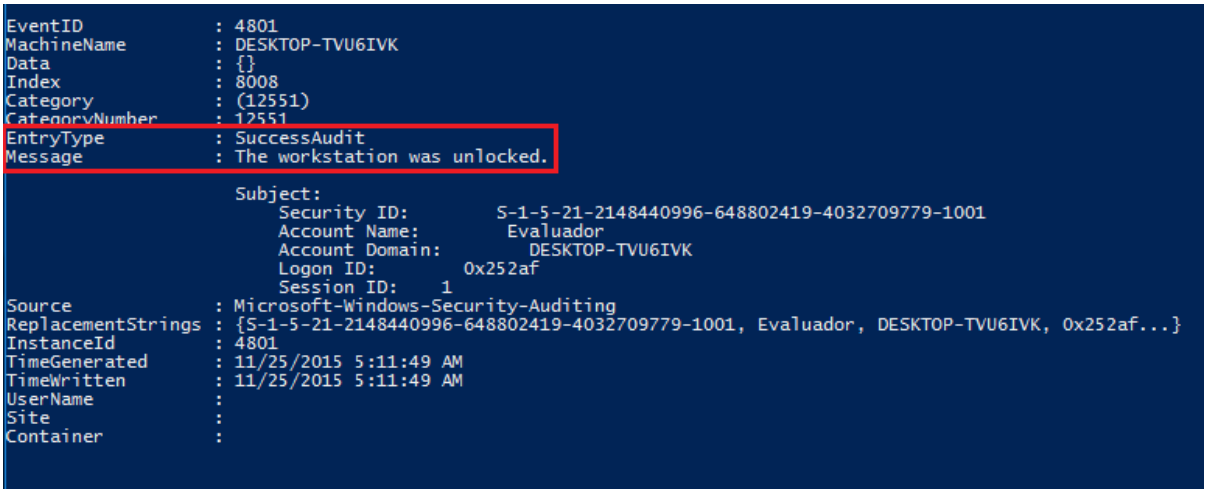
27.3.2.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both user account and administrator account, and has obtained the same result for each one. The following two audit entries have been generated:





27.3.2.4. Verdict

As the above results state, both user with administrator rights and user without administrator rights can enable and disable the screen lock.

Due to this, the evaluator considers that, the behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Enable/Disable screen lock** management function**.**

27.3.3. Configure screen lock inactivity timeout

27.3.3.1. Setup

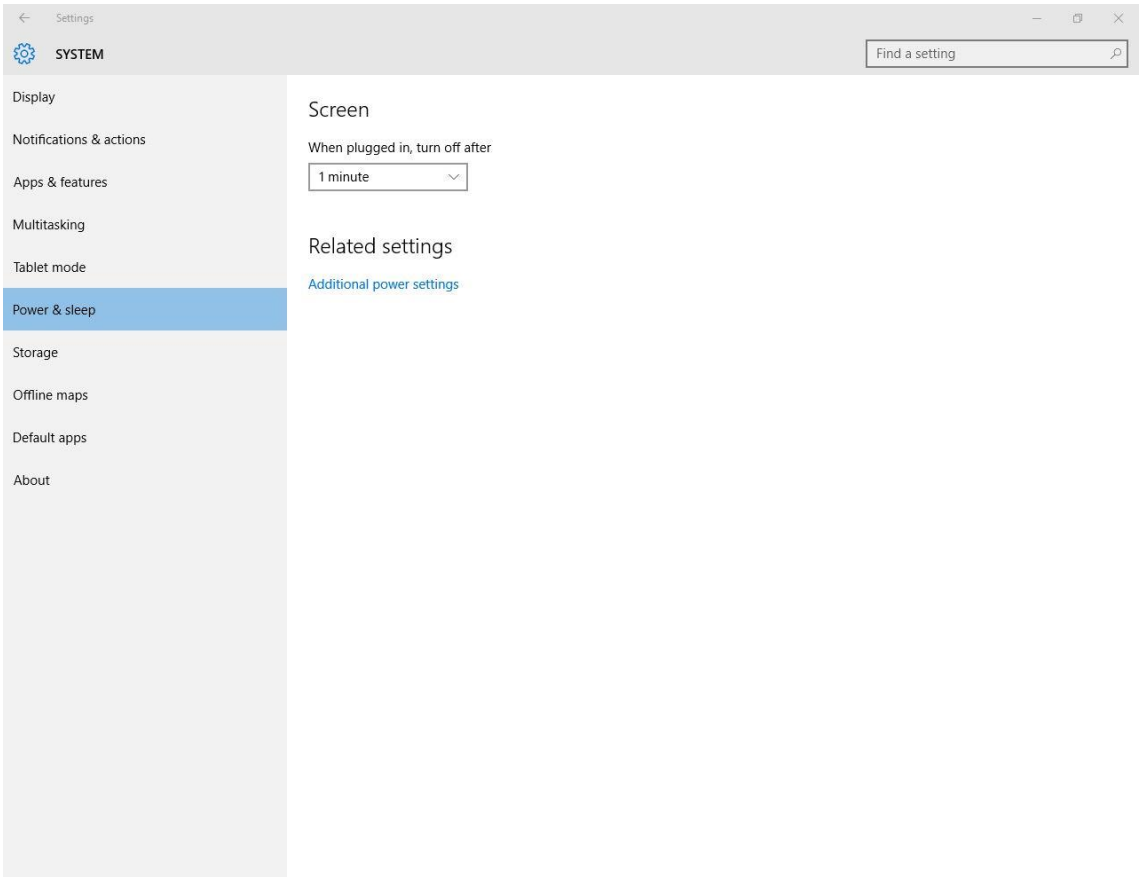
The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.3.2. Procedure

The evaluator shall carry out the following steps in order to perform this test. The procedure steps depend on the operating system installed in the tested platform:.

Windows 10

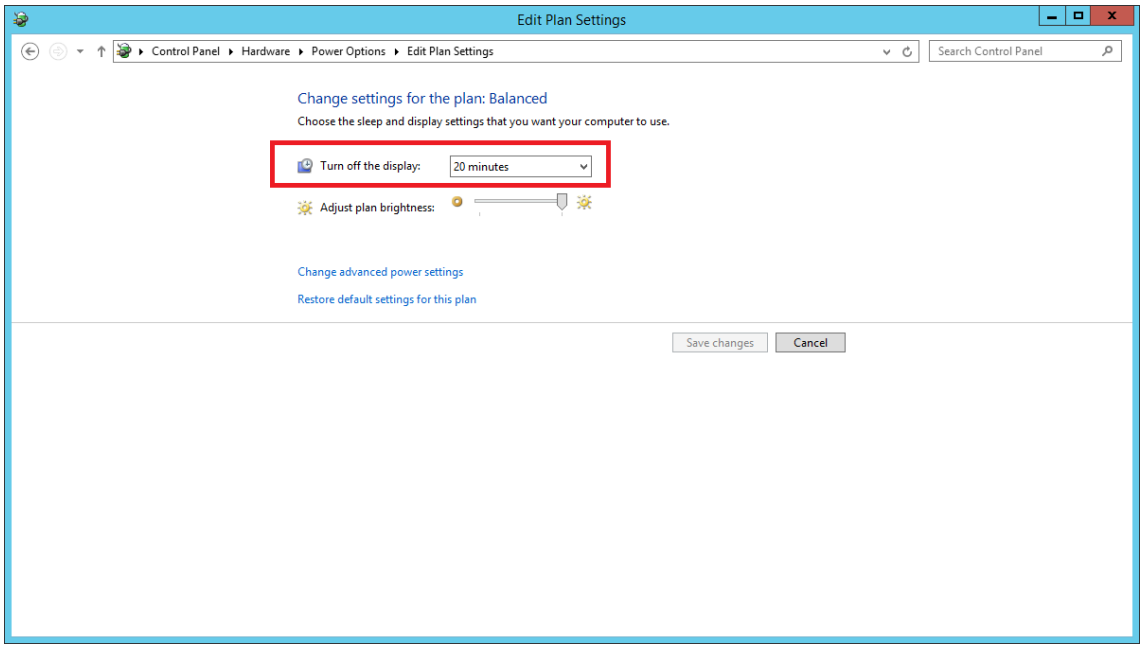
1. Log in using the administrator account.
2. Click-in the *Start* button, and then click-in *Settings* option.
3. After that, go to *System* and select *Power & Setting* in the left menu.
4. Select one of the possible values in the combo box (e.g. 1 minute).



5. Close the window and wait during the configured period time (e.g. 1 minute).
6. Observe the result.
7. Log out and log in again, but this time using the user account without administrator rights.
8. Repeat the steps 2-6.

Windows Server 2012 R2

1. Log in using the administrator account.
2. Go to *Control Panel -> Hardware -> Power Options*.
3. Select *Choose when to turn off the display* option in the left menu. Then, select one of the possible values in *Turn off the display* combo box (e.g. 1 minute)



4. Close the window and wait during the configured period time (e.g. 1 minute).
5. Observe the result.
6. Log out and log in again, but this time using the user account without administrator rights.
7. Repeat the steps 2-5.

27.3.3.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has carried out the procedure defined above using both Windows 10 and Windows Server 2012. In both cases the screen has been locked after the configured inactivity time is reached.

27.3.3.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure screen lock inactivity timeout** management function.

27.3.4. Configure remote connection inactivity timeout

27.3.4.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.4.2. Procedure

To configure the remote connection inactivity timeout, the evaluator shall modify the *MaxConnectionTime* key in the Windows Registry. This modification can be performed in two different ways, manually or automatically.

In order to do this test using the manually method, the evaluator shall open *Local Group Policy Editor* and go to *Computer Configuration -> Administrative templates -> Windows Components -> Remote Desktop Services -> Remote desktop Session Host -> Session Time Limits,* and configure the value for *Set the time limit for active Remote Desktop Services sessions* key.

In order to make easier the test execution, the evaluator has developed a script, which directly modifies the key in the Windows Registry. This method shall be the used one in this procedure. The script source code is as follows:

```
auditpol /set /subcategory:"Registry" /success:enable /failure:enable


$pathFolder = "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services"
$keyName = "MaxConnectionTime"
$fullPath = $pathFolder+"\"+$keyName

If (!(Test-Path -Path $fullPath)){
    New-ItemProperty -Path $pathFolder -Name $keyName -Value "60000" -PropertyType DWORD -Force
}else{
    Set-ItemProperty -Path $pathFolder -Name $keyName -Value "60000" -PropertyType DWORD
}

Start-Sleep -s 1
Get-EventLog Security -InstanceId 4657 -Newest 1
auditpol /set /subcategory:"Registry" /success:disable /failure:disable
```

The evaluator shall carry out the following steps in order to perform this test.

1. Log in using the administrator account.

2. Open PowerShell as administrator and run command “regedit.exe” and in Registry Editor select the folder *HKEY_LOCAL_MACHINE -> Software -> Policies -> Microsoft -> Windows NT -> Terminal Services*

3. After that, right-click and select Permissions… in the context menu to open the Permissions dialog.

4. Click the Advanced button to open the Advanced Security Settings dialog, click in the Auditing tab and click the Add button to open the Auditing Entry dialog.

5. Click the Select a principal to open the Select User or Group dialog, type “Users”, click Check Names and click the OK button.

6. Select Type: All.

7. Select Applies to: This key and sub keys.

8. Click Show advanced permissions and click Set Value.

9. Click OK and Apply.

10. Open a PowerShell terminal as administrator, run the script *test8RemoteInactivityTimeout.ps1* and observe the result

11. Log out and log in again, but this time using the local user account without administrator rights.

12. Open a PowerShell terminal and run “gpedit.msc”.

13. Observe the result.

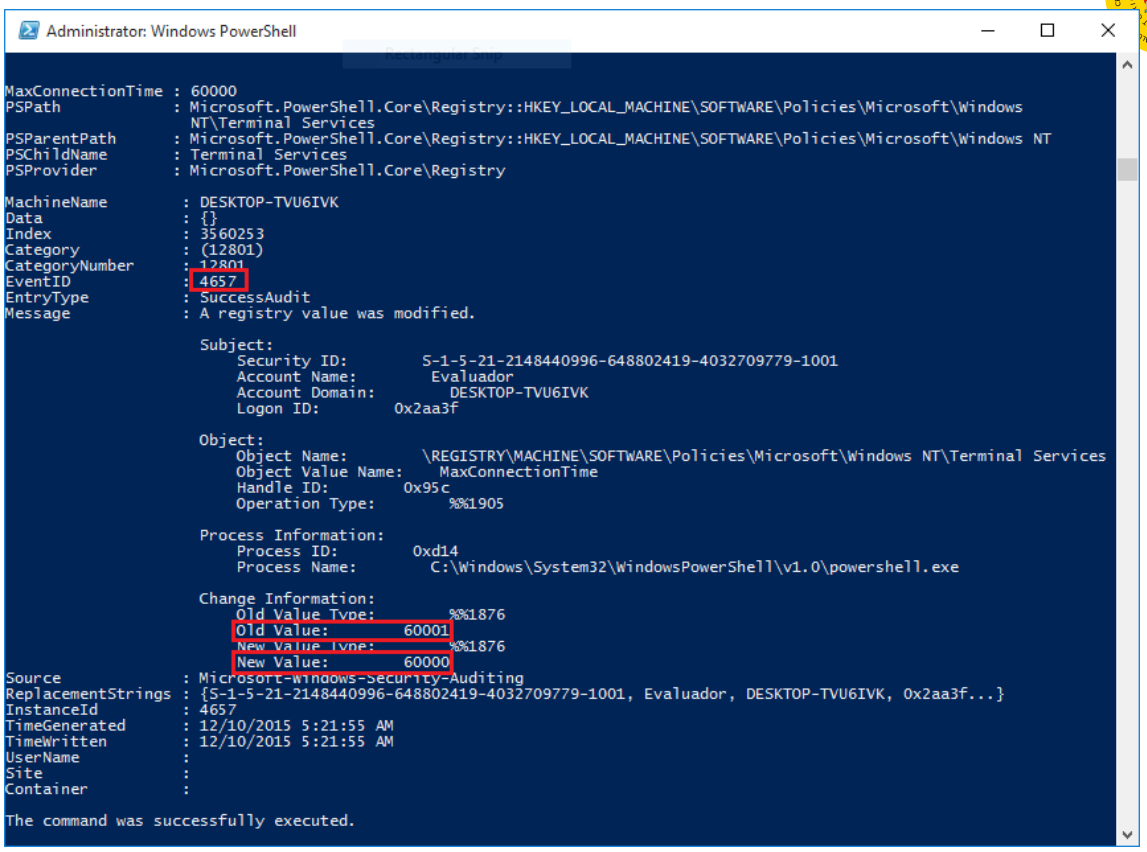
27.3.4.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

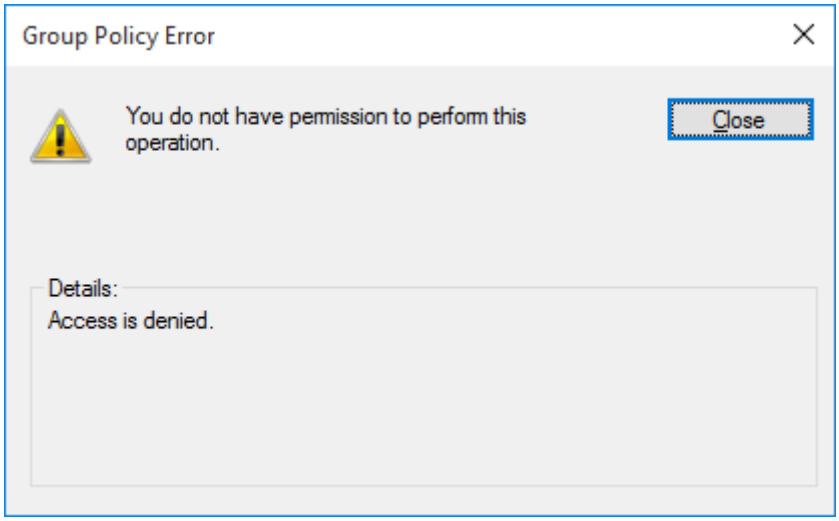
The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

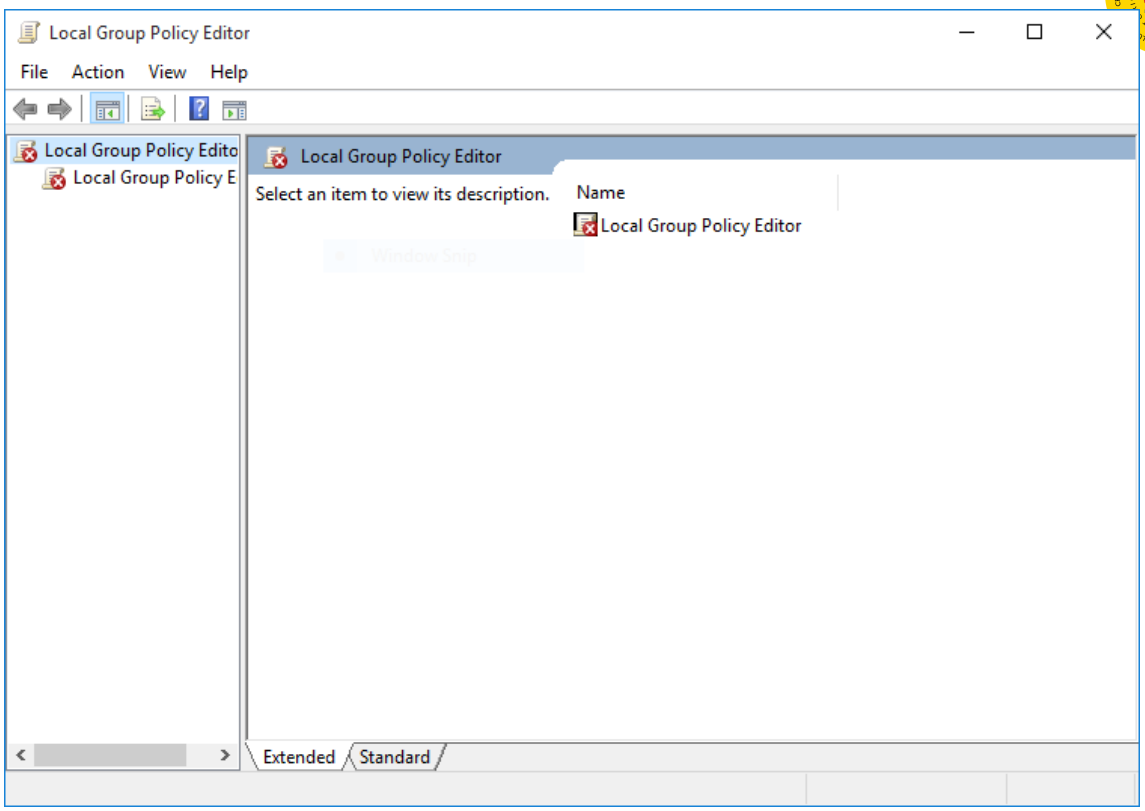
- Administrator account: The script has been executed properly and the following audit entry has been shown.



As it can be observed in the image above, the change has been applied successfully and the new value is shown.

- User account: The evaluator has obtained the following result when has attempted to open *Local Group Policy Editor*:





In addition, user without administrator right cannot execute the script because this kind of accounts has not permissions to modify registry keys.

27.3.4.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure remote connection inactivity timeout** management function

27.3.5. Configure lockout policy for unsuccessful authentication attempts through [select attempts, limiting number of attempts during a time period]

27.3.5.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.5.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *tes10LockoutPolicy.ps1*. The main command in the script is the following, which allow the evaluator configure the lockout policy for the user accounts:

```
net accounts
Start-Sleep -s 1

net accounts /lockoutthreshold:3 /lockoutwindow:30 /lockoutduration:30

Start-Sleep -s 1
net accounts
```

3. Observe the result.
4. Log out, and log in again, but this time using the user account without administrator rights.
5. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

27.3.5.3. Results

The evaluator has performed the test in the following evaluated platforms:

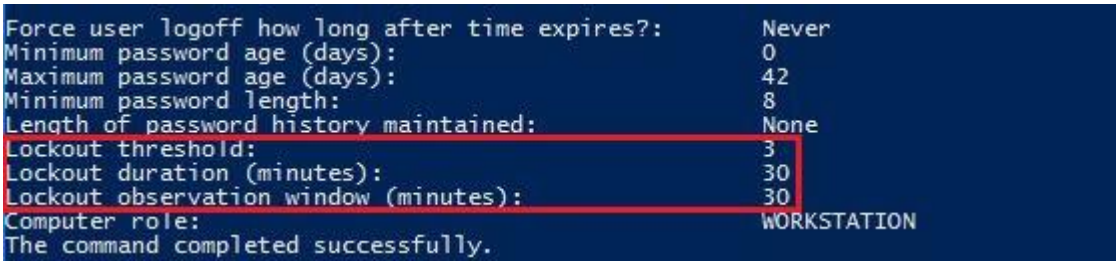
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

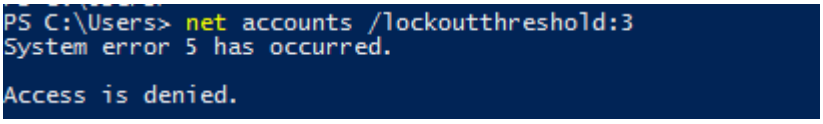
- Administrator account: The script has been executed properly and the following information has been shown. After changing the lockout threshold, the shown information is as follows:

```
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              8
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.
```

Once the threshold has been modified, the information has been updated showing the new values.



- User account: The evaluator has obtained the following result when has attempted to configure lockout threshold:



27.3.5.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure lockout policy for unsuccessful authentication attempts through** management function.

27.3.6. Configure host-based firewall

27.3.6.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.6.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *test11ConfigureHost_basedFirewall.ps1*. The main commands in the script is the following, which allow the evaluator enable or disable the firewall:

```
if($b.Enabled -eq 'True'){
    Set-NetFirewallProfile -Profile Public -Enabled False
}else{

    Set-NetFirewallProfile -Profile Public -Enabled True
}
```

3. Observe the result.
4. Log out and log in again, but this time using a user without administrator rights.
5. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

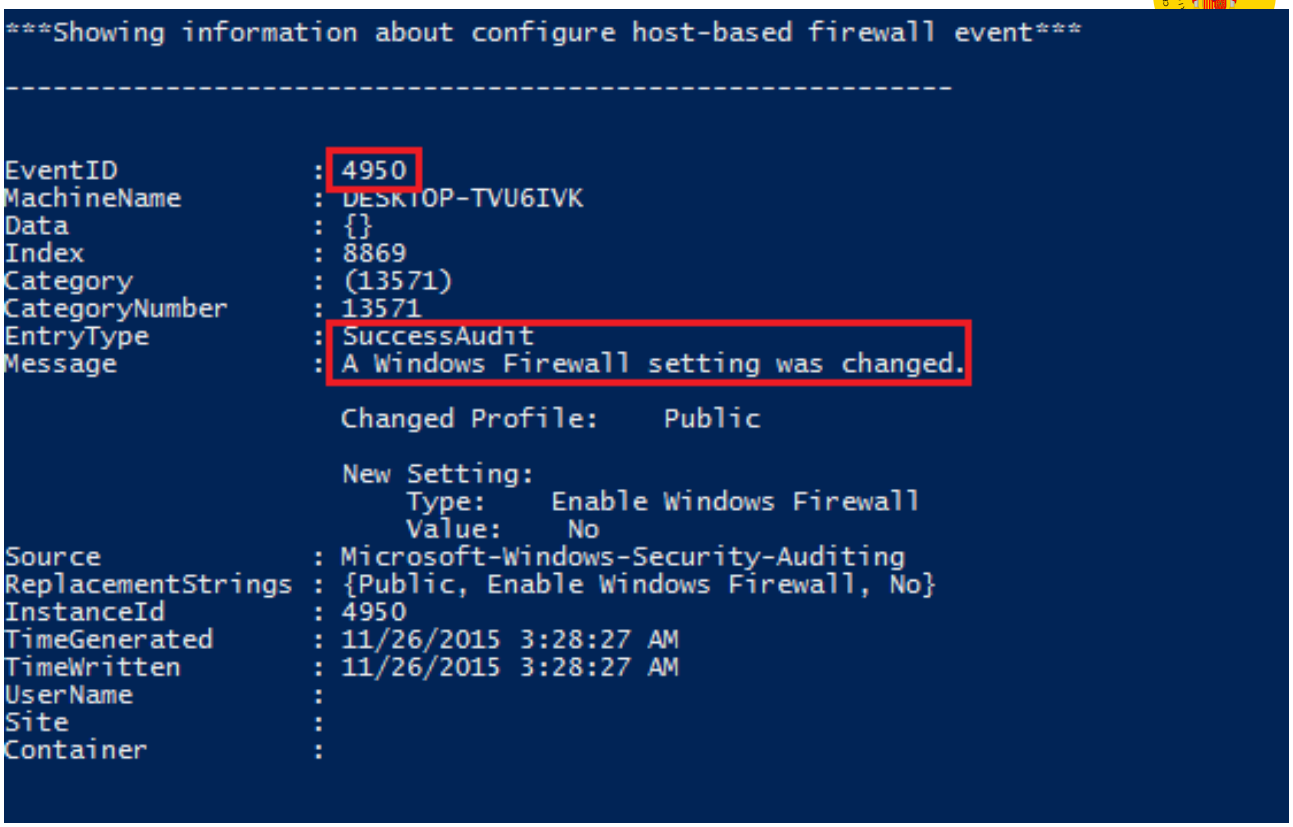
27.3.6.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

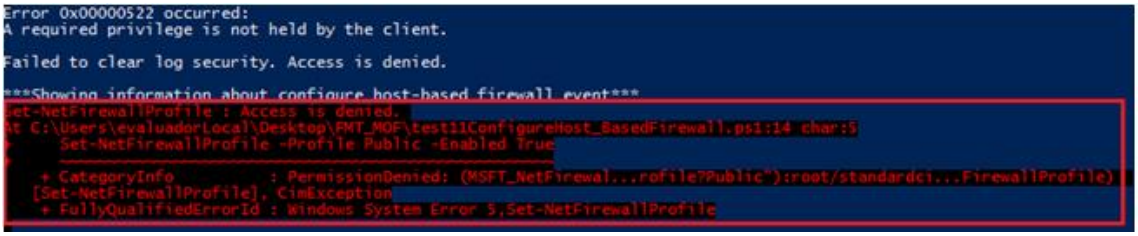
The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

- Administrator account: The script has been executed properly and the following audit entry has been shown.



As it can be observed in the image above, the change has been applied successfully.

- User account: The evaluator has obtained the following result when has attempted to configure firewall with the script:



27.3.6.4. Verdict

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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure host-based firewall** management function.

27.3.7. Configure name/address of directory server to bind with & Configure name/address of remote management server from which to receive management settings.

27.3.7.1. Setup

In order to configure a domain controller, the applicable setup for this test is described in *Configuring Domain Controller and Certificate Authority* document. Moreover, the setup defined for *Configure minimum password length* test is also applicable.

In order 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 Edition, this test is not applicable for the following evaluated platform:

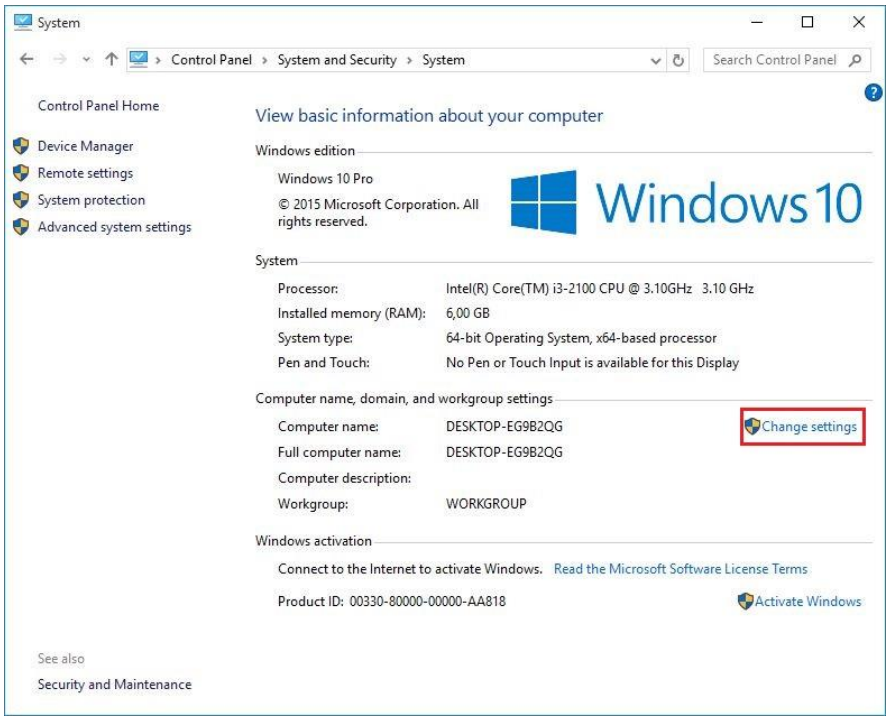
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.

27.3.7.2. Procedure

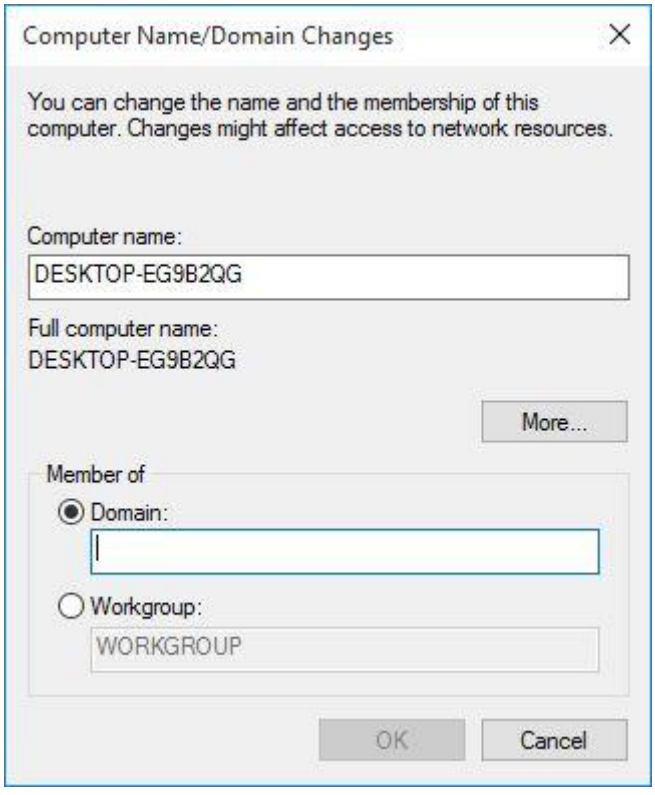
In order to perform this test, the evaluator has followed the steps defined below depends on the user account type.

User with administrator rights:

1. Log in as Administrator in the computer which is going to be joined to the domain.
2. Open PowerShell as administrator and run the following command: *auditpol /set /category:"Account Management" /success:enable /failure:enable*.
3. Right-click on the Start button and select Control Panel.
4. Go to System and Security -> System. After that and in the System window, click-over Change Settings.



5. In the System Properties windows, click in Change button and then enter the domain name in the domain text box.



6. Once the computer has been joined to the domain, run a PowerShell terminal as administrator and write the following commands:

- *Get-EventLog System –InstanceID 3260 –Newest 1 | fl **
- *Auditpol /set / category:"Account Management" /success:disable /failure:disable*

7. Observe the result.
8. Open a PowerShell terminal as Administrator in the domain controller machine, run the following command and observe the result: *Get-EventLog Security –InstanceID 4741 –Newest 1 | fl **

User without administrator rights:

9. In the domain-joined computer, log out log in again, but this time using a user without administrator rights.
10. Repeat the steps 3 to 5.
11. A prompt asking for the password of the administrator account shall be shown.

27.3.7.3. Results

The evaluator has performed the test in the following evaluated platforms:

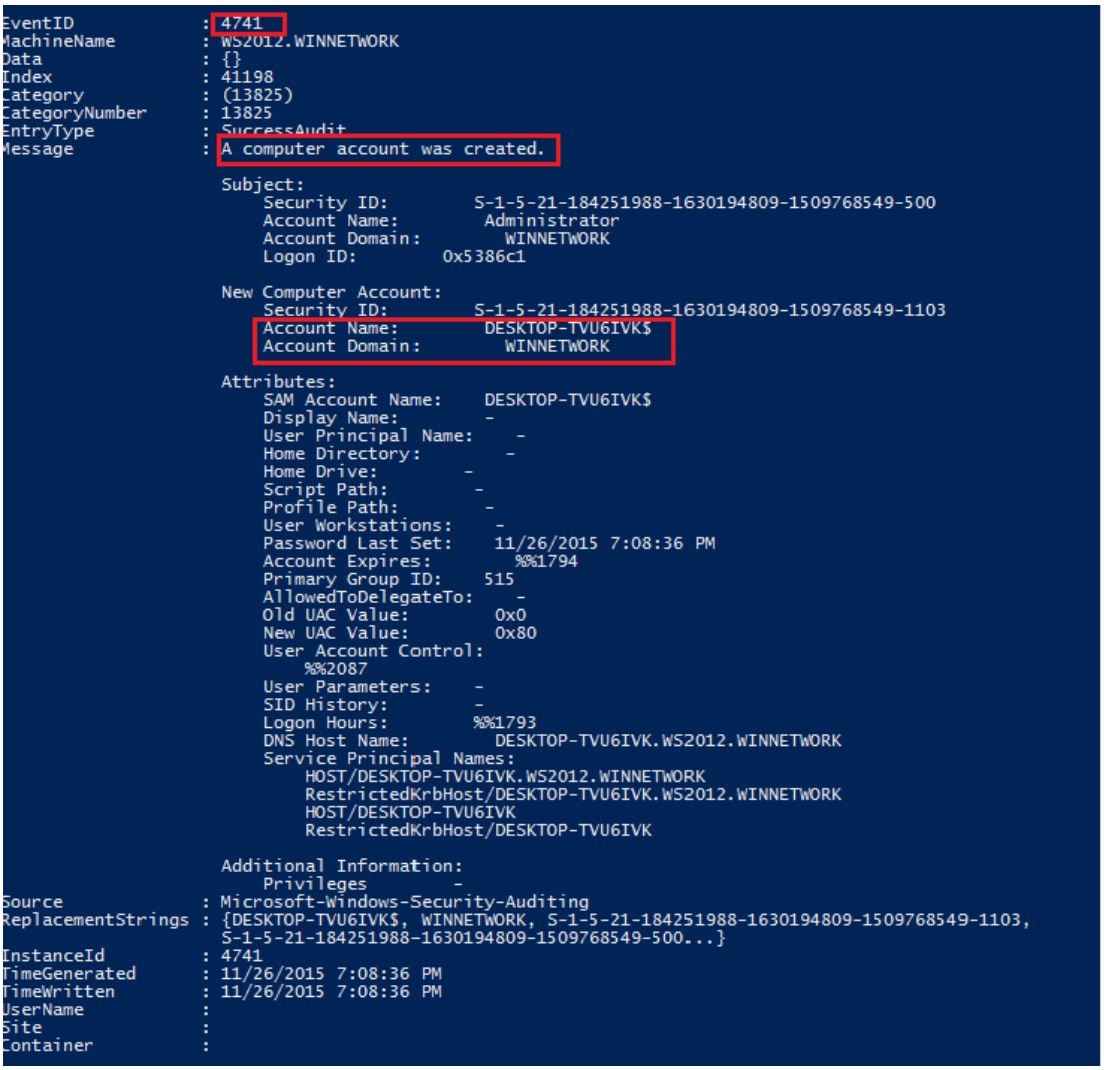
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

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

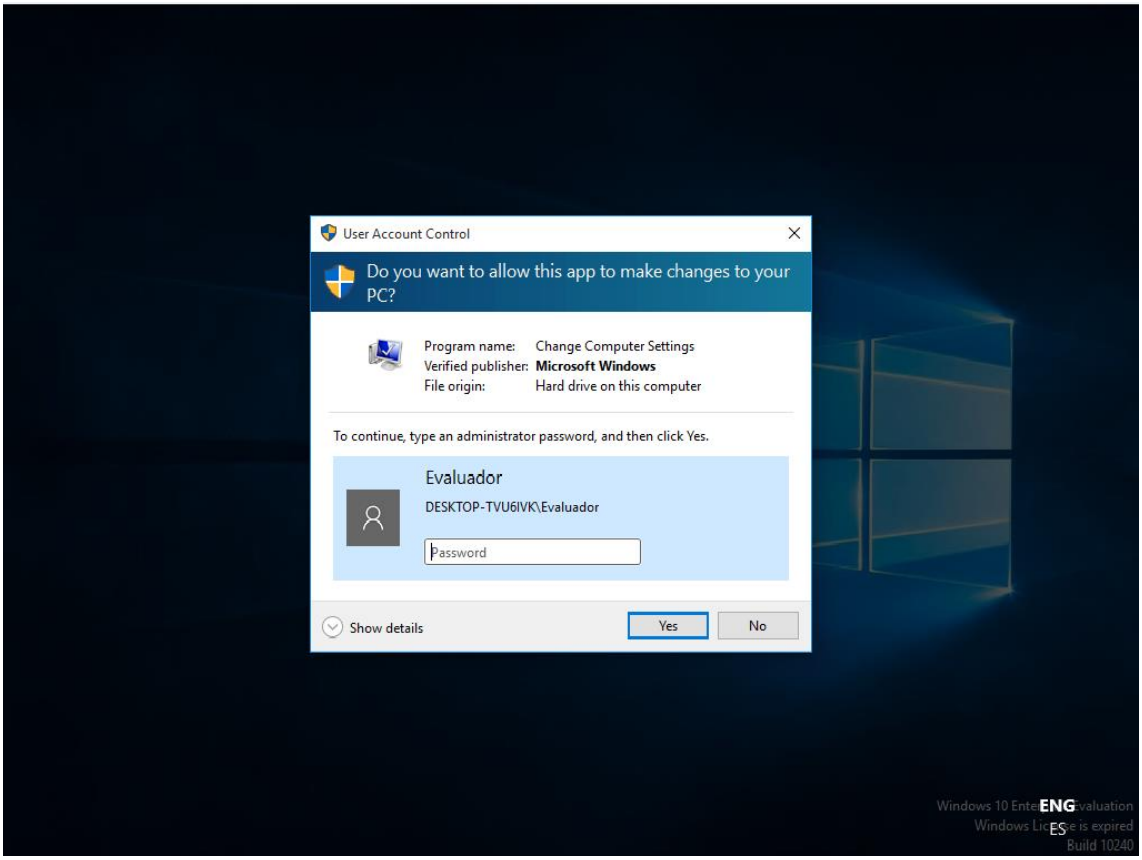
The evaluator has performed the test using an administrator account in the computer which is going to be joined to the domain and the domain controller, and has obtained the audit result for each one. The following two audit entries have been generated when an administrator account is used. The first one shows the audit log generated in the domain-joined computer:

```
EventID             : 3260
MachineName         : DESKTOP-TVU6IVK
Data                : {}
Index               : 3024
Category            : (0)
CategoryNumber      : 0
EntryType           : Information
Message             : This computer has been successfully joined to domain 'WINNETWORK'.
Source              : Workstation
ReplacementStrings  : {WINNETWORK}
InstanceId          : 3260
TimeGenerated       : 11/26/2015 7:10:28 AM
TimeWritten         : 11/26/2015 7:10:28 AM
UserName            :
Site                :
Container           :
```

And the second one shows the audit log generated in the domain controller computer, when the computer has been joined to the domain.



On the other hand, the evaluator has obtained a prompt asking for the administrator password when has attempted to join the tested platform to the domain. The below image shows this fact:



27.3.7.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure name/address of directory server to bind with… & Configure name/address of remote management server from...** management functions.

27.3.8. Configure local audit storage capacity

27.3.8.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.8.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *Test15ConfigureLocalAuditStorageCapacity.ps1*. The main commands in the script is the following, which allow the evaluator configure the storage capacity:

```
#Set the maximum size of log
wevtutil sl security /ms:20971994
```

3. Observe the result.
4. Log out log in again, but this time using a user without administrator rights.
5. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

27.3.8.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

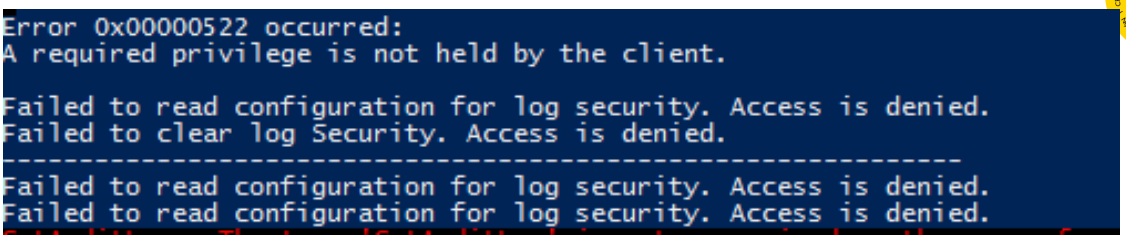
- Administrator account: The script has been executed properly and the local audit storage capacity has been modified. The following message is shown after modifying the limit and the audit entry has been generated.

```
The command was successfully executed.
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
-------------------------------------------------------------------
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: 20971994
publishing:
  fileMax: 1
```

```
EventID             : 4657
MachineName         : DESKTOP-TVU6IVK
Data                : {}
Index               : 9064
Category            : (12801)
CategoryNumber      : 12801
EntryType           : SuccessAudit
Message             : A registry value was modified.

                      Subject:
                          Security ID:        S-1-5-21-2148440996-648802419-4032709779-1001
                          Account Name:       Evaluador
                          Account Domain:     DESKTOP-TVU6IVK
                          Logon ID:           0x3d204

                      Object:
                          Object Name:        \REGISTRY\MACHINE\SYSTEM\ControlSet001\Services\EventLog\Security
                          Object Value Name:  MaxSizeUpper
                          Handle ID:          0xab4
                          Operation Type:     %%1904

                      Process Information:
                          Process ID:         0x38c
                          Process Name:       C:\Windows\System32\svchost.exe

                      Change Information:
                          Old Value Type:     -
                          Old Value:          -
                          New Value Type:     %%1876
                          New Value:          0
Source              : Microsoft-Windows-Security-Auditing
ReplacementStrings  : {S-1-5-21-2148440996-648802419-4032709779-1001, Evaluador, DESKTOP-TVU6IVK, 0x3d204...}
InstanceId          : 4657
TimeGenerated       : 11/26/2015 5:00:13 AM
TimeWritten         : 11/26/2015 5:00:13 AM
UserName            :
Site                :
Container           :
```

As it can be observed in the image above, the change has been applied successfully in the storage capacity log.

- User account: The evaluator has obtained the following error message when has attempted to configure storage capacity using the script:



27.3.8.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure local audit storage capacity** management function.

27.3.9. Configure audit rules

27.3.9.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.9.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open PowerShell as administrator and run the following commands: *auditpol /get /category:*.*
3. Observe the section *System Integrity* and write the following command in order to enable the audit for this subcategory: *auditpol /set /subcategory:"System Integrity" /success:enable /failure:enable.*
4. Repeat the step 2 and observe the section *System Integrity*. The value of this field shall change.
5. Execute the following command in order to disable the audit for this subcategory: *auditpol /set /subcategory:"System Integrity" /success:disable /failure:disable.*
6. Repeat the step 2 and observe the section *System Integrity*. The value of this field shall change again.
7. Log out log in again, but this time using a user without administrator rights.

8. Repeat the steps 2 to 6, running the commands from a non-administrator PowerShell terminal.

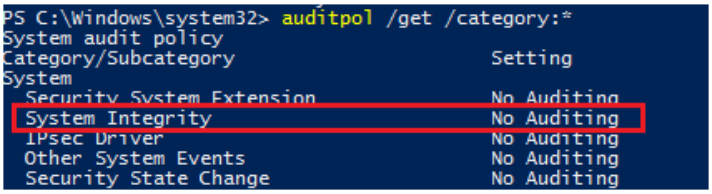
27.3.9.3. Results

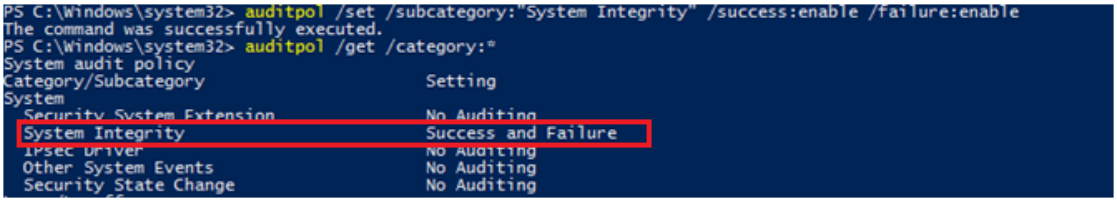
The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

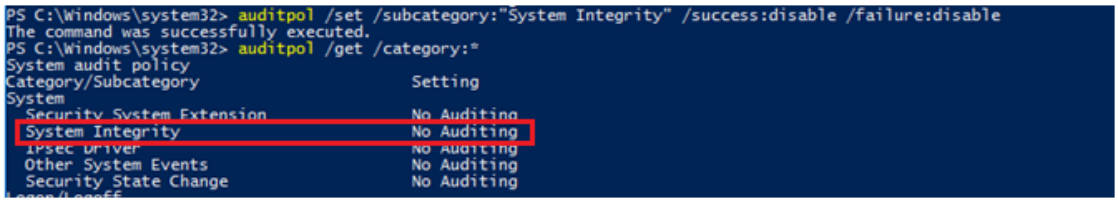
The evaluator has obtained the same results for all the tested platforms.

The evaluator has performed the test using both local user account and administrator account. The evaluator has obtained the following result when an administrator account has been used. Initially, the audit for System Integrity subcategory is disabled. After enabling the audit for this category, the change is shown:



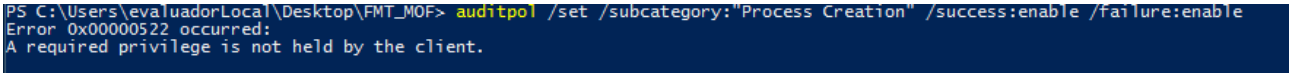


Once the audit is disabled, the change is shown.



So, as it can be observed in the images above, the evaluator has disabled or enabled successfully the audit rule using a user with administrator rights.

However, the evaluator has obtained the following result when has attempted to configure audit rules using a user without administrator rights:



```
PS C:\Users\evaluadorLocal\Desktop\FMT_MOF> auditpol /set /subcategory:"Process Creation" /success:disable /failure:disable
Error 0x00000522 occurred:
A required privilege is not held by the client.
```

27.3.9.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure audit rules** management function.

27.3.10. Configure name/address of network time server

27.3.10.1. Setup

The applicable setup for this test is the same as the one defined for *Configure name/address of directory server to bind with* test.

27.3.10.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Open a PowerShell terminal as administrator and run the script *test17ConfigureNameAddressOfNetworkTimeServer.ps1*. The main commands in the script are the following, which allow the evaluator configure the address of network time server:

```
w32tm /config /manualpeerlist:time.nist.gov /syncfromflags:manual /update
w32tm /config /syncfromflags:manual /update
```

3. Observe the result.
4. Log out and log in as local user, but this time using a user without administrator rights.
5. Repeat the steps 2-3, running the script from a non-administrator PowerShell terminal.

27.3.10.3. Results

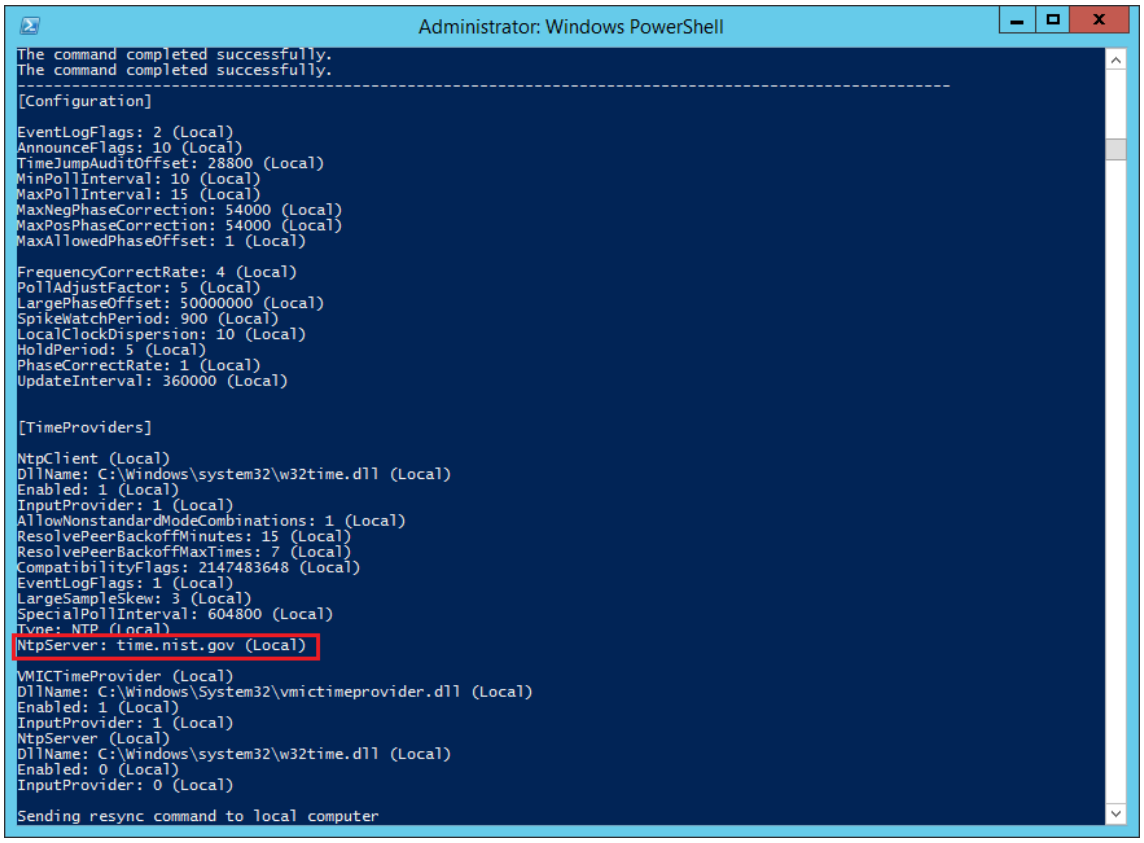
The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.

- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

- Administrator account: The script has been executed properly and the following information has been shown after configuring the address of the network time server.



As it can be observed in the image above, the evaluator has configured the address of time server and information about the name of the new network time server is shown.

- User account: The evaluator has obtained the following result when has attempted to configure address of network time server:

```
PS C:\Users\evaluadorLocal\Desktop\FMT_MOF> w32tm /config /syncfromflags:DOMHIER /update
The following error occurred: Access is denied. (0x80070005)
PS C:\Users\evaluadorLocal\Desktop\FMT_MOF>
```

27.3.10.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure name/address of network time server** management function.

27.3.11. Enable/disable automatic software update

27.3.11.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.11.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

User with administrator rights:

1. Log in using the administrator account.
2. Open Group Policy Editor (*gpedic.msc*) and navigate to *Computer Configuration -> Administrative Templates -> Windows Components -> Windows Update.*
3. Set the "*Configure Automatic Updates*" value to the disabled state and click *Apply*.
4. Open a PowerShell terminal as administrator and run command "*regedit.exe*" and in Registry Editor select the folder *HKEY_LOCAL_MACHINE -> Software -> Policies -> Microsoft -> Windows -> Windows Update -> AU.*
5. After that, right-click and select *Permissions…* on the context menu to open the Permissions dialog.
6. Click the *Advanced* button to open the *Advanced Security Settings* dialog, click in the *Auditing* tab and click the *Add* button to open *the Auditing Entry* dialog.
7. Click the *Select* a principal to open the *Select User* or *Group* dialog, type "*Users*", click *Check Names* and click the *OK* button.
8. Select Type: *All*
9. Select Applies to: *This key and sub keys*.
10. Click Show advanced permissions and click *Set Value*.
11. Click *OK* and *Apply*.
12. Open a PowerShell terminal as administrator and write the following commands:

    - *auditpol /set /subcategory:"Registry" /success:enable /failure:enable*

- *wevtutil cl Security*

13. Open Group Policy Editor utility *gpedic.msc* and navigate to Computer *Configuration -> Administrative Templates -> Windows Components -> Windows Update.*
14. Set the "*Configure Automatic Updates*" value to the enabled state and click *Apply*.
15. Open a PowerShell terminal as administrator and write the following commands:
    - *Get-EventLog Security –InstanceId 4657 –Newest 1 | fl **
    - *auditpol /set /subcategory:"Registry" /success:disable /failure:disable*
16. Observe the result.

User without administrator rights:

17. Log out and log in again, but this time using a user without administrator rights.
18. Repeat the steps 13-14.
19. Observe the result.

27.3.11.3. Results

The evaluator has performed the test in the following evaluated platforms:

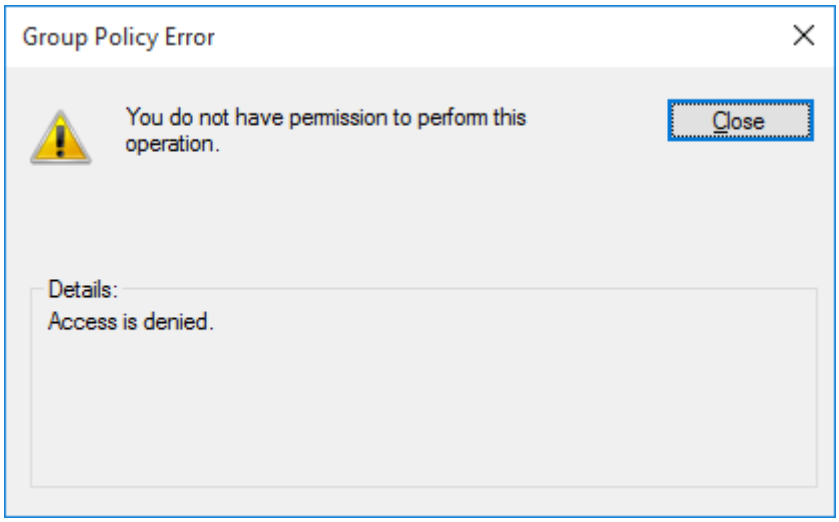
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

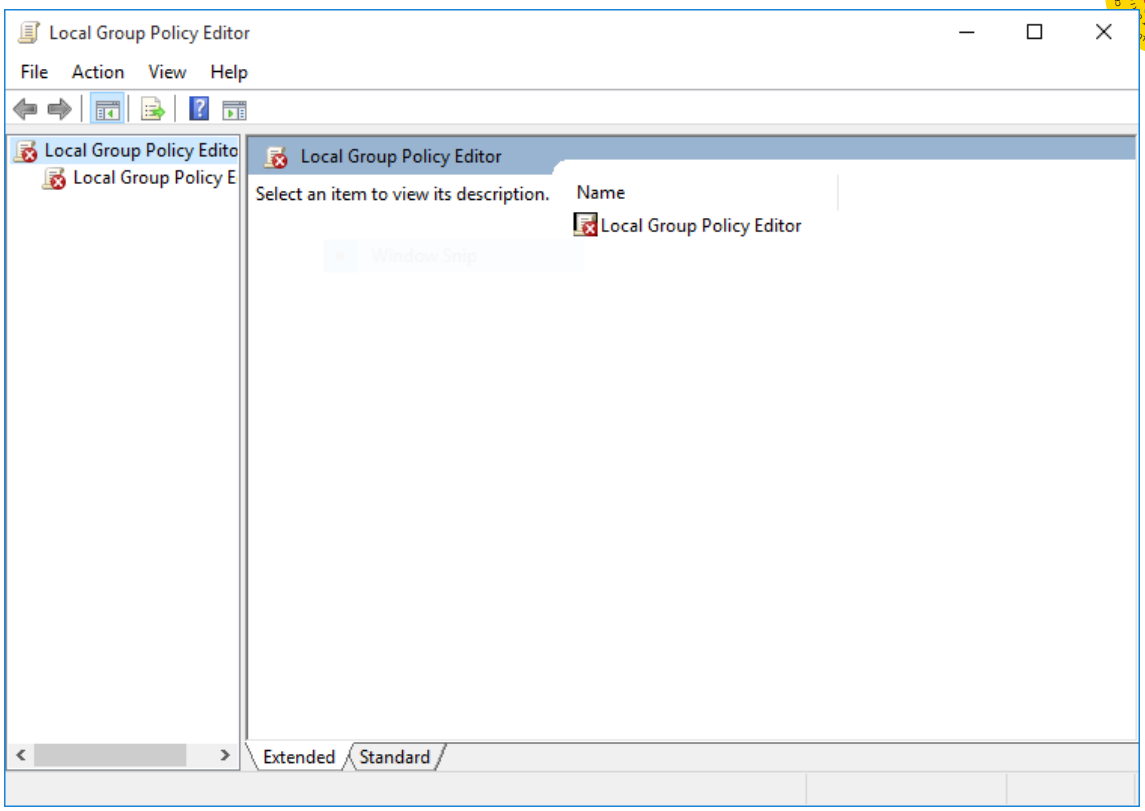
The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the above steps using both local user account and administrator account, and the obtained result is as follows:

- Administrator account: The evaluator has enabled and has disabled the automatic software updates using a user with administrator rights. Additionally and after the value modification, the following audit event has been generated:

```
EventID            : 4657
MachineName        : DESKTOP-TVU6IVK
Data               : {}
Index              : 16049
Category           : (12801)
CategoryNumber     : 12801
EntryType          : SuccessAudit
Message            : A registry value was modified.

                     Subject:
                         Security ID:        S-1-5-18
                         Account Name:       DESKTOP-TVU6IVK$
                         Account Domain:     WORKGROUP
                         Logon ID:           0x3e7

                     Object:
                         Object Name:        \REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU
                         Object Value Name:  ScheduledInstallTime
                         Handle ID:          0x10d0
                         Operation Type:     %%1906

                     Process Information:
                         Process ID:         0x35c
                         Process Name:       C:\Windows\System32\svchost.exe

                     Change Information:
                         Old Value Type:     %%1876
                         Old Value:          3
                         New Value Type:     -
                         New Value:          -
Source             : Microsoft-Windows-Security-Auditing
ReplacementStrings : {S-1-5-18, DESKTOP-TVU6IVK$, WORKGROUP, 0x3e7...}
InstanceId         : 4657
TimeGenerated      : 11/27/2015 1:48:46 AM
TimeWritten        : 11/27/2015 1:48:46 AM
UserName           :
Site               :
Container          :
```

- User account: The evaluator has obtained the following error message when has attempted to open *Local Group Policy Editor* in order to modify the value for the automatic software updates:





27.3.11.4. Verdict

As the above results state, the automatic software update can only be enabled or disabled 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Enable/Disable automatic software update** management function.

27.3.12. Configure Wi-Fi interface

27.3.12.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

In order to perform this test, the evaluated platforms should have a Wi-Fi interface. Since not all of the evaluated platforms have this kind of hardware, this test is not applicable for the following evaluated platform:

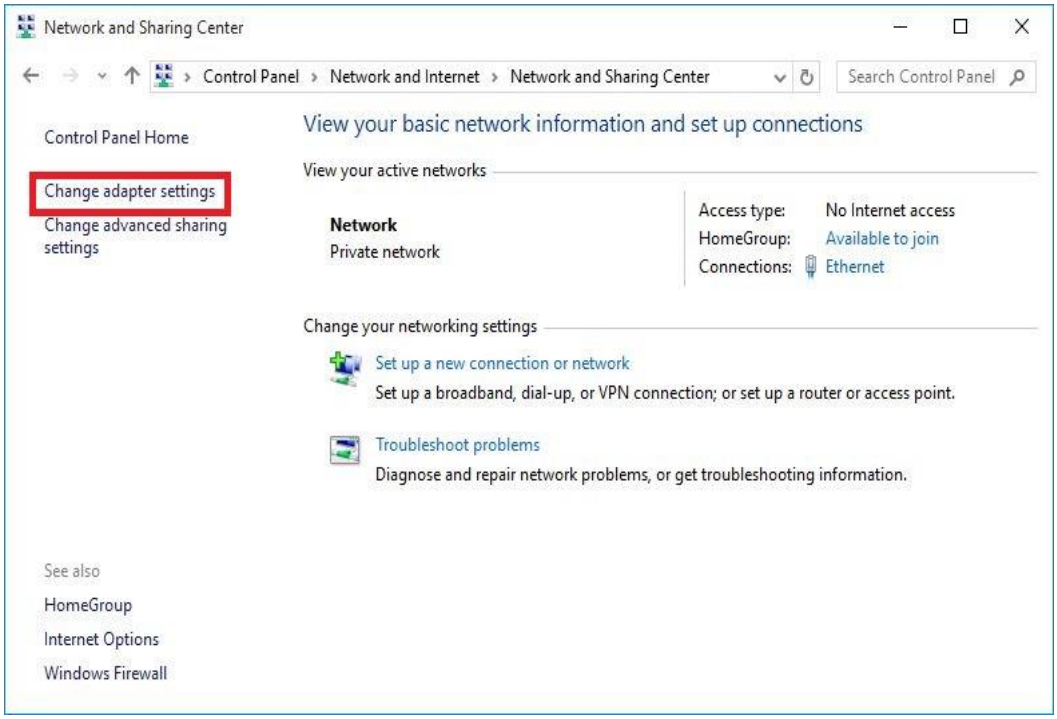
- Dell 755 Optiplex with Windows 10 x86 Pro Edition.
- Dell 755 Optiplex with Windows 10 x86 Enterprise Edition.

- Dell 755 Optiplex with Windows Server 2012 R2 Datacenter Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

27.3.12.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using a user with administrator rights.
2. Go to the *Network and Internet -> Network and Sharing Center* in the *Control Panel*. Click-in "*Change adapter settings*" link to open the *Network Connections* window.



3. Right-click the Wi-Fi adapter and select "*Enable*" or "*Disable*" to modify the interface state.
4. Start PowerShell and run the following command and observe the result: *Get-NetAdapter.*
5. After that, log out and log in using a user without administrator rights.
6. Repeat the steps 2-5.

27.3.12.3. Results

This test can only be performed in those evaluated platforms which have Wi-Fi interface. This test is not applicable for Dell platforms and those platforms running over Windows Server

2012 Hyper-V. Due to this, the evaluator has performed this test in the following evaluated platform:

- Surface 3 with Windows 10 x64 Enterprise Edition

The evaluator has obtained the following evidence using a user with administrator rights. After disabling the Wi-Fi interface the state of the interface is as follows:

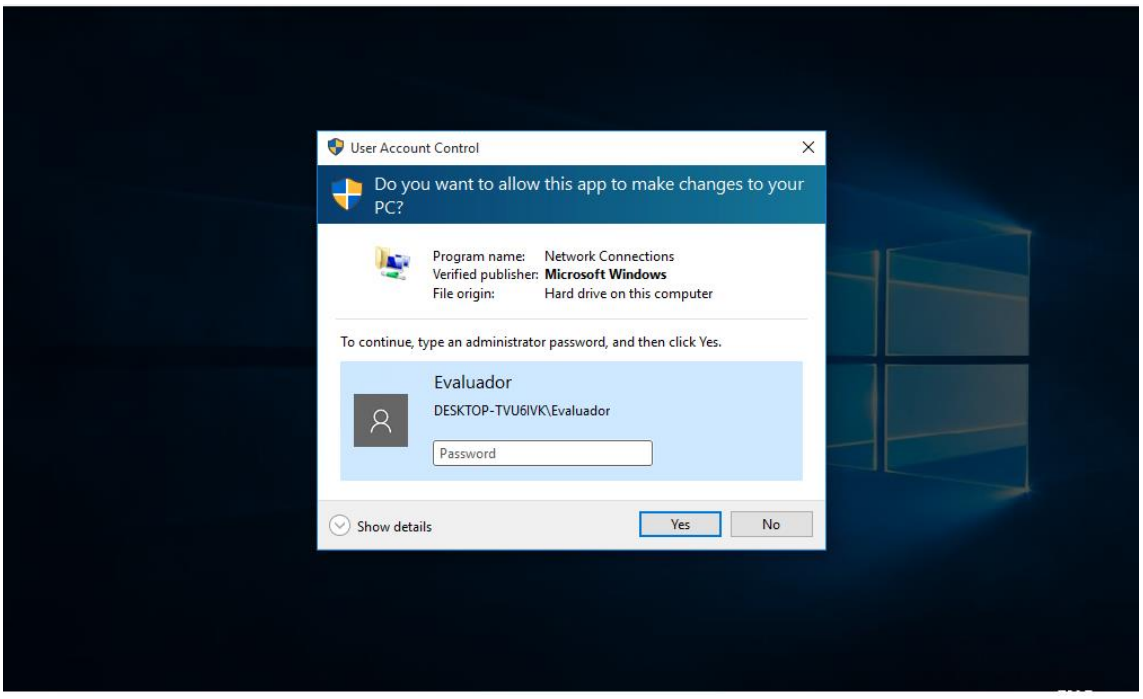
```
Confirm
Are you sure you want to perform this action?
Disable-NetAdapter 'Wi-Fi'
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): Y

Name                     InterfaceDescription                    ifIndex Status       MacAddress
----                     --------------------                    ------- ------       ----------
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       5 Disconnected B4-AE-2B-2C-79-08
Wi-Fi                    Marvell AVASTAR Wireless-AC Network ...       7 Disabled     B4-AE-2B-2C-79-07
```

Once the Wifi interface is enabled again, the state is shown as follows.

```
Name                     InterfaceDescription                    ifIndex Status       MacAddress
----                     --------------------                    ------- ------       ----------
Bluetooth Network Conn... Bluetooth Device (Personal Area Netw...       5 Disconnected B4-AE-2B-2C-79-08
Wi-Fi                    Marvell AVASTAR Wireless-AC Network ...       7 Disconnected B4-AE-2B-2C-79-07
```

As it can be observed in the images above, the change has been applied successfully and the evaluator has checked that the state of Wi-Fi interface has been modified.

The evaluator has attempted to modify the state of Wi-Fi interface using a user without administrator rights, and the following prompt is shown asking for the administrator password.



27.3.12.4. Verdict

As the above results state, a Wi-Fi interface can only be enabled or disabled using a user with administrator rights. When a user without administrator rights is used, a prompt is shown asking for the administrator password.

Due to this, the evaluator considers that, the behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure Wi-Fi interface** management function.

27.3.13. Enable/disable Bluetooth interface

27.3.13.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

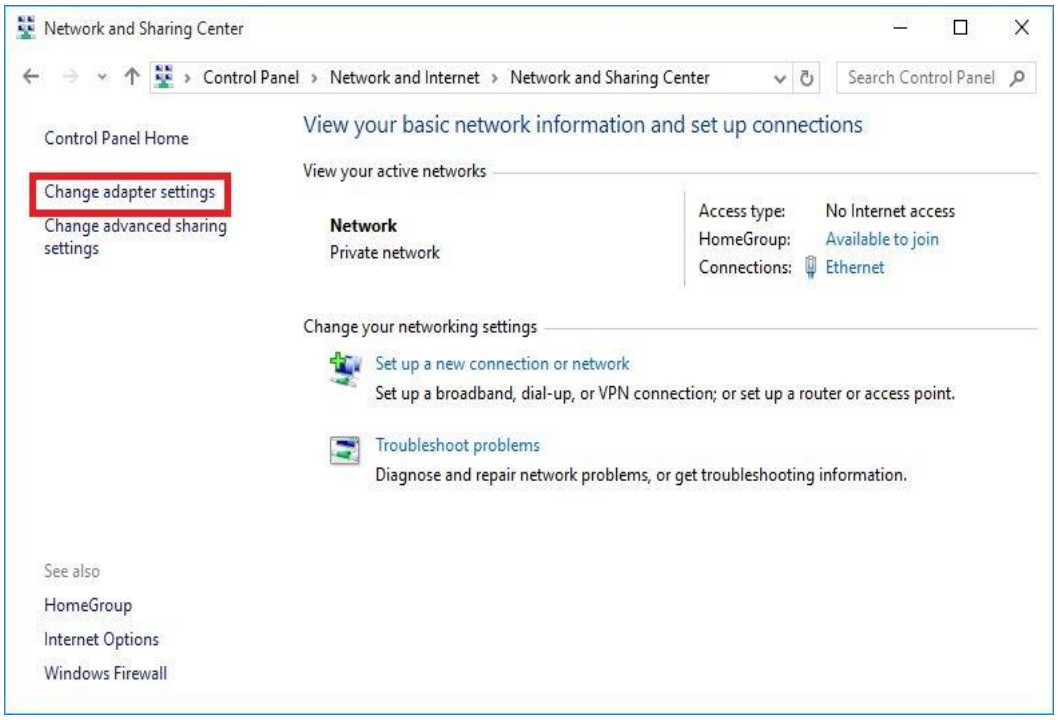
In order to perform this test, the evaluated platforms should have a Bluetooth interface. Since not all of the evaluated platforms have this kind of hardware, this test is not applicable for the following evaluated platform:

- Dell 755 Optiplex with Windows 10 x86 Pro Edition.
- Dell 755 Optiplex with Windows 10 x86 Enterprise Edition.
- Dell 755 Optiplex with Windows Server 2012 R2 Datacenter Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.

27.3.13.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.
2. Go to the *Network and Internet->Network and Sharing Center* in the *Control Panel*. Click-in *Change adapter settings* link to open the *Network Connections* window.



3. Right-click the *Bluetooth Network Connection* adapter and select "*Enable*" or "*Disable*" to modify the enabled state.

4. Start PowerShell and run the following command and observe the result: *Get-NetAdapter.*

5. After that, log out and log in using a user without administrator rights.

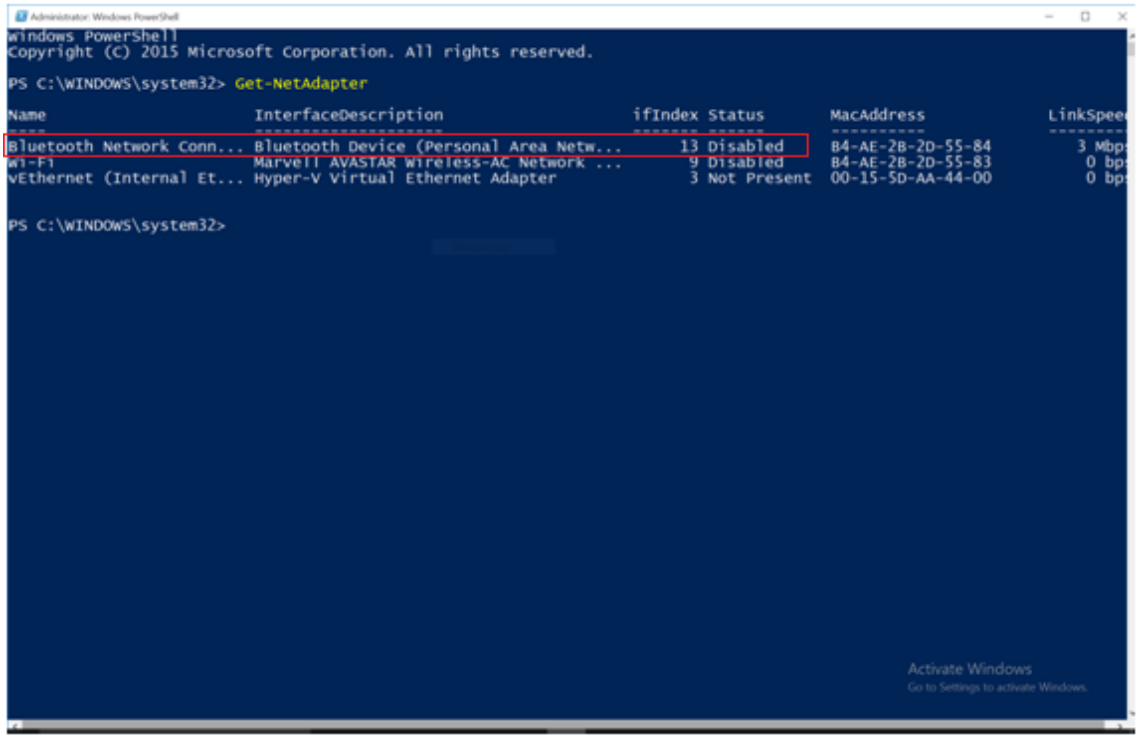
6. Repeat the steps 2-5.

27.3.13.3. Results

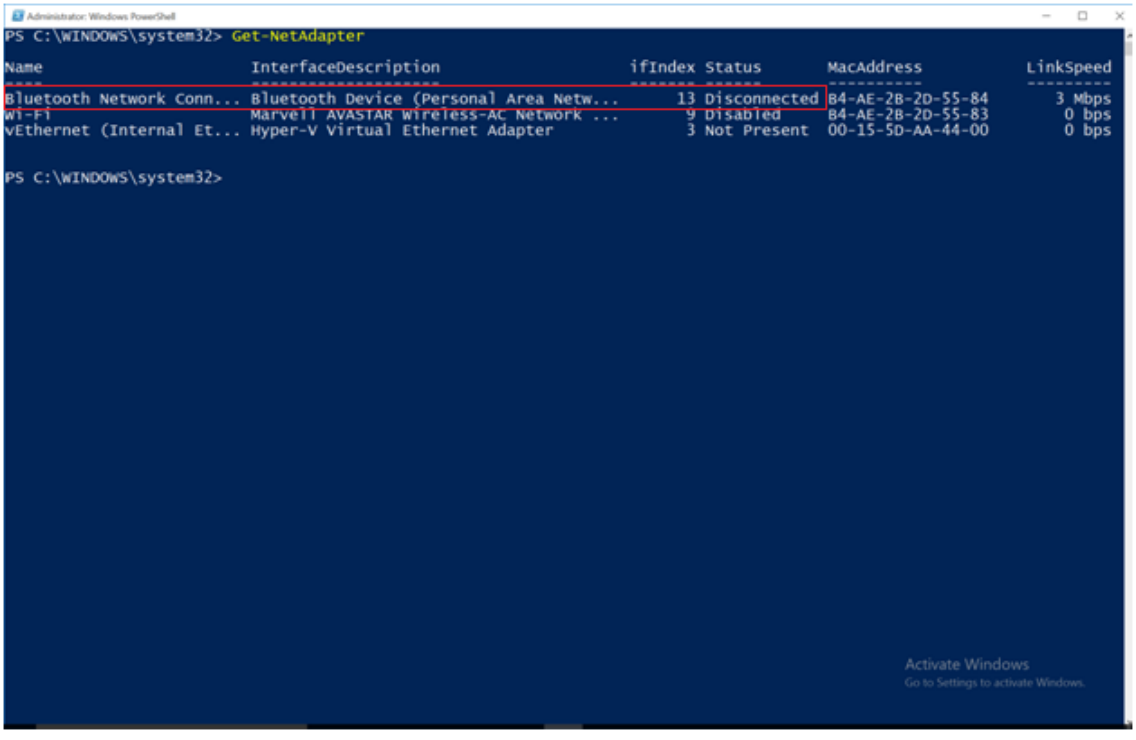
This test can only be performed in those evaluated platforms which have Bluetooth interface. This test is not applicable for Dell platforms and those platforms running over Windows Server 2012 Hyper-V. Due to this, the evaluator has performed this test in the following evaluated platform:

- Surface 3 with Windows 10 x64 Enterprise Edition

The evaluator has obtained the following evidence using a user with administrator rights. After disabling the Bluetooth interface the state of the interface is as follows:

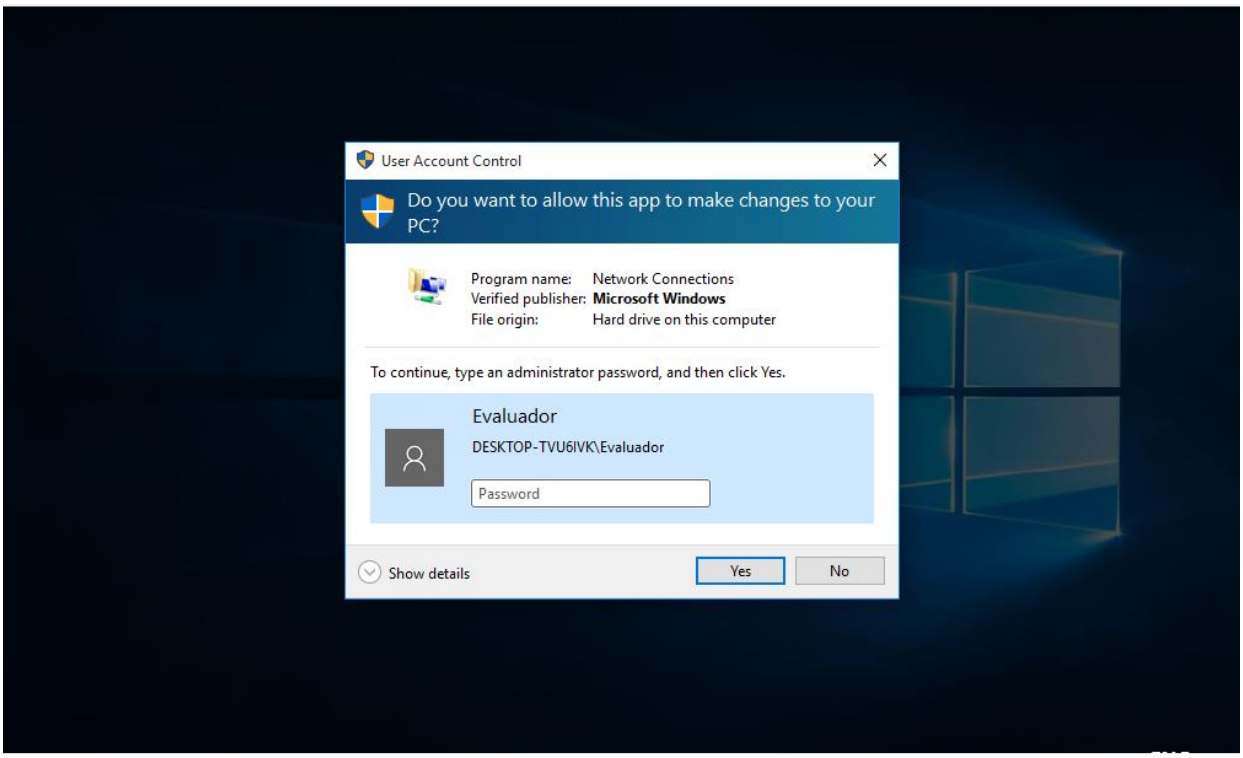


Once the Bluetooth interface is enabled again, the state is shown as follows.



As it can be observed in the images above, the change has been applied successfully and the evaluator has checked that the state of Bluetooth interface has been modified.

The evaluator has attempted to modify the state of Bluetooth interface using a user without administrator rights, and the following prompt is shown asking for the administrator password.



27.3.13.4. Verdict

As the above results state, a Bluetooth interface can only be enabled or disabled using a user with administrator rights. When a user without administrator rights is used, a prompt is shown asking for the administrator password.

Due to this, the evaluator considers that, the behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Enable/disable Bluetooth** management function.

27.3.14. Configure USB interfaces

27.3.14.1. Setup

The applicable setup for this test is the same as the one defined for *Configure minimum password length* test.

27.3.14.2. Procedure

To configure the USB interfaces, the evaluator shall modify the *DenyRemovableDevices* key in the Windows Registry. This modification can be performed in two different ways, manually or automatically.

In order to do this test using the manually method, the evaluator shall open *Local Group Policy Editor* and go to *Computer Configuration -> Administrative templates -> System -> Device Installation Restrictions* and configure the value for *Prevent installation of removable devices* key.

In order to make easier the test execution, the evaluator has developed a script, which directly modifies the key in the Windows Registry. This method shall be the used one in this procedure. The script source code is as follows:

```
auditpol /set /subcategory:"Registry" /success:enable /failure:enable

$pathFolder = "Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\DeviceInstall\Restrictions"
$keyName = "DenyRemovableDevices"
$key = Get-ItemProperty -Path $pathFolder -Name $keyName -ErrorAction SilentlyContinue

If ($key -eq $null){
    New-Item -Path Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows -Name DeviceInstall -ItemType Folder
    New-Item -Path Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\DeviceInstall -Name Restrictions -ItemType Folder
    New-ItemProperty -Path $pathFolder -Name $keyName -Value "1" -PropertyType DWORD -Force
}else{
    Set-ItemProperty -Path $pathFolder -Name $keyName -Value "0"
}

Start-Sleep -s 1
Get-EventLog Security -InstanceId 4657 -Newest 1 | fl *
auditpol /set /subcategory:"Registry" /success:disable /failure:disable
```

The evaluator shall carry out the following steps in order to perform this test.

1. Log in as administrator in the machine.
2. Open a PowerShell terminal as administrator and run command "*regedit.exe*" and in Registry Editor select the folder *HKEY_LOCAL_MACHINE -> SOFTWARE -> Policies -> Microsoft -> Windows.*
3. After that, right-click and select *Permissions*… in the context menu to open the *Permissions* dialog.
4. Click the Advanced button to open the *Advanced Security Settings* dialog, click in the *Auditing* tab and click the *Add* button to open the *Auditing Entry* dialog.
5. Click the Select a principal to open the Select User or Group dialog, type "*Users*", click Check Names and click the OK button.
6. Select Type: *All.*
7. Select Applies to: *This key and sub keys*.
8. Click Show advanced permissions and click *Set Value*.

9. Click *OK* and *Apply*.

10. After that, open a PowerShell terminal as administrator, run the script *test21ConfigureUSBinterface.ps1* and observe the result.

11. Log out and log in again, but this time using the local user account without administrator rights.

12. Open a PowerShell terminal and run "*gpedit.msc*".

13. Observe the result.

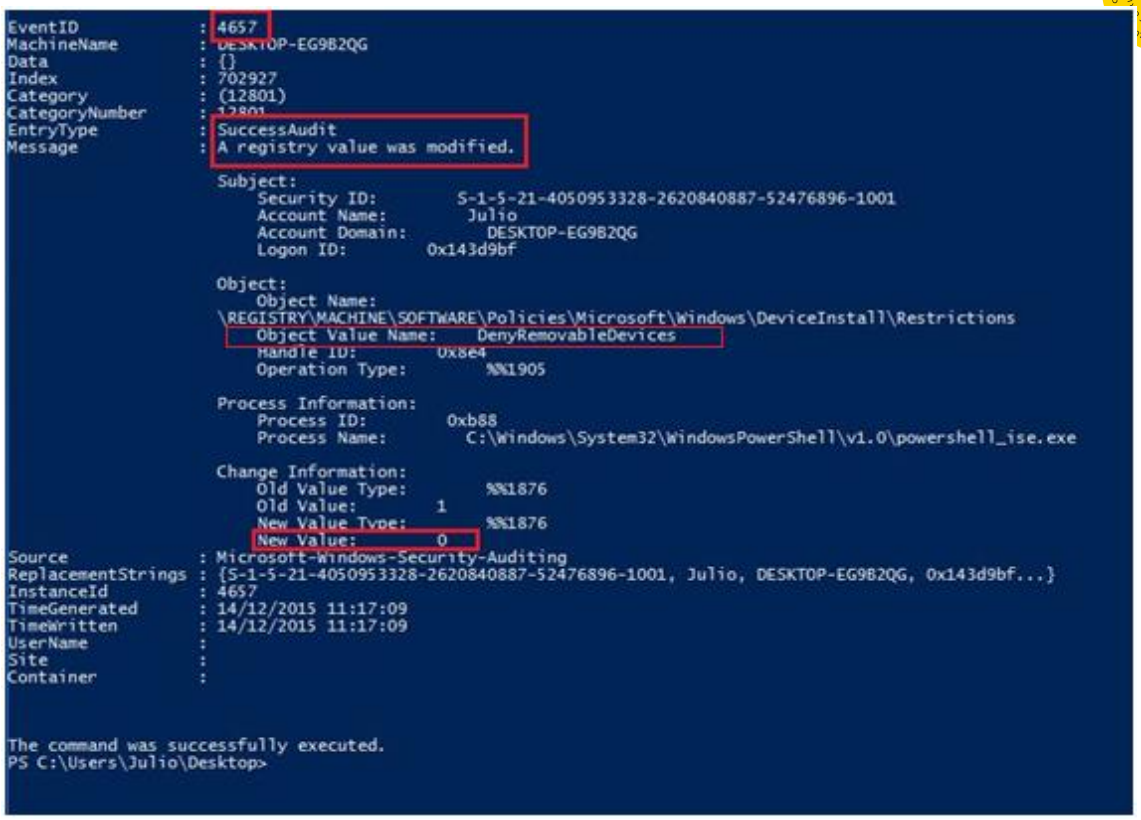
27.3.14.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

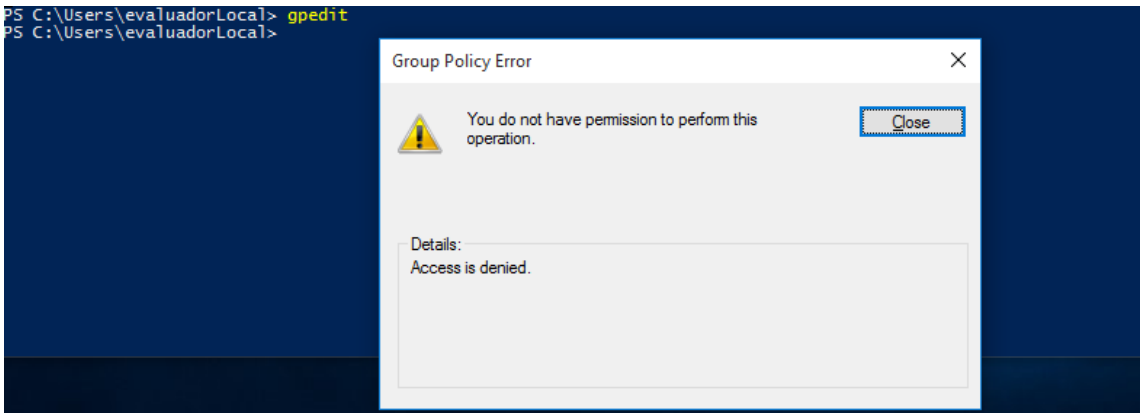
The evaluator has obtained the same results for all the tested platforms. The evaluator has executed the script using both local user account and administrator account, and the obtained result is as follows:

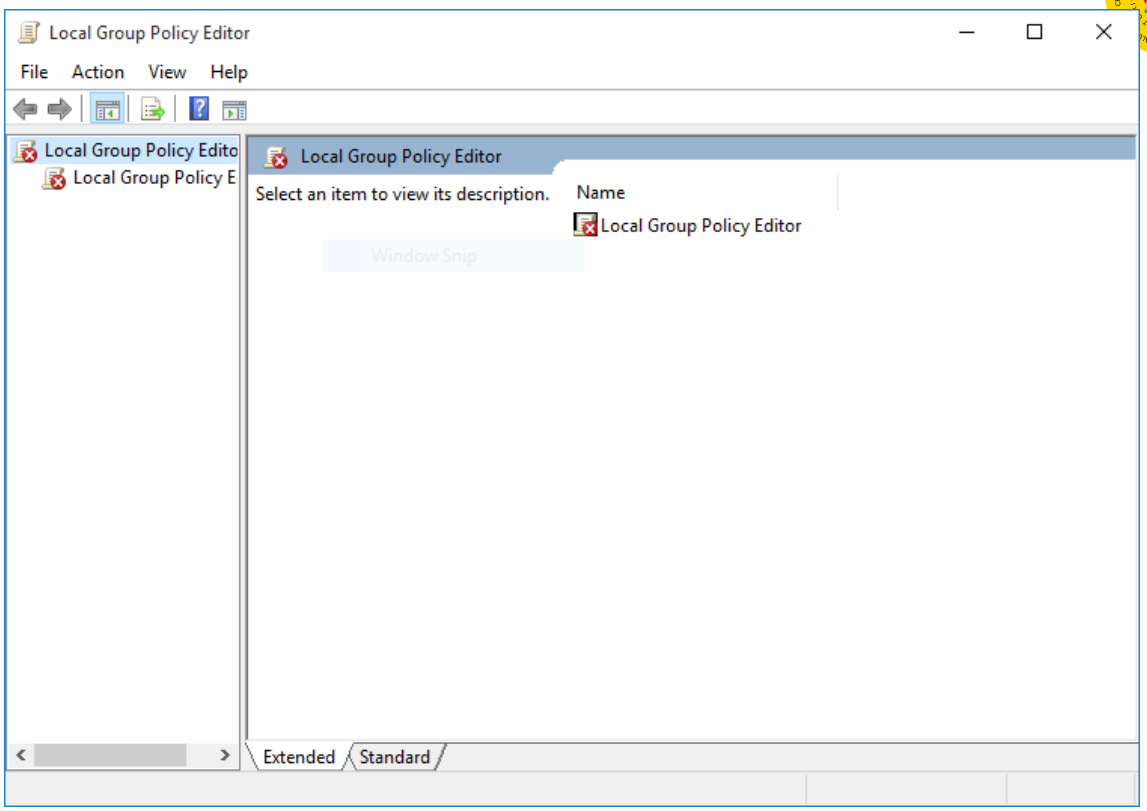
- Administrator account: The script has been executed properly and the following audit entry has been shown.



As it can be observed in the image above, the change has been applied successfully and the new value is shown.

- User account: The evaluator has obtained the following result when has attempted to open *Local Group Policy Editor*:





In addition, user without administrator right cannot execute the script because this kind of accounts has not permissions to modify registry keys.

27.3.14.4. 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 behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Configure USB interfaces** management function

27.3.15. Enable/disable [Local area Network interface]

27.3.15.1. Setup

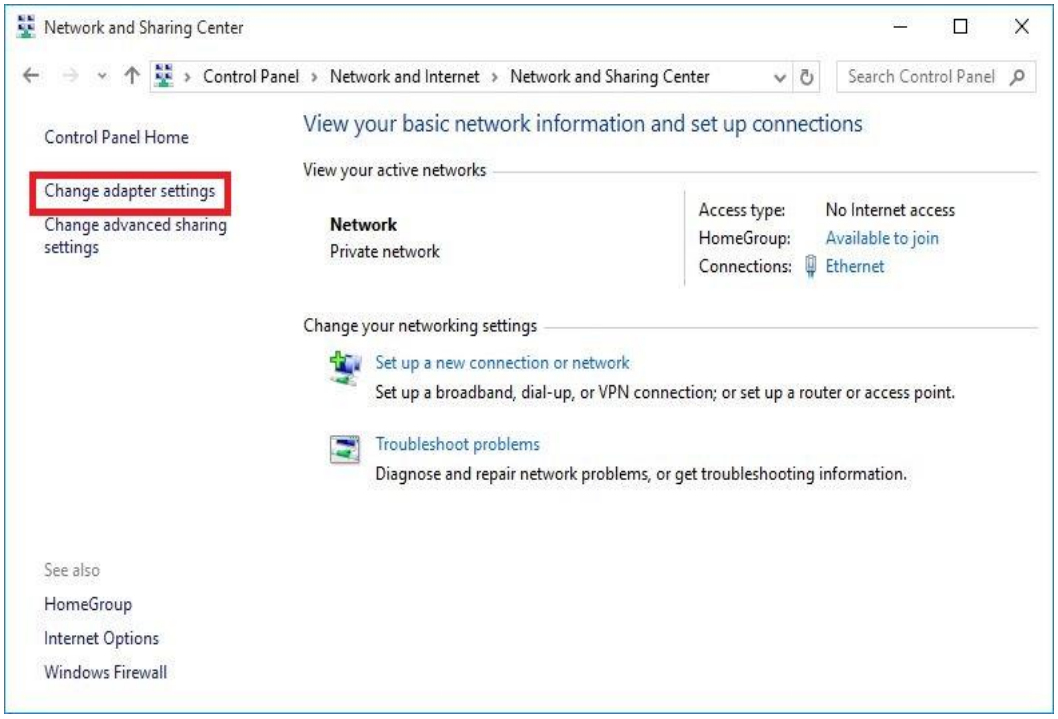
The applicable setup for this test is the same as the one defined for *Configure minimum password length test.*

27.3.15.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Log in using the administrator account.

2. Go to the *Network and Internet -> Network and Sharing Center* in the Control Panel. Click-in "*Change adapter settings*" link to open the Network Connections window.



3. Right-click the Local Area Network adapter and select "*Enable*" or "*Disable*" to modify the enabled state.

4. Open a PowerShell terminal, run the following command and observe the results: *Get-NetAdapter.*

5. Log out, and log in again, but this time using the user account without administrator rights.

6. Repeat the steps 2-5 and observe the results.

27.3.15.3. Results

The evaluator has performed the test in the following evaluated platforms:

- Surface 3 with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has obtained the same results for all the tested platforms. The evaluator has obtained the following evidence using a user with administrator rights. After disabling the local area network interface the state of the interface is as follows:

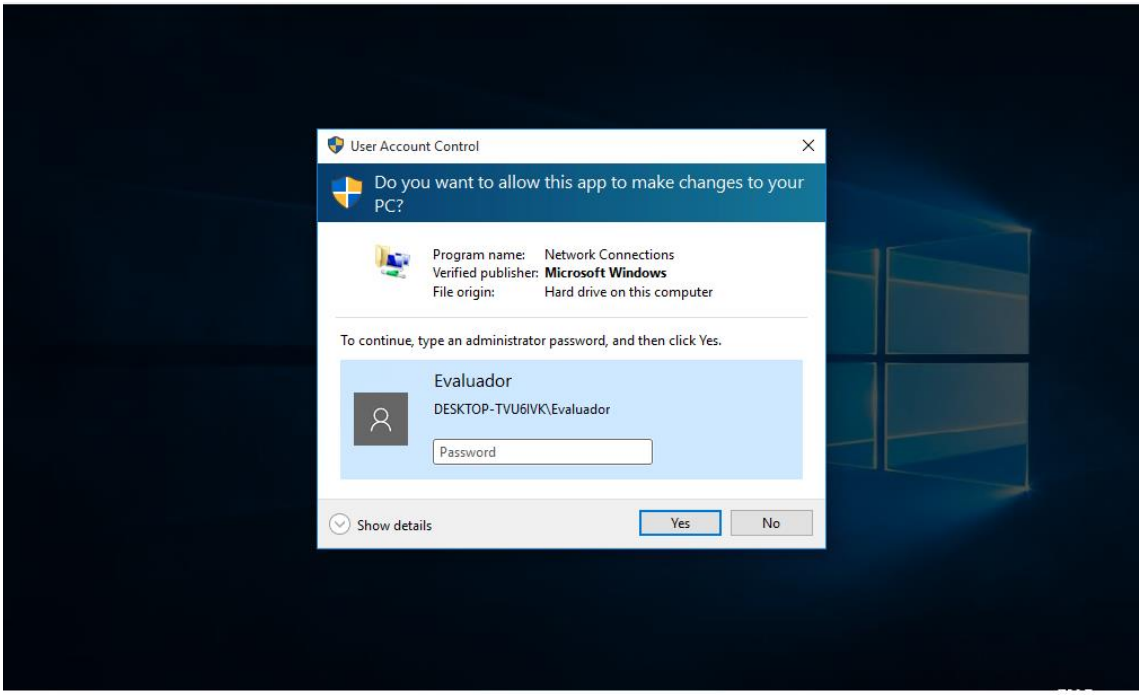
```
PS C:\Windows\system32> Get-NetAdapter

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Ethernet                  Realtek PCIe GBE Family Controller            4 Disabled     F4-6D-04-D8-80-C1      100 Mbps
VMware Network Adapte...8 VMware Virtual Ethernet Adapter for ...       6 Up           00-50-56-C0-00-08      100 Mbps
VMware Network Adapte...1 VMware Virtual Ethernet Adapter for ...       3 Up           00-50-56-C0-00-01      100 Mbps
```

Once the Ethernet interface is enabled again, the state is shown as follows.

```
PS C:\Windows\system32> Get-NetAdapter

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Ethernet                  Realtek PCIe GBE Family Controller            4 Up           F4-6D-04-D8-80-C1      100 Mbps
VMware Network Adapte...8 VMware Virtual Ethernet Adapter for ...       6 Up           00-50-56-C0-00-08      100 Mbps
VMware Network Adapte...1 VMware Virtual Ethernet Adapter for ...       3 Up           00-50-56-C0-00-01      100 Mbps
```

As it can be observed in the images above, the change has been applied successfully and the evaluator has checked that the state of local area network interface has been modified.

The evaluator has attempted to modify the state of the local area network interface using a user without administrator rights, and the following prompt is shown asking for the administrator password.



27.3.15.4. Verdict

As state the above results, the Local area Network interface can only be configured by a user with administrator rights.

Due to this, the evaluator considers that, the behavior and results obtained during this test activity match with the selection made in the security target. Therefore, the **PASS** verdict is assigned to **Enable/disable [Local area network interface]** management function.

27.4. Final Verdict

Due to the documentation review activity and all test activities 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 FMT_MOF_EXT.1.1.

28. FPT_ACF_EXT.1.1

28.1. Assurance activity

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 all 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.

28.2. Documentation review activity

28.2.1. Findings

The evaluator has reviewed the information provided in TSS, section 6.6.2 **Protection of OS Binaries, Audit and Configuration Data**. This information states that kernel 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.

Moreover, the permissions over these kinds of files are also explained. Standard users have permissions to read and execute kernel drivers, system executable and libraries, and they are not authorized to access audit logs and configuration files. On the other hand, administrator

users have permissions to write and modify kernel drivers, system executable and libraries, and full control over audit logs and configuration files.

28.2.2. Verdict

The evaluator considers that the TSS provides enough information related to where kernel drivers, system executables, libraries, configuration files, and audit logs are stored in the system and how the permissions are applied over these files.

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.

28.3. Test Activity

28.3.1. Test 1

28.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- User account with user name *user1* shall exist. This user shall belong to the default Users group. Password for this account must be *p@ss1234*.
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

28.3.1.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

9. Logon using a user account without administrator rights, e.g. user1 account.
10. Run the script FPT_ACF_EXT.1.1 - Test 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: ".\FPT_ACF_EXT.1.1 - Test 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 "`n------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{
        Add-Content -Path $driver.FullName -Encoding Byte -Value ([Byte] 0xAA) -ErrorAction Stop
    } catch [Exception]{
        Write-Warning $Error[0].Exception.Message
    }
}
```

11. Observe the results.

28.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

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

The evaluator has obtained the same message for each file which has been attempted to modify. The message stated that the access is denied. The following screenshot shows a fragment of the obtained result:

```
PS C:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.1> & '.\FPT_ACF_EXT.1.1 - Test 1.ps1'

------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\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\HTTPERR' 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\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\wfp' 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\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\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.
```

28.3.1.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 1** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 1**.

28.3.2. Test 2

28.3.2.1. Setup

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

28.3.2.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Logon using a user account without administrator rights, e.g. user1 account.
2. Run the script FPT_ACF_EXT.1.1 - Test 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: ".\FPT_ACF_EXT.1.1 - Test 2.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 "`n------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
    }
}
```

3. Observe the results.

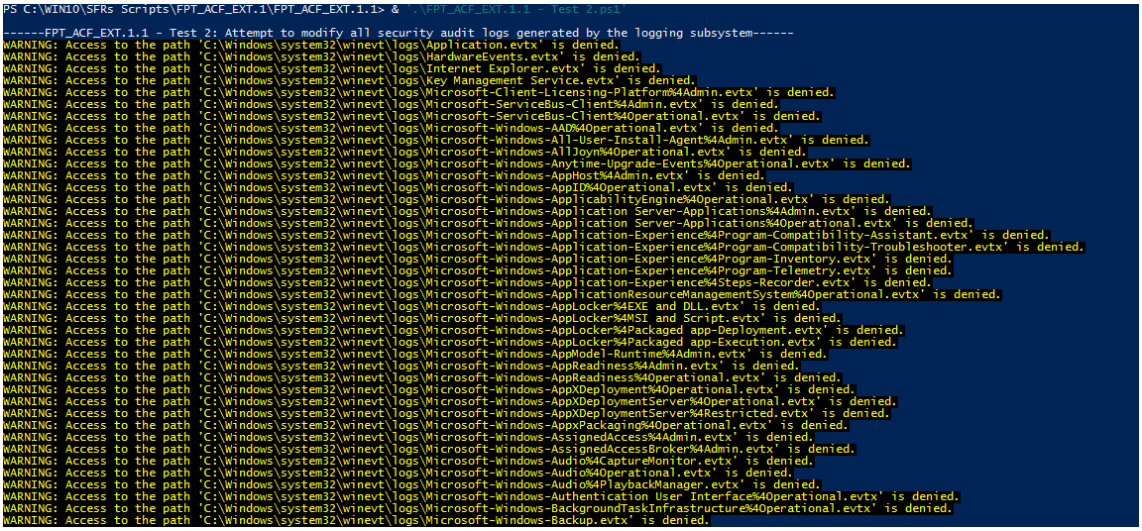
28.3.2.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

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

The evaluator has obtained the same message for each file which has been attempted to modify. The message stated that the access is denied. The following screenshot shows a fragment of the obtained result:



28.3.2.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 2** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 2**.

28.3.3. Test 3

28.3.3.1. Setup

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

28.3.3.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Logon using a user account without administrator rights, e.g. user1 account.
2. Run the script FPT_ACF_EXT.1.1 - Test 3.ps1. To execute the script, open a PowerShell terminal, go to the path location where the script are stored and type the following command: ".\FPT_ACF_EXT.1.1 - Test 3.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 "`n------FPT_ACF_EXT.1.1 - Test 3: Attempt to modify all shared libraries thath 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
    }
}
```

3. Observe the results.

28.3.3.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

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

The evaluator has obtained the same message for each file which has been attempted to modify. The message stated that the access is denied. The following screenshot shows a fragment of the obtained result:

```
PS C:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.1> & '.\FPT_ACF_EXT.1.1 - Test 3.ps1'

------FPT_ACF_EXT.1.1 - Test 3: Attempt to modify all shared libraries thath are used throughout the system------
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\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\HTTPERR' 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\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\wfp' is denied.
WARNING: Access to the path 'C:\Windows\system32\aadauthhelper.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\aadcloudap.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\aadtb.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\AboveLockAppHost.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\accessibilitycpl.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\accountaccessor.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\AccountsControlInternal.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\AccountsRt.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ACCTRES.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\acledit.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\aclui.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\acmigration.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ACPBackgroundManagerPolicy.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\acppage.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\acproxy.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ActionCenter.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ActionCenterCPL.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ActionQueue.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ActivationClient.dll' is denied.
WARNING: Access to the path 'C:\Windows\system32\ActivationManager.dll' is denied.
```

28.3.3.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 3** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 3**.

28.3.4. Test 4

28.3.4.1. Setup

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

28.3.4.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Logon using a user account without administrator rights, e.g. user1 account.
2. Run the script FPT_ACF_EXT.1.1 - Test 4.ps1. To execute the script, open a PowerShell terminal, go to the path location where the script are stored and type the following command: ".\FPT_ACF_EXT.1.1 - Test 4.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 "`n------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
    }
}
```

3. Observe the results.

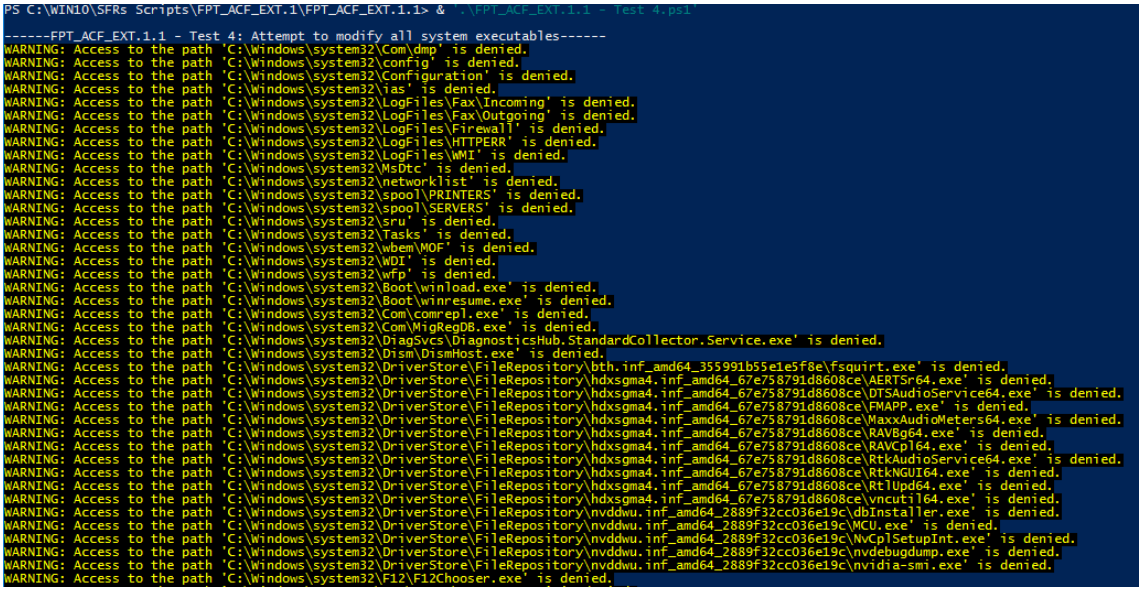
28.3.4.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

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

The evaluator has obtained the same message for each file which has been attempted to modify. The message stated that the access is denied. The following screenshot shows a fragment of the obtained result:



28.3.4.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 4** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 4**.

28.3.5. Test 5

28.3.5.1. Setup

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

28.3.5.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

1. Logon using a user account without administrator rights, e.g. user1 account.
2. Run the script FPT_ACF_EXT.1.1 - Test 5.ps1. To execute the script, open a PowerShell terminal, go to the path location where the script are stored and type the following command: ".\FPT_ACF_EXT.1.1 - Test 5.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 "`n------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
        #Add-Content -Path $driver.FullName -Encoding Byte -Value ([Byte] 0xAA) -ErrorAction Stop
    } catch [Exception]{
        Write-Warning $Error[0].Exception.Message
    }
}
```

3. Observe the results.

28.3.5.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same result for all the tested platforms. Users without administrator privileges cannot access to system32\config folder. The evaluator has obtained the following message during the test execution:

```
PS C:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.1> & '.\FPT_ACF_EXT.1.1 - Test 5.ps1'

------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:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.1>
```

28.3.5.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 5** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 5**.

28.3.6. Test 6

This test is not applicable in this evaluation due to the assignment made (**None**) in the security target.

28.4. Final Verdict

Due to both documentation review activity and test activities 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 FPT_ACF_EXT.1.1.

29. FPT_ACF_EXT.1.2

29.1. Assurance activity

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.

29.2. Documentation review activity

There is no specific documentation review activity related to FPT_ACF_EXT.1.2. To see information about the FPT_ACF_EXT.1 documentation review activity, please check the FPT_ACF_EXT.1.1 report.

29.3. Test Activity

29.3.1. Test 1

29.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- User account with user name *user1* shall exist. This user shall belong to the default Users group. Password for this account must be *p@ss1234*.
- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

29.3.1.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

12. Logon using a user account without administrator rights, e.g. user1 account.

13. Run the script FPT_ACF_EXT.1.2 - Test 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: ".\FPT_ACF_EXT.1.2 - Test 1.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 "`n------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
    }
}
```

14. Observe the results.

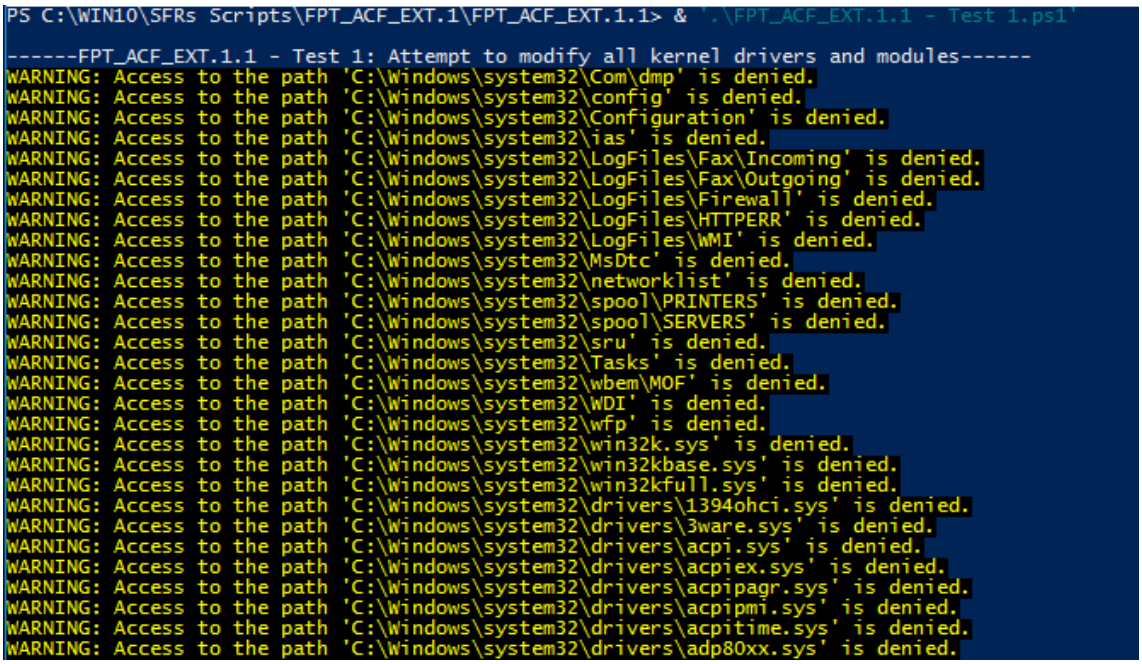
29.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

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

The evaluator has obtained the same message for each file which has been attempted to read. The message stated that the access is denied. The following screenshot shows a fragment of the obtained result:



29.3.1.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 1** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 1**.

29.3.2. Test 2

29.3.2.1. Setup

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

29.3.2.2. Procedure

In order to perform this test, the evaluator has followed the steps defined below:

4. Logon using a user account without administrator rights, e.g. user1 account.

5. Run the script FPT_ACF_EXT.1.2 - Test 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: ".\FPT_ACF_EXT.1.2 - Test 2.ps1". 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 "`n------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
}
```

6. Observe the results.

29.3.2.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same result for all the tested platforms. Users without administrator privileges cannot access to read the value stored in HKLM\SAM\SAM. The evaluator has obtained the following message during the test execution:

```
PS C:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.2> & '.\FPT_ACF_EXT.1.2 - Test 2.ps1'

------FPT_ACF_EXT.1.2 - Test 2: Attempt to read system.wide credential repositories------
WARNING: Requested registry access is not allowed.
PS C:\WIN10\SFRs Scripts\FPT_ACF_EXT.1\FPT_ACF_EXT.1.2>
```

29.3.2.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 2** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 2**.

29.3.3. Test 3

This test is not applicable in this evaluation due to the assignment made (**None**) in the security target.

29.4. Final Verdict

Due to all test activities 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 FPT_ACF_EXT.1.2.

30. FPT_ASLR_EXT.1.1

30.1. Assurance activity

The evaluator will select 3 executables included with the TSF. These must include any web browser or mail client included with the TSF. 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.

30.2. Documentation review activity

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

30.3. Test Activity

30.3.1. Test 1

30.3.1.1. Setup

VMMap tool (a sysinternal utility that provides the ability to analysis the physical memory) is available.

The three executables, which shall be used during the test execution and included with the TSF, are the following:

- Command Prompt (cmd.exe)
- Internet Explorer 11 (iexplore.exe)
- Mail App (Trusted Windows Store App, its process name is HxMail.exe). Due to this application is not available in Windows Server 2012 R2, Powershell.exe shall be used instead of this application.

30.3.1.2. Procedure

Method 1

The following steps must be performed in order to complete this test assurance activity:

1. Choose one pair of the hardware platforms over which the test must be performed, e.g. Surface Pro 3 with Windows 10 x64 Enterprise Edition.
2. In each one, run one of the executable defined above, e.g. Internet Explorer 11
3. Run the *VMMap* tool and select the process related to Internet Explorer to examine the memory mapping.

4. Observe the results. The memory mappings shall be located in different location of the memory.
5. Repeat the steps 2-4, using the other executables (Mail App and Command Prompt).

Method 2

For those platforms that there is only one available instance, the evaluator shall follow the next procedure:

1. Run one of the executable defined above, e.g. Internet Explorer 11.
2. Run the *VMMap* tool and select the process related to Internet Explorer to examine the memory mapping. Take a screenshot or save the analysis generated by *VMMap*.
3. Restart the TOE and repeat the steps 1-2 in order to execute the second round.
4. Compare the obtained results. The memory mappings shall be located in different location of the memory.
5. Repeat the steps 1-4, using the other executables (Mail App and Command Prompt).

30.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- HP Pro X2 with Windows 10 x64 Pro Edition
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-v with Windows 10 x86 Home Edition

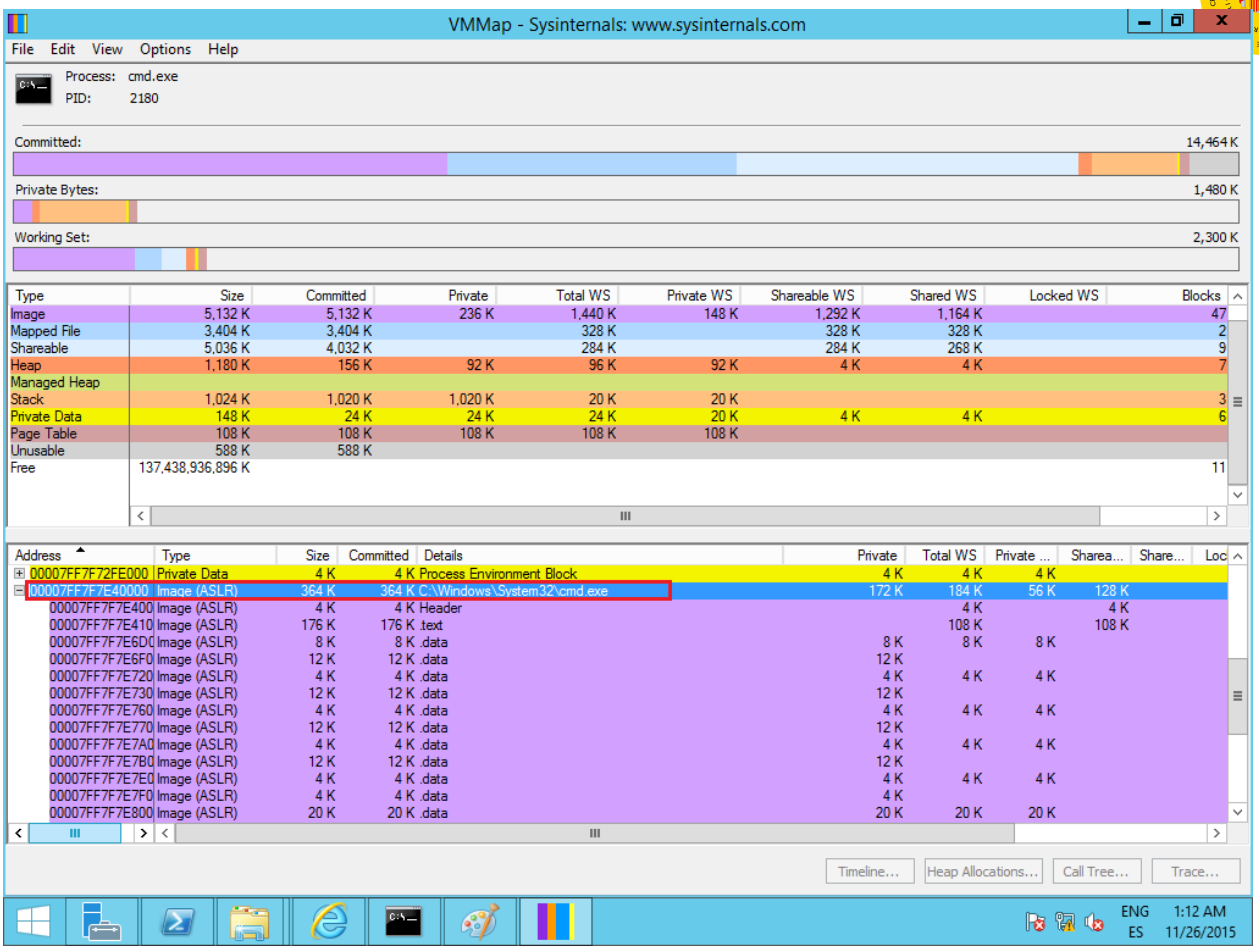
Testing platform:

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition

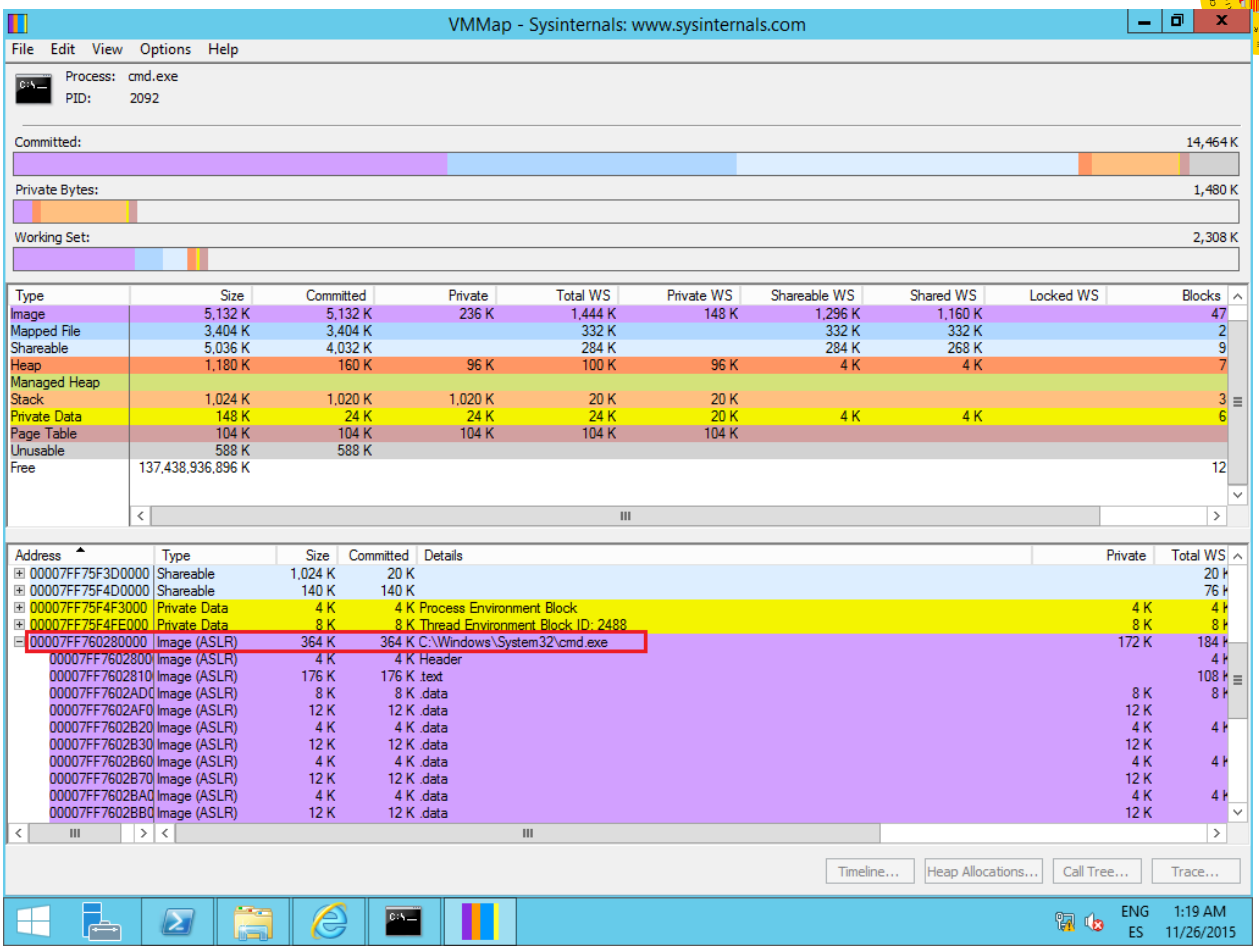
Obtained results:

Due to the fact that there was only one available platform, the evaluator has followed the steps defined in *Procedure* section, *Method 2*. The evaluator has executed one executable file in two different rounds over the same platform, and has observed that the memory address is different from each one. The following images demonstrate this fact:

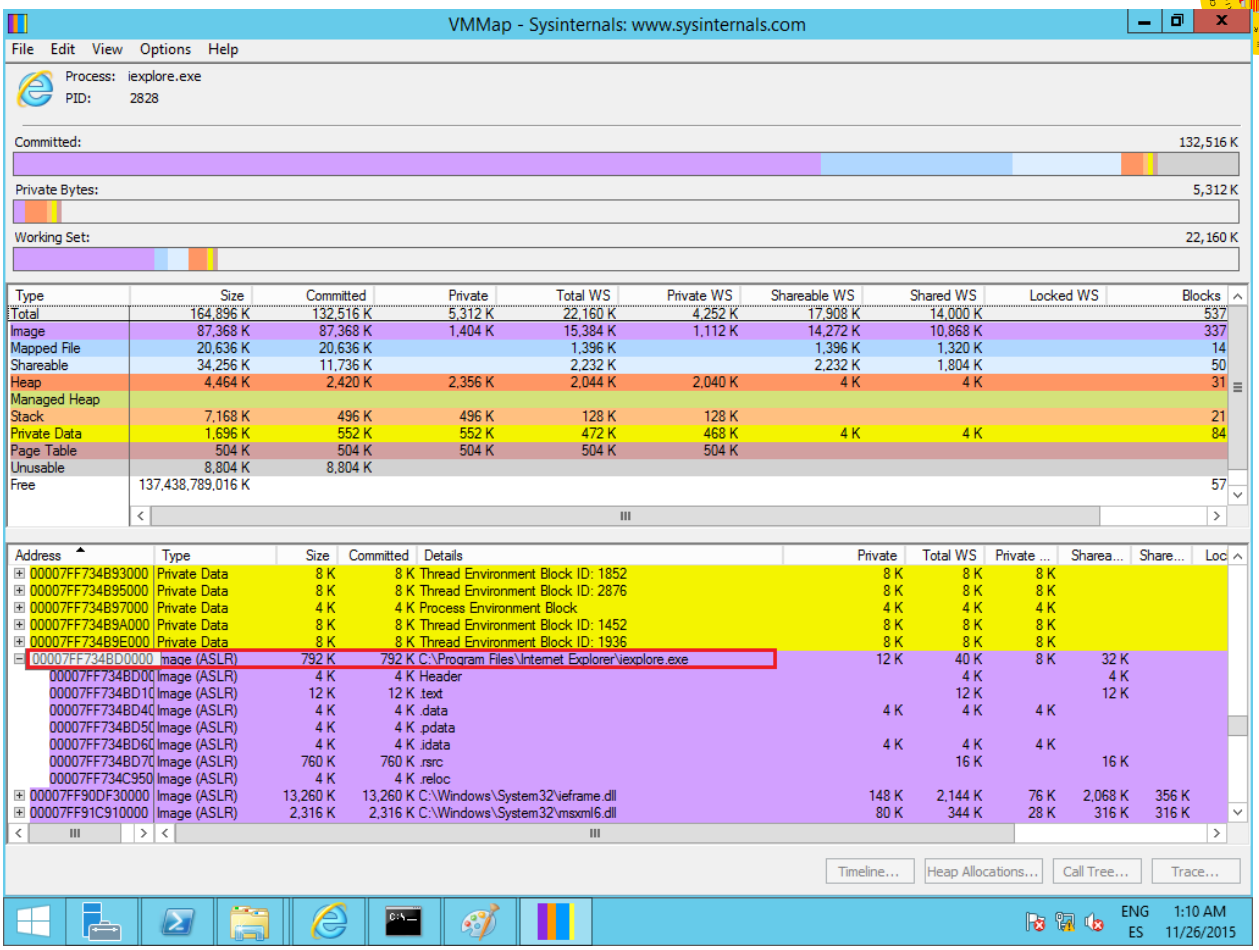
- Process: Command Prompt (cmd.exe)
    - Platform A - Round 1 (Address 0x00007FF7F7E4000):



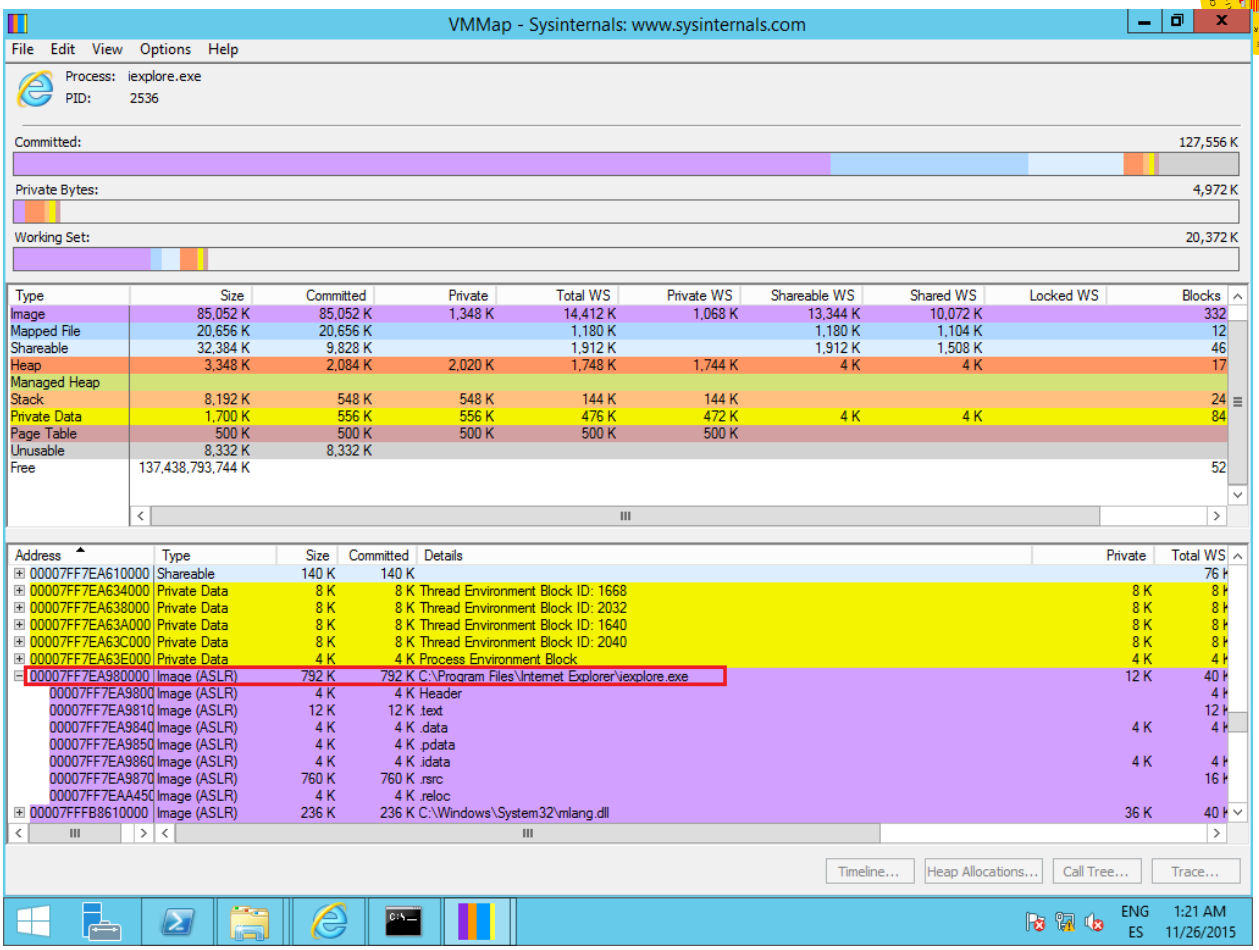
- Platform A - Round 2 (Address 0x00007FF760280000):



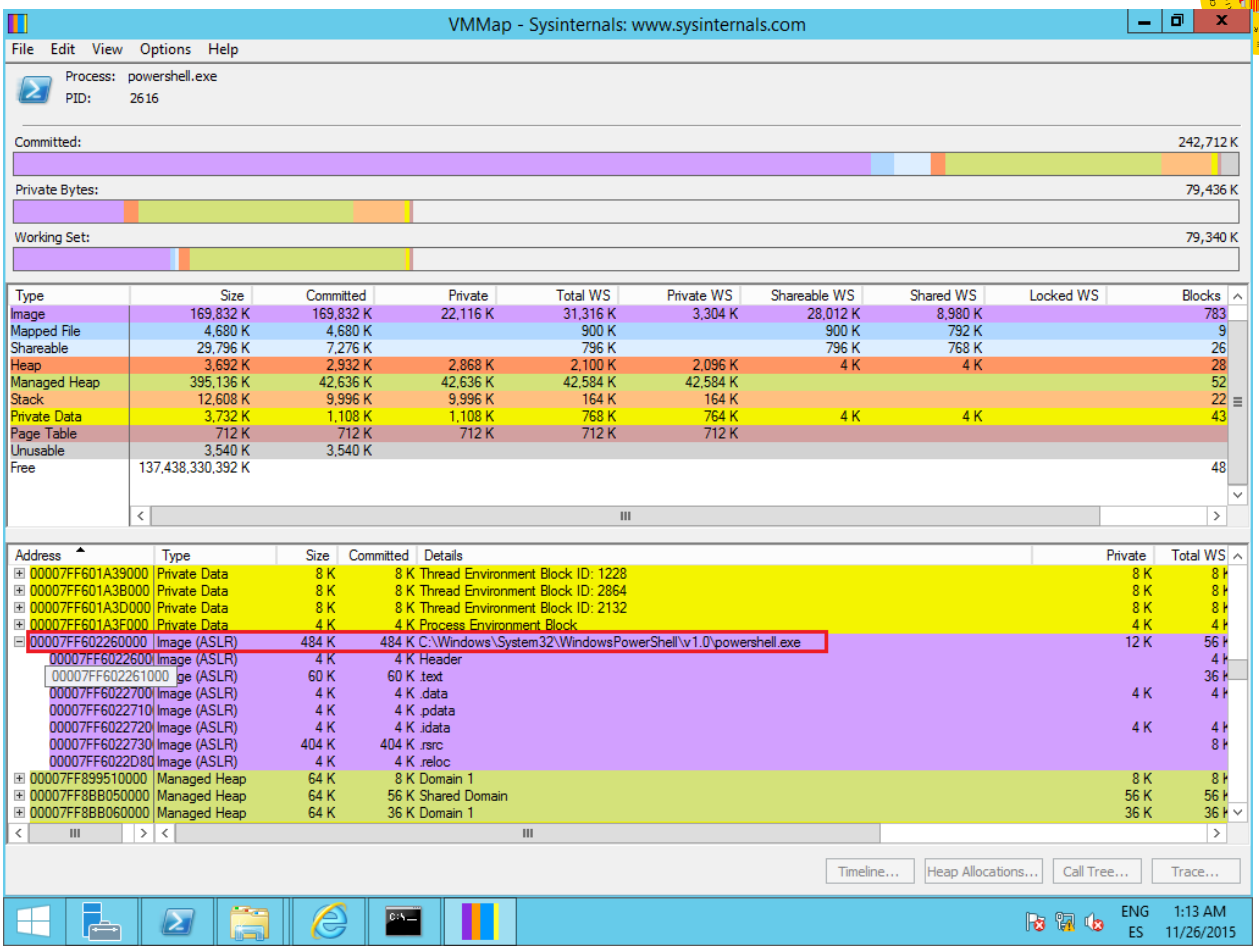
- Process: Internet Explorer 11 (iexplore.exe)
    - Platform A - Round 1 (Address 0x00007FF734BD0000):



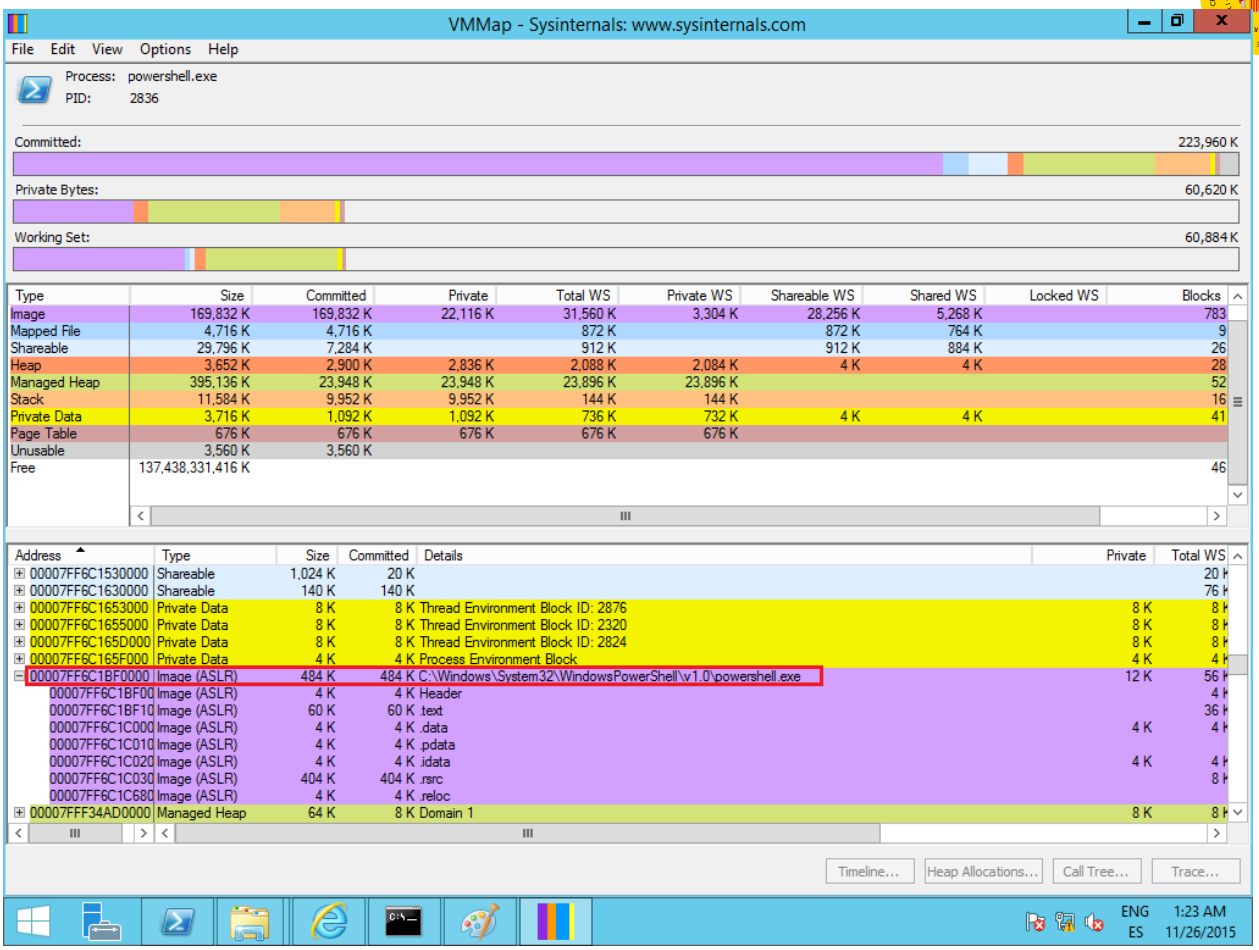
- Platform A - Round 2 (Address 0x00007FF7EA980000):



- Process: PowerShell (powershell.exe)
    - Platform A - Round 1 (Address 0x00007FF602260000):



- Platform A - Round 2 (Address 0x00007FF6C1BF0000):



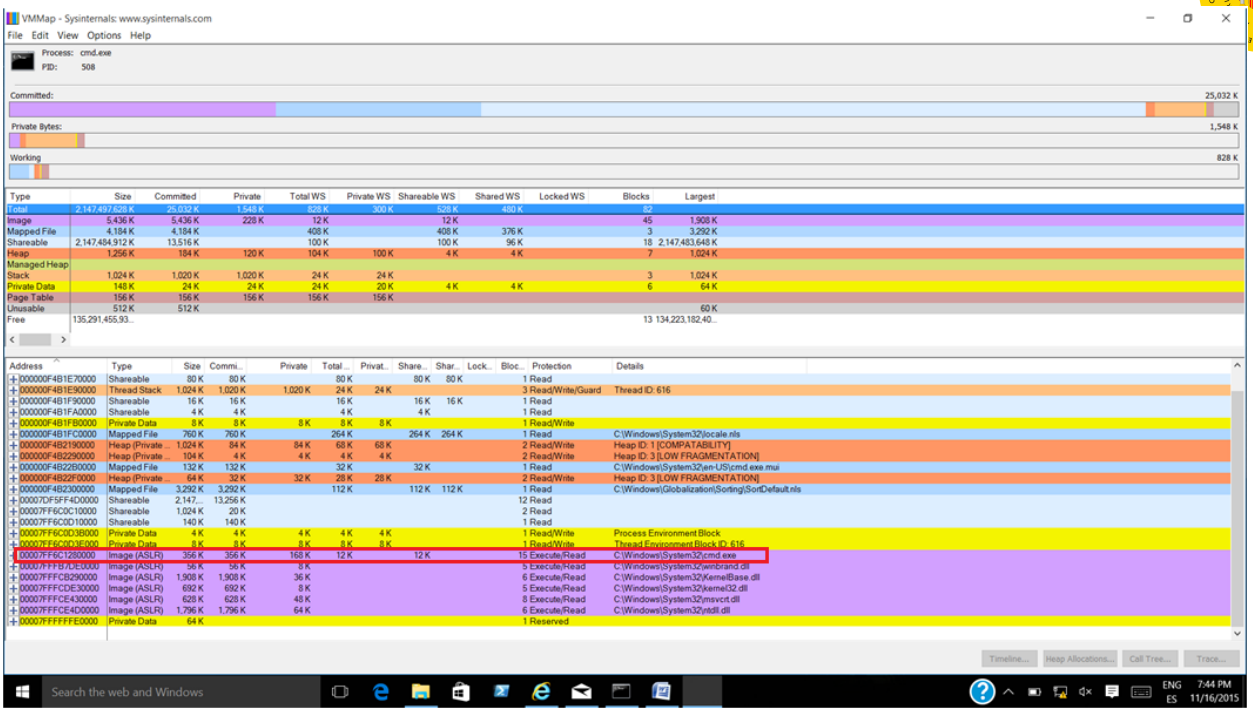
Testing platform:

- HP Pro X2 with Windows 10 x64 Pro Edition

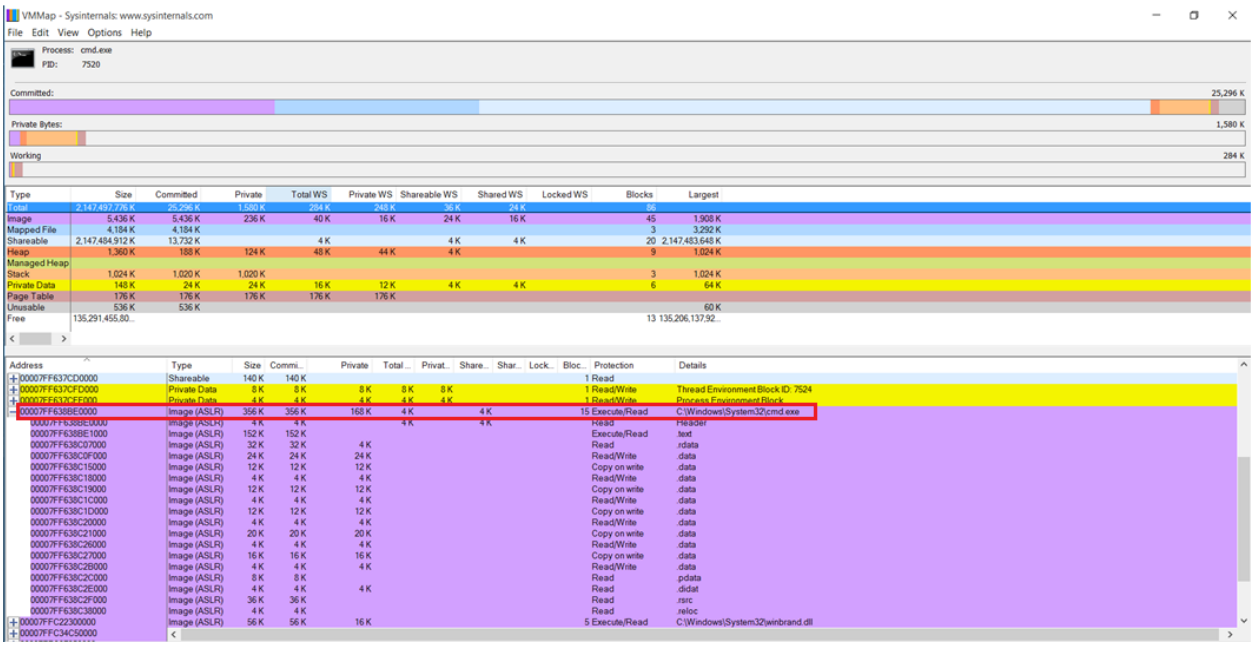
Obtained results:

The evaluator has executed two instances of the same executable file at different platforms, and has observed that the memory address is different from each one. The following images demonstrate this fact:

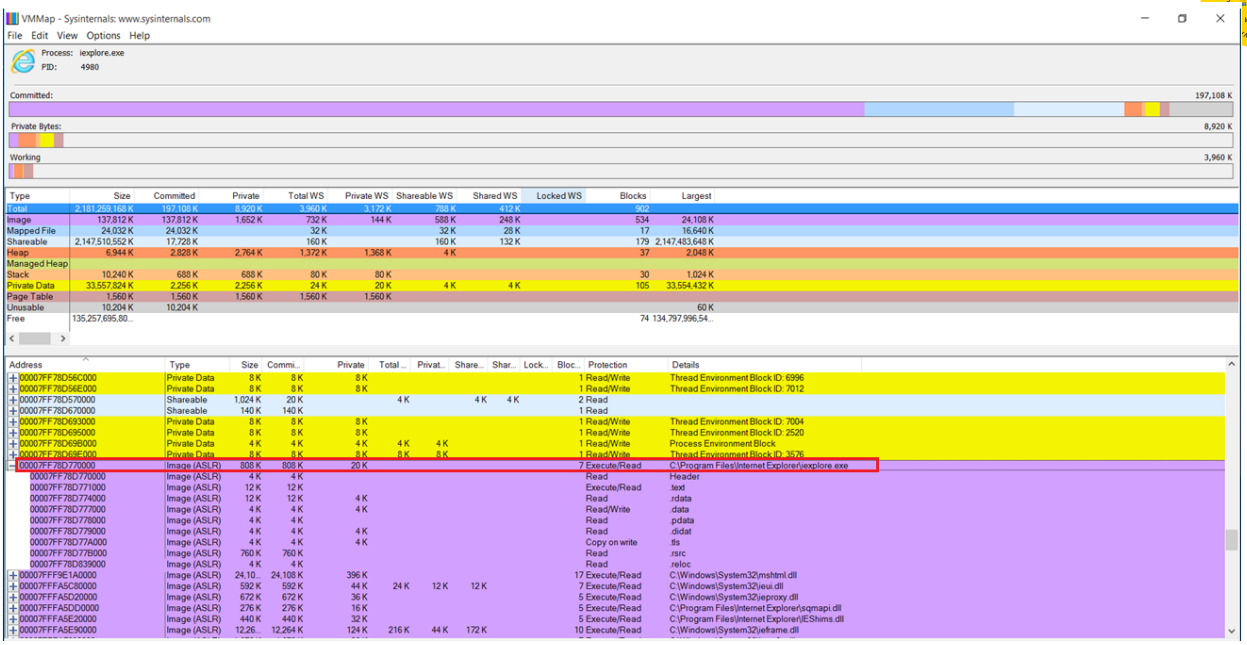
- Process: Command Prompt (cmd.exe)
    - Platform A (Address 0x00007FF6C1280000):



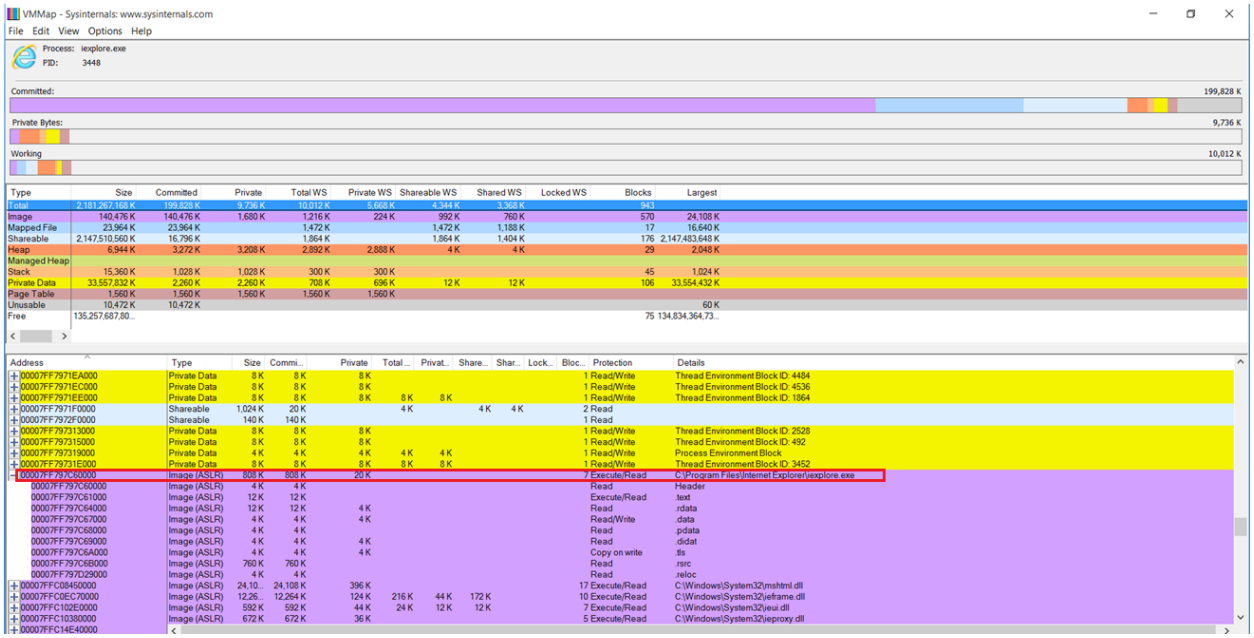
- Platform B (Address 0x00007FF638BE0000):



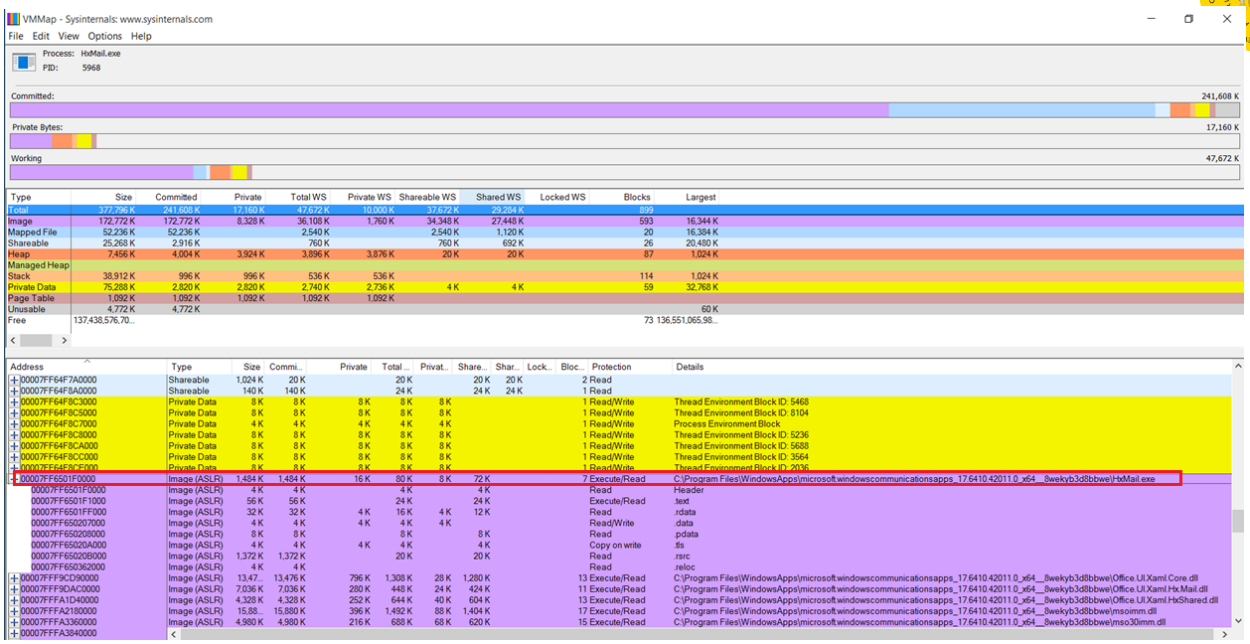
- Process: Internet Explorer 11 (iexplore.exe)
  - Platform A (Address 0x00007FF78D770000):



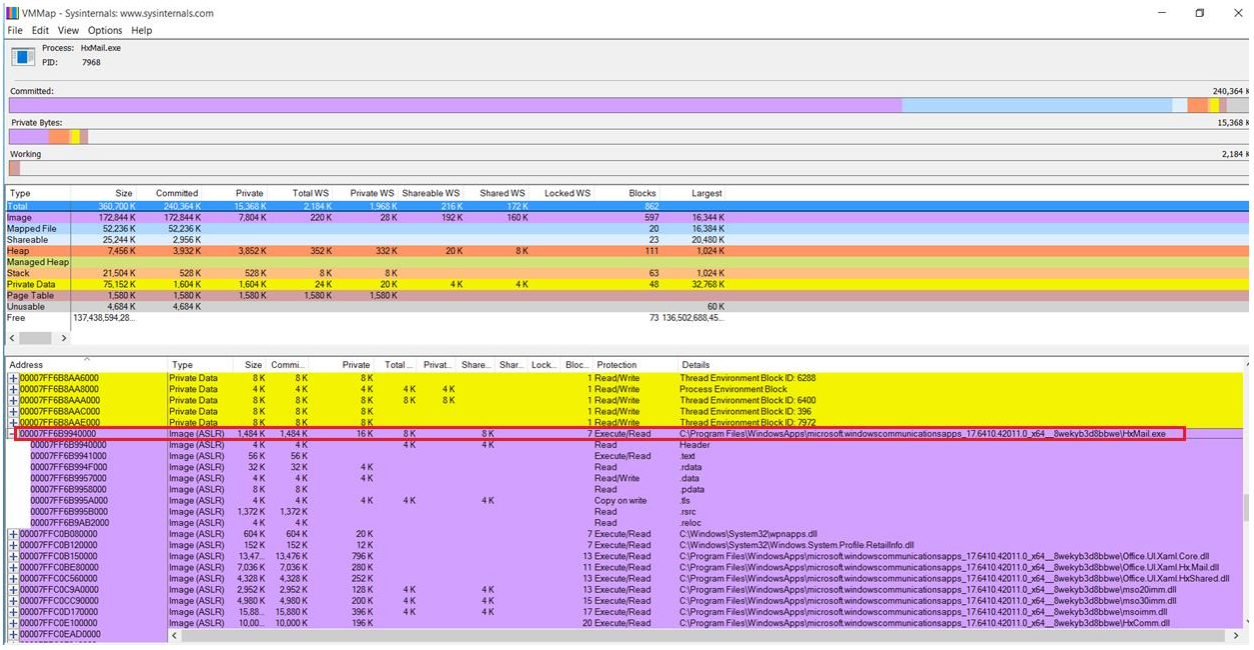
- Platform B (Address 0x00007FF797C60000):



- Process: Mail App (HxMail.exe)
  - Platform A (Address 0x00007FF6501F0000):



- Platform B (Address 0x00007FF6B9940000):



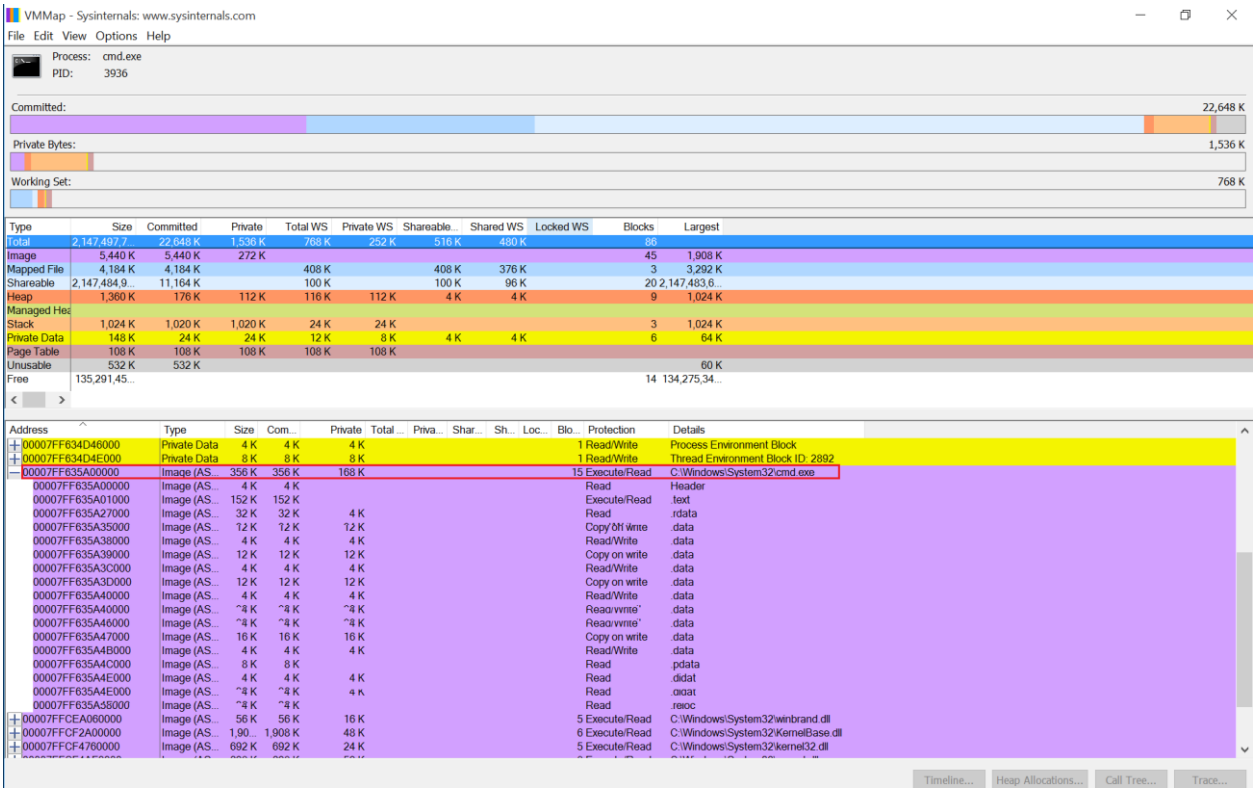
Testing platform:

- Surface 3 with Windows 10 x64 Enterprise Edition

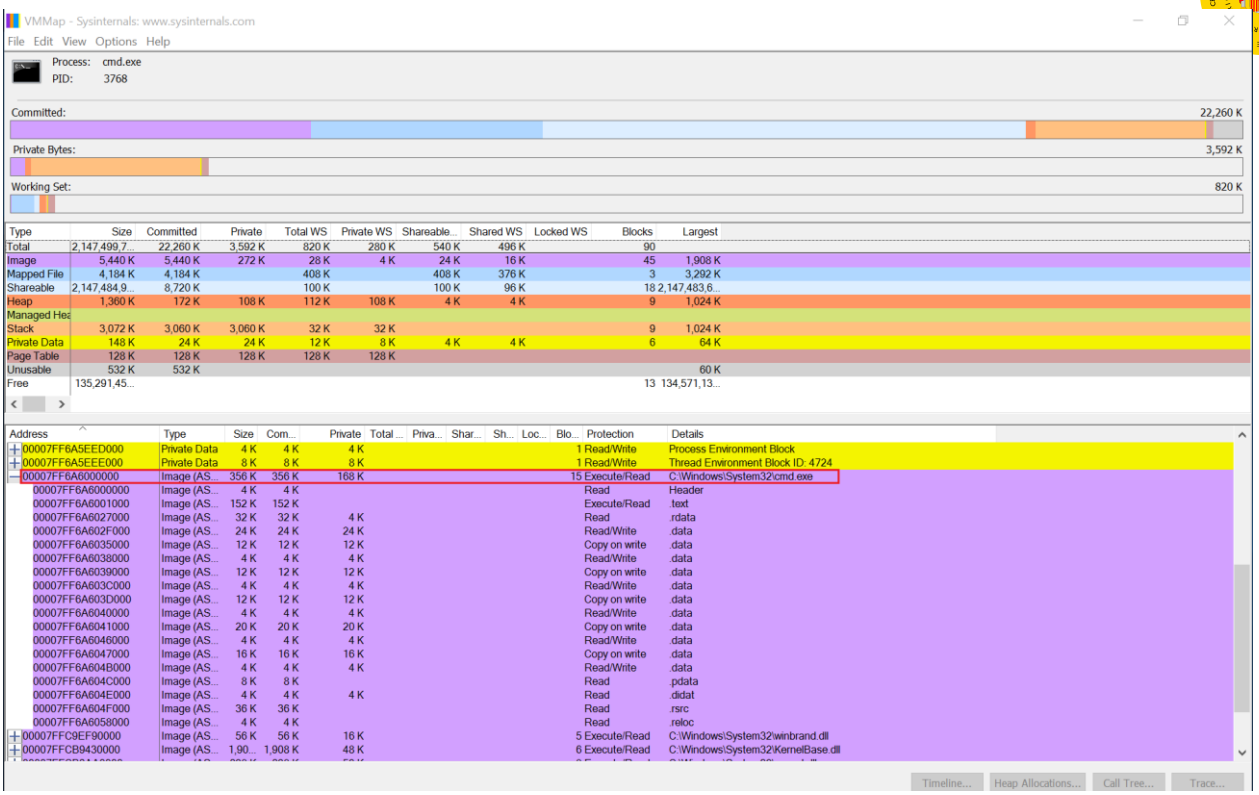
Obtained results:

The evaluator has executed two instances of the same executable file at different platforms, and has observed that the memory address is different from each one. The following images demonstrate this fact:

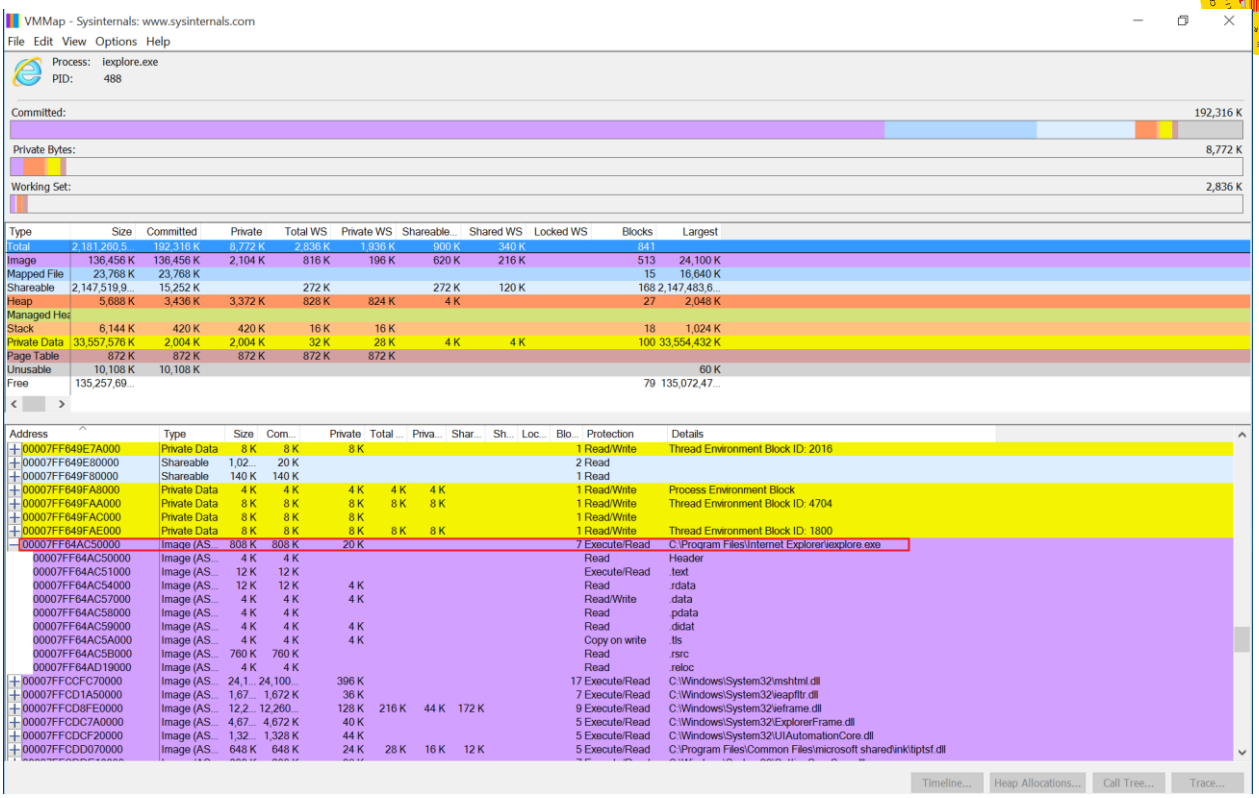
- Process: Command Prompt (cmd.exe)
    - Platform A (Address 0x00007FF635A00000):



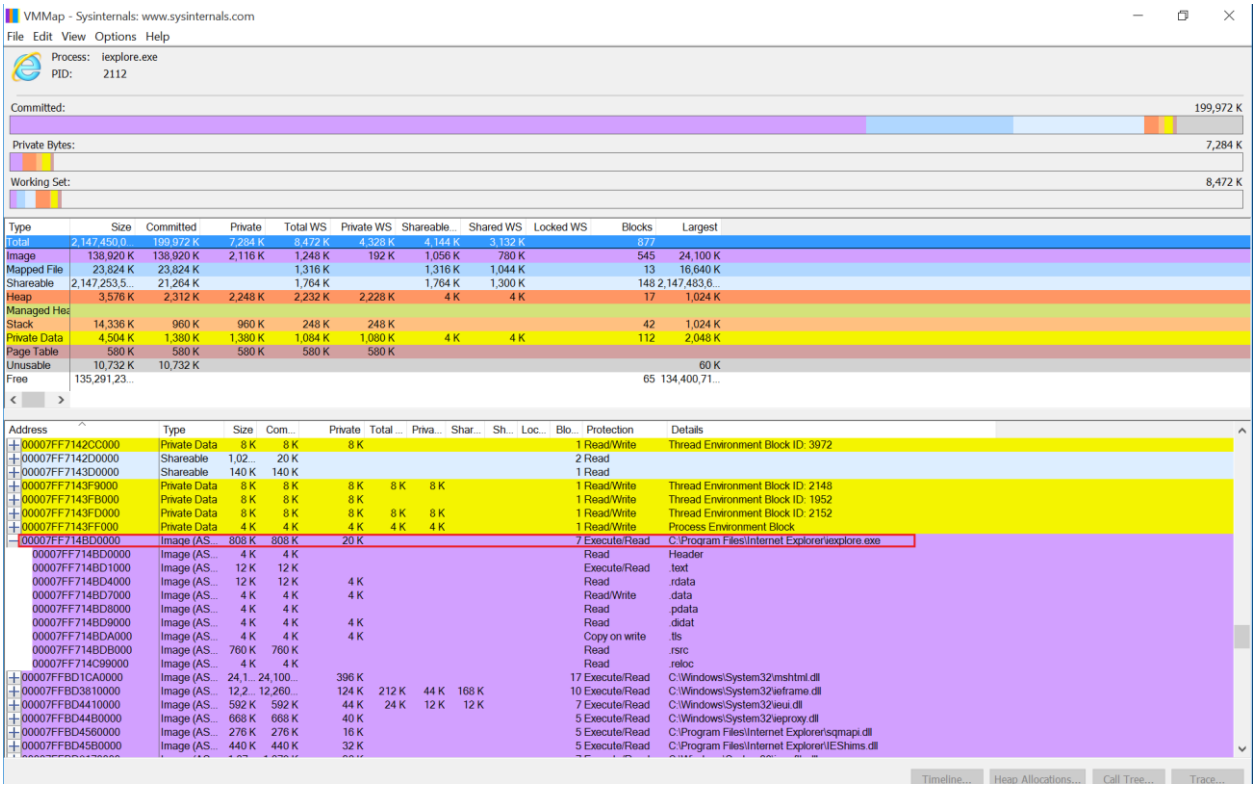
    - Platform B (Address 0x00007FF6A6000000):



- Process: Internet Explorer 11 (iexplore.exe)
  - Platform A (Address 0x00007FF64AC50000):

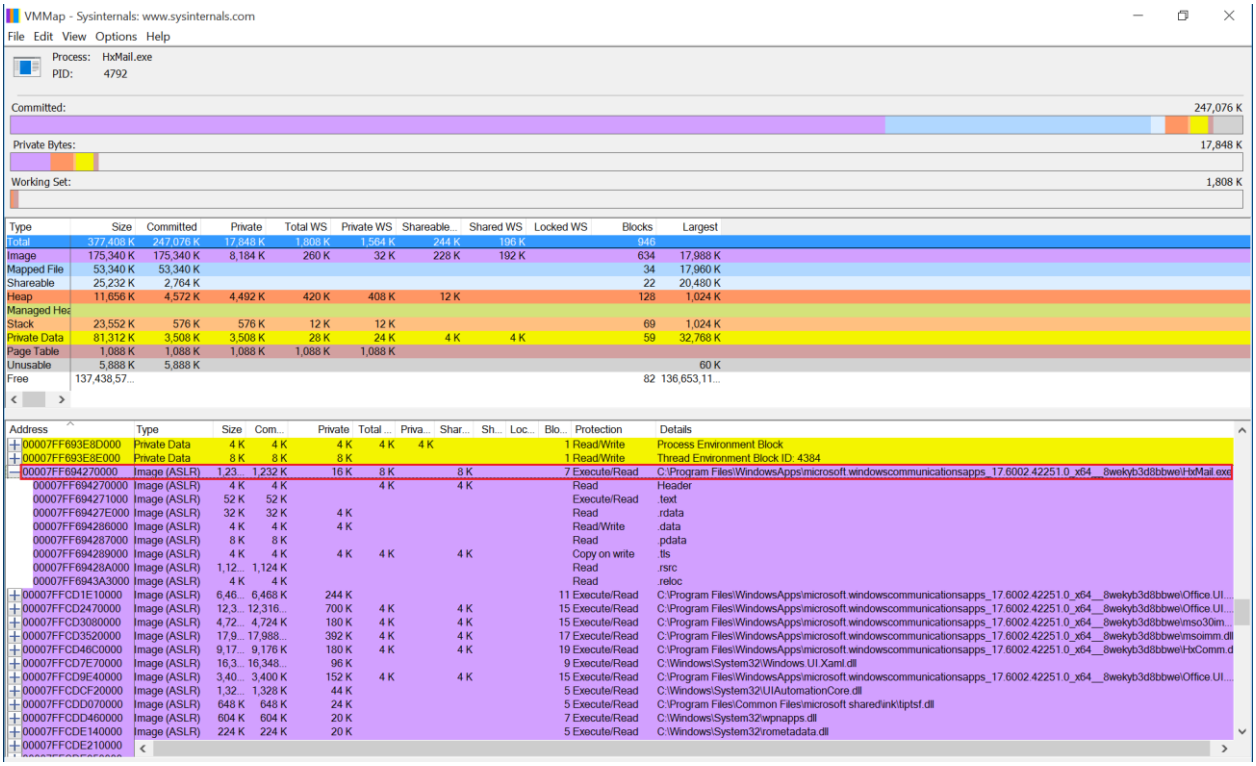


- Platform B (Address 0x00007FF7114BD0000):

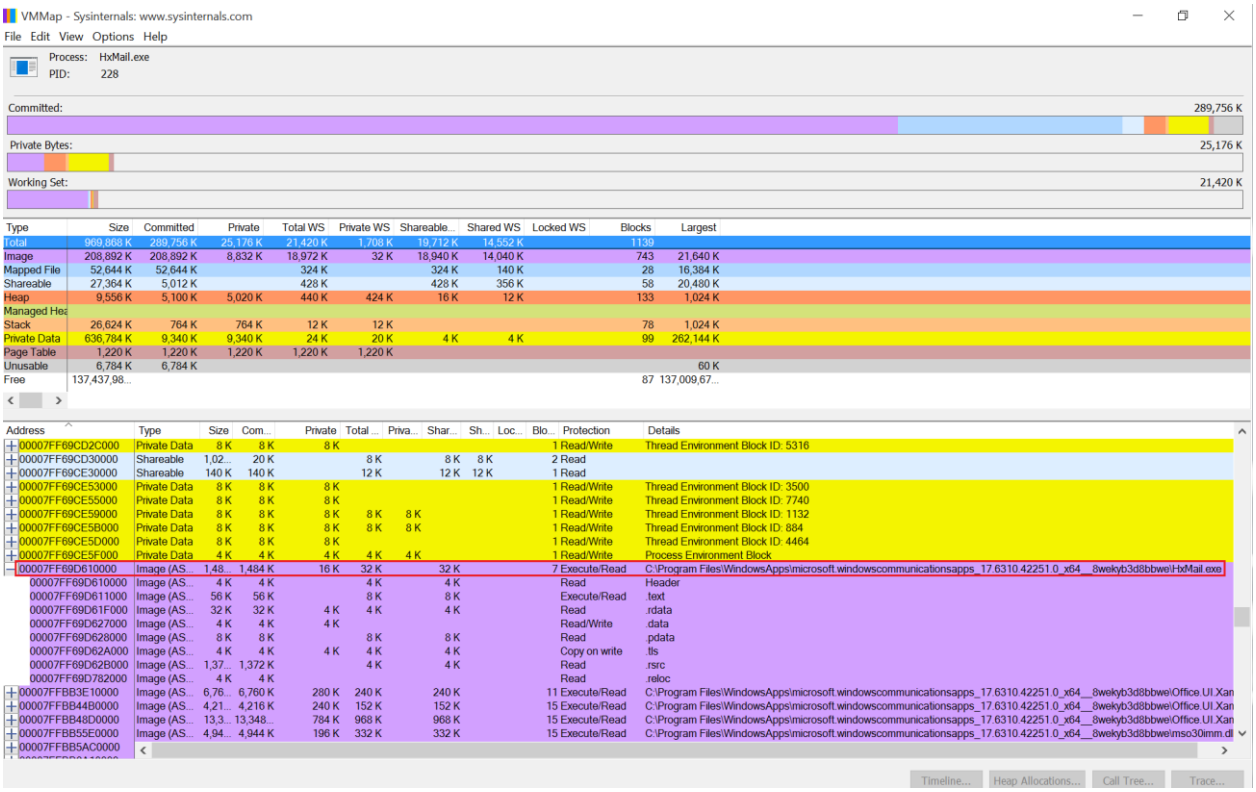


- Process: Mail App (HxMail.exe)

- Platform A (Address 0x 00007FF694270000):



- Platform B (Address 0x00007FF69D610000):



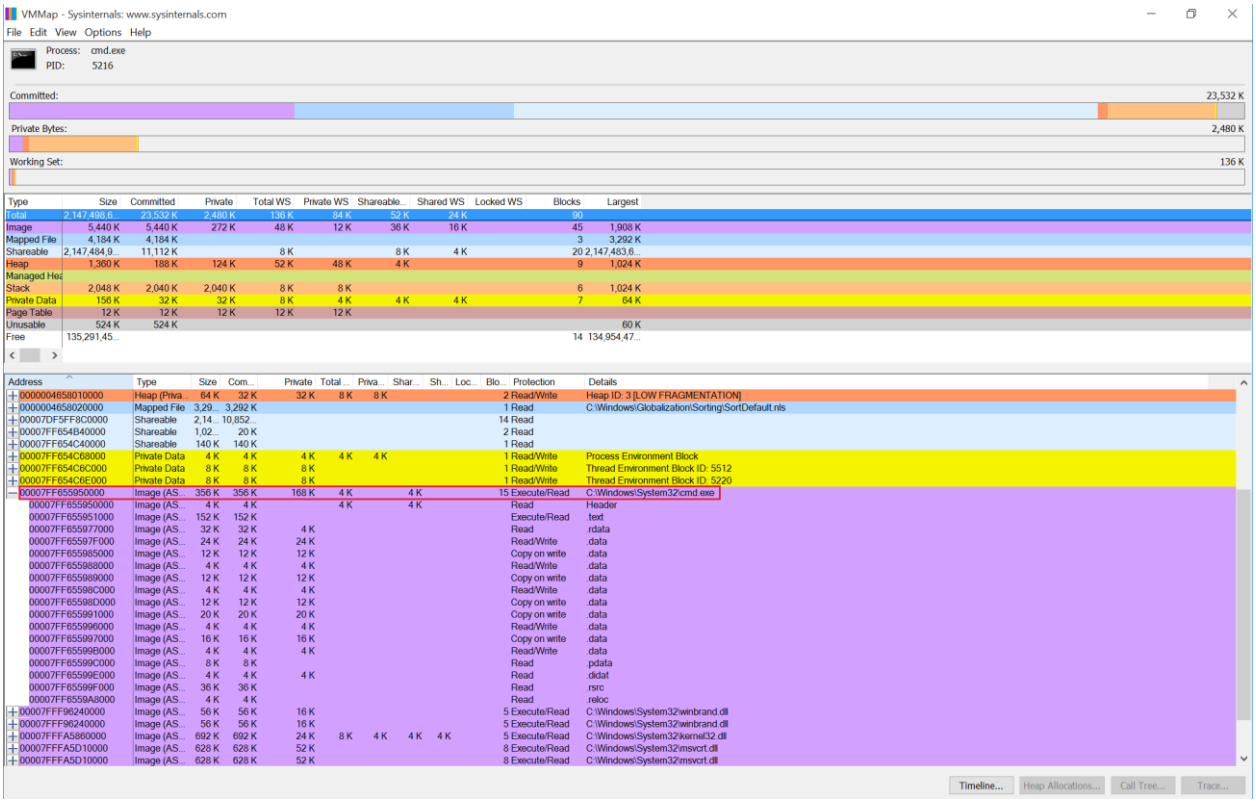
Testing platform:

- Surface 3 Pro with Windows 10 x64 Enterprise Edition

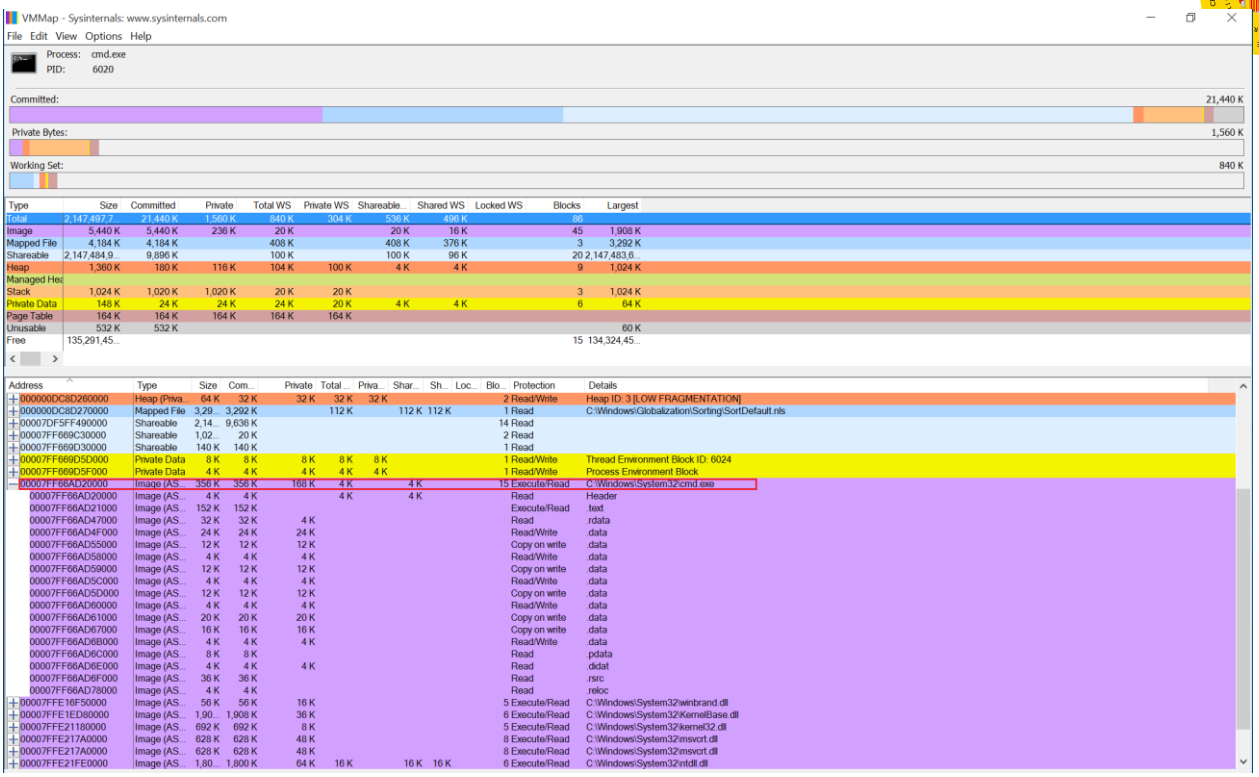
Obtained results:

The evaluator has executed two instances of the same executable file at different platforms, and has observed that the memory address is different from each one. The following images demonstrate this fact:

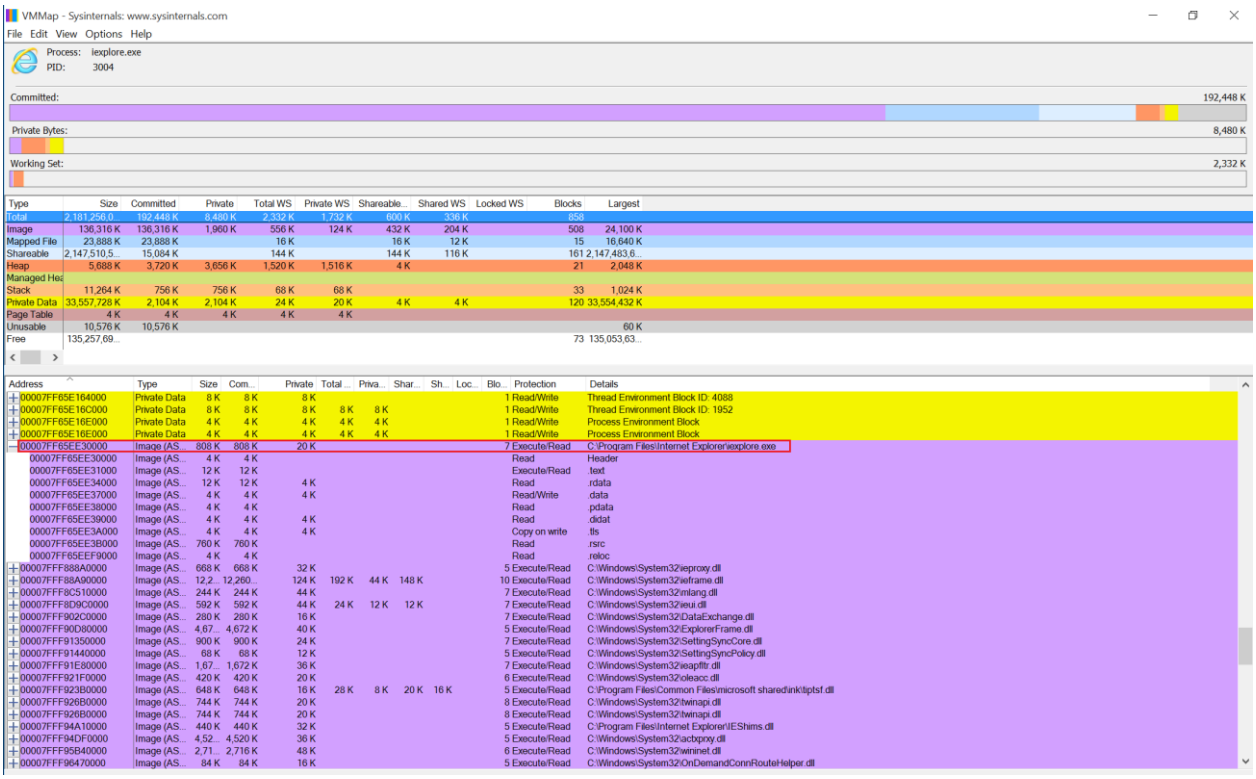
- Process: Command Prompt (cmd.exe)
  - Platform A (Address 0x00007FF655950000):



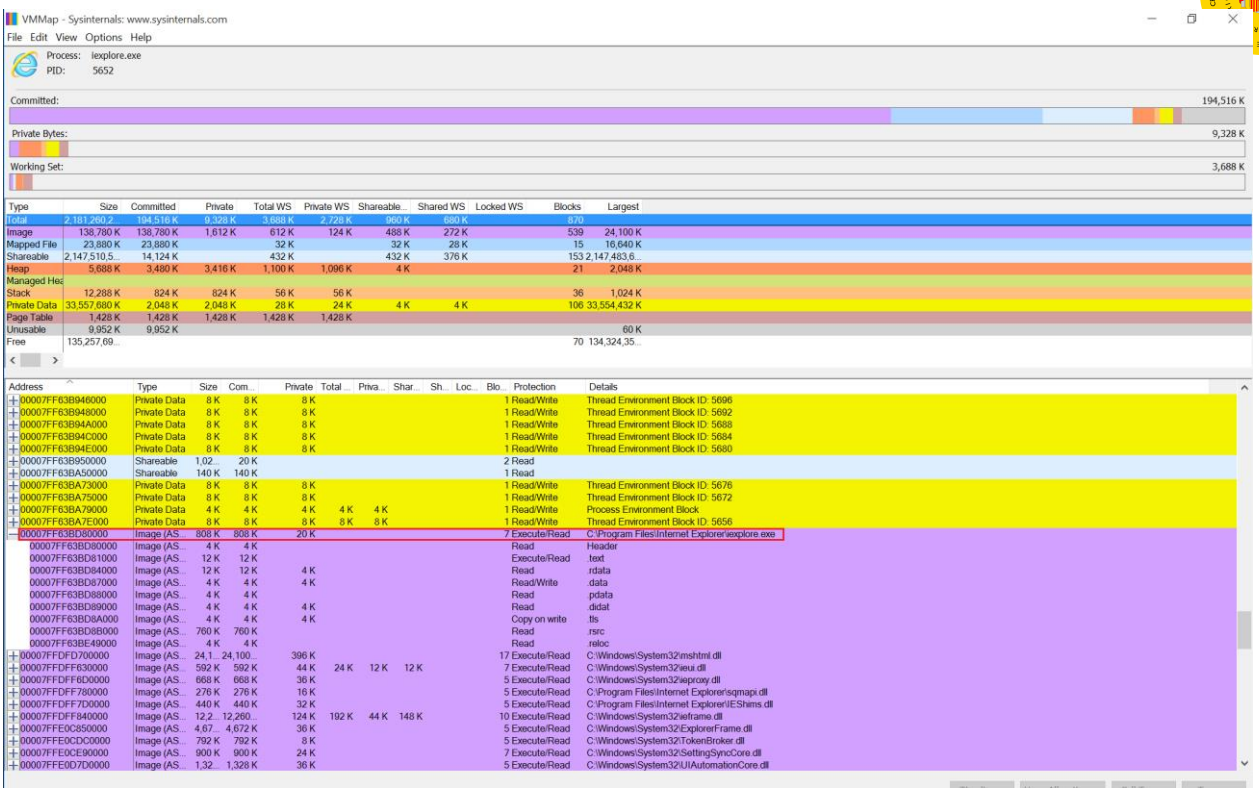
  - Platform B (Address 0x00007FF66AD20000):



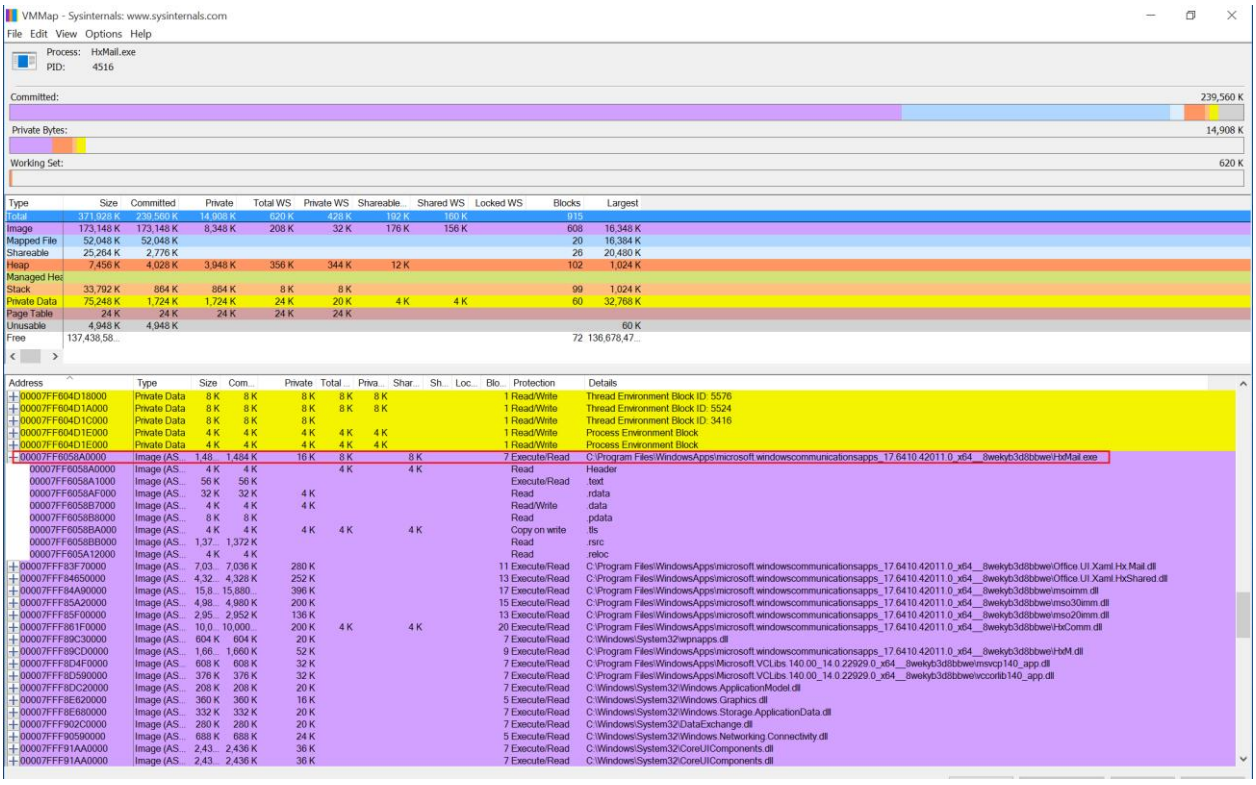
- Process: Internet Explorer 11 (iexplore.exe)
  - Platform A (Address 0x00007FF65EE30000):



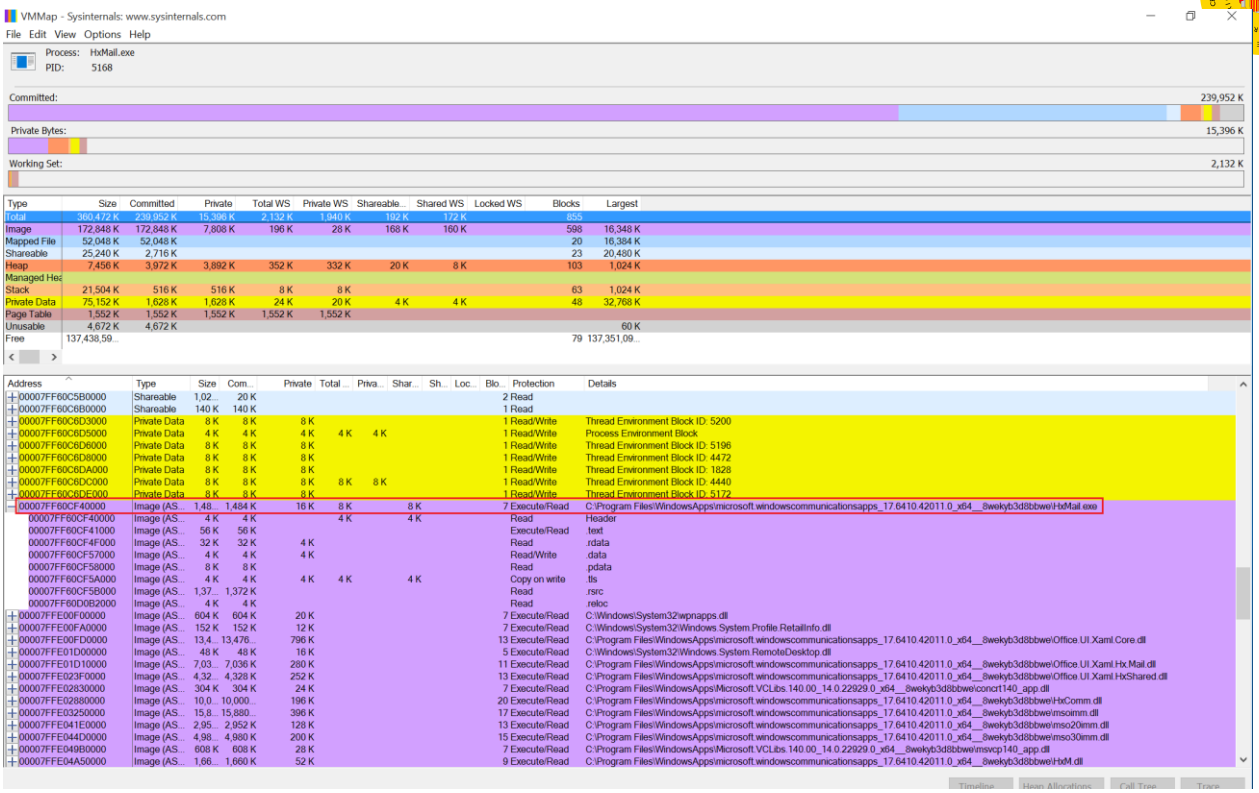
  - Platform B (Address 0x00007FF638D80000):



- Process: Mail App (HxMail.exe)
  - Platform A (Address 0x00007FF6058A0000):



  - Platform B (Address 0x00007FF60CF40000):



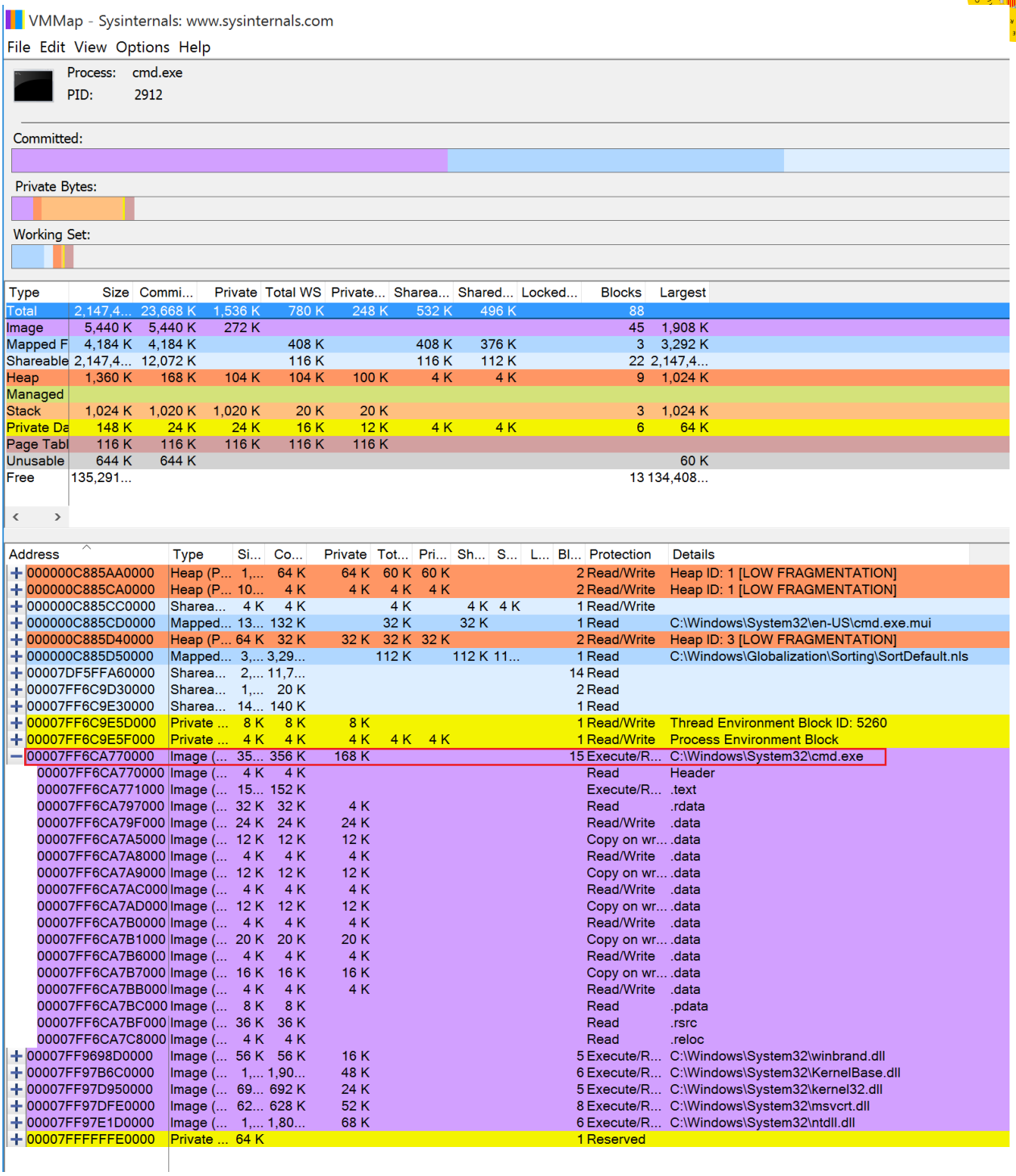
Testing platform:

- Surface Book with Windows 10 x64 Enterprise Edition

Obtained results:

Due to the fact that there was only one available platform, the evaluator has followed the steps defined in *Procedure* section, *Method 2*. The evaluator has executed one executable file in two different rounds over the same platform, and has observed that the memory address is different from each one. The following images demonstrate this fact:

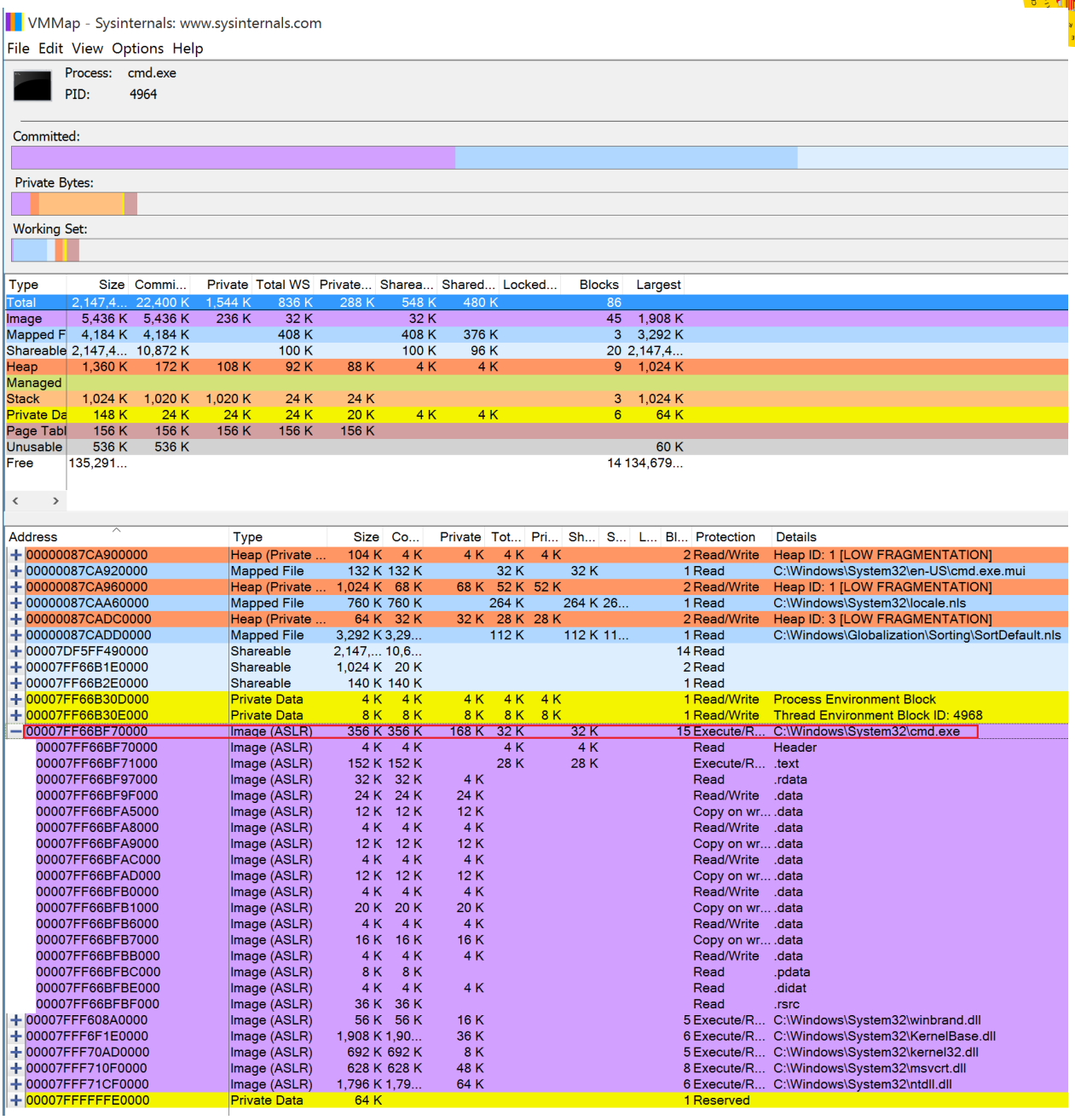
- Process: Command Prompt (cmd.exe)
    - Platform A – Round 1 (Address 0x00007FF6CA770000):



Type	Size	Commi...	Private	Total WS	Private...	Sharea...	Shared...	Locked...	Blocks	Largest
Total	2,147,4...	23,668 K	1,536 K	780 K	248 K	532 K	496 K		88	
Image	5,440 K	5,440 K	272 K						45	1,908 K
Mapped F	4,184 K	4,184 K		408 K		408 K	376 K		3	3,292 K
Shareable	2,147,4...	12,072 K		116 K		116 K	112 K		22	2,147,4...
Heap	1,360 K	168 K	104 K	104 K	100 K	4 K	4 K		9	1,024 K
Managed										
Stack	1,024 K	1,020 K	1,020 K	20 K	20 K				3	1,024 K
Private Da	148 K	24 K	24 K	16 K	12 K	4 K	4 K		6	64 K
Page Tabl	116 K	116 K	116 K	116 K	116 K					
Unusable	644 K	644 K								60 K
Free	135,291...								13	134,408...

Address	Type	Si...	Co...	Private	Tot...	Pri...	Sh...	S...	L...	Bl...	Protection	Details
000000C885AA0000	Heap (P...	1,...	64 K	64 K	60 K	60 K				2	Read/Write	Heap ID: 1 [LOW FRAGMENTATION]
000000C885CA0000	Heap (P...	10...	4 K	4 K	4 K	4 K				2	Read/Write	Heap ID: 1 [LOW FRAGMENTATION]
000000C885CC0000	Sharea...	4 K	4 K		4 K		4 K	4 K		1	Read/Write	
000000C885CD0000	Mapped...	13...	132 K		32 K		32 K			1	Read	C:\Windows\System32\en-US\cmd.exe.mui
000000C885D40000	Heap (P...	64 K	32 K	32 K	32 K	32 K				2	Read/Write	Heap ID: 3 [LOW FRAGMENTATION]
000000C885D50000	Mapped...	3,...	3,29...		112 K		112 K	11...		1	Read	C:\Windows\Globalization\Sorting\SortDefault.nls
00007DF5FFA60000	Sharea...	2,...	11,7...							14	Read	
00007FF6C9D30000	Sharea...	1,...	20 K							2	Read	
00007FF6C9E30000	Sharea...	14...	140 K							1	Read	
00007FF6C9E5D000	Private ...	8 K	8 K	8 K						1	Read/Write	Thread Environment Block ID: 5260
00007FF6C9E5F000	Private ...	4 K	4 K	4 K	4 K	4 K				1	Read/Write	Process Environment Block
00007FF6CA770000	Image (...	35...	356 K	168 K						15	Execute/R...	C:\Windows\System32\cmd.exe
00007FF6CA770000	Image (...	4 K	4 K								Read	Header
00007FF6CA771000	Image (...	15...	152 K								Execute/R...	.text
00007FF6CA797000	Image (...	32 K	32 K	4 K							Read	.rdata
00007FF6CA79F000	Image (...	24 K	24 K	24 K							Read/Write	.data
00007FF6CA7A5000	Image (...	12 K	12 K	12 K							Copy on wr...	.data
00007FF6CA7A8000	Image (...	4 K	4 K	4 K							Read/Write	.data
00007FF6CA7A9000	Image (...	12 K	12 K	12 K							Copy on wr...	.data
00007FF6CA7AC000	Image (...	4 K	4 K	4 K							Read/Write	.data
00007FF6CA7AD000	Image (...	12 K	12 K	12 K							Copy on wr...	.data
00007FF6CA7B0000	Image (...	4 K	4 K	4 K							Read/Write	.data
00007FF6CA7B1000	Image (...	20 K	20 K	20 K							Copy on wr...	.data
00007FF6CA7B6000	Image (...	4 K	4 K	4 K							Read/Write	.data
00007FF6CA7B7000	Image (...	16 K	16 K	16 K							Copy on wr...	.data
00007FF6CA7BB000	Image (...	4 K	4 K	4 K							Read/Write	.data
00007FF6CA7BC000	Image (...	8 K	8 K								Read	.pdata
00007FF6CA7BF000	Image (...	36 K	36 K								Read	.rsrc
00007FF6CA7C8000	Image (...	4 K	4 K								Read	.reloc
00007FF9698D0000	Image (...	56 K	56 K	16 K						5	Execute/R...	C:\Windows\System32\winbrand.dll
00007FF97B6C0000	Image (...	1,...	1,90...	48 K						6	Execute/R...	C:\Windows\System32\KernelBase.dll
00007FF97D950000	Image (...	69...	692 K	24 K						5	Execute/R...	C:\Windows\System32\kernel32.dll
00007FF97DFE0000	Image (...	62...	628 K	52 K						8	Execute/R...	C:\Windows\System32\msvcrt.dll
00007FF97E1D0000	Image (...	1,...	1,80...	68 K						6	Execute/R...	C:\Windows\System32\ntdll.dll
00007FFFFFFE0000	Private ...	64 K								1	Reserved	

- Platform A – Round 2 (Address 0x00007FF66BF70000):



VMMap - Sysinternals: www.sysinternals.com

File Edit View Options Help

Process: cmd.exe
PID: 4964

Committed:

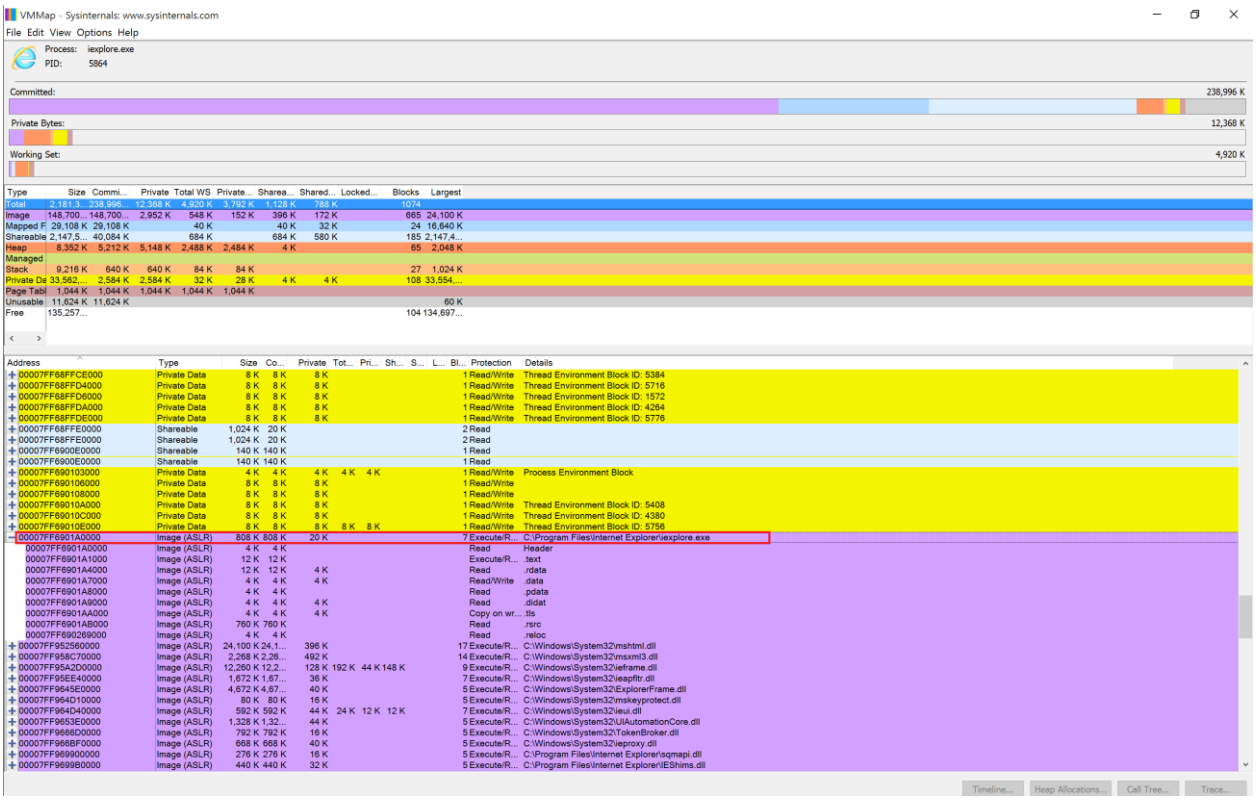
Private Bytes:

Working Set:

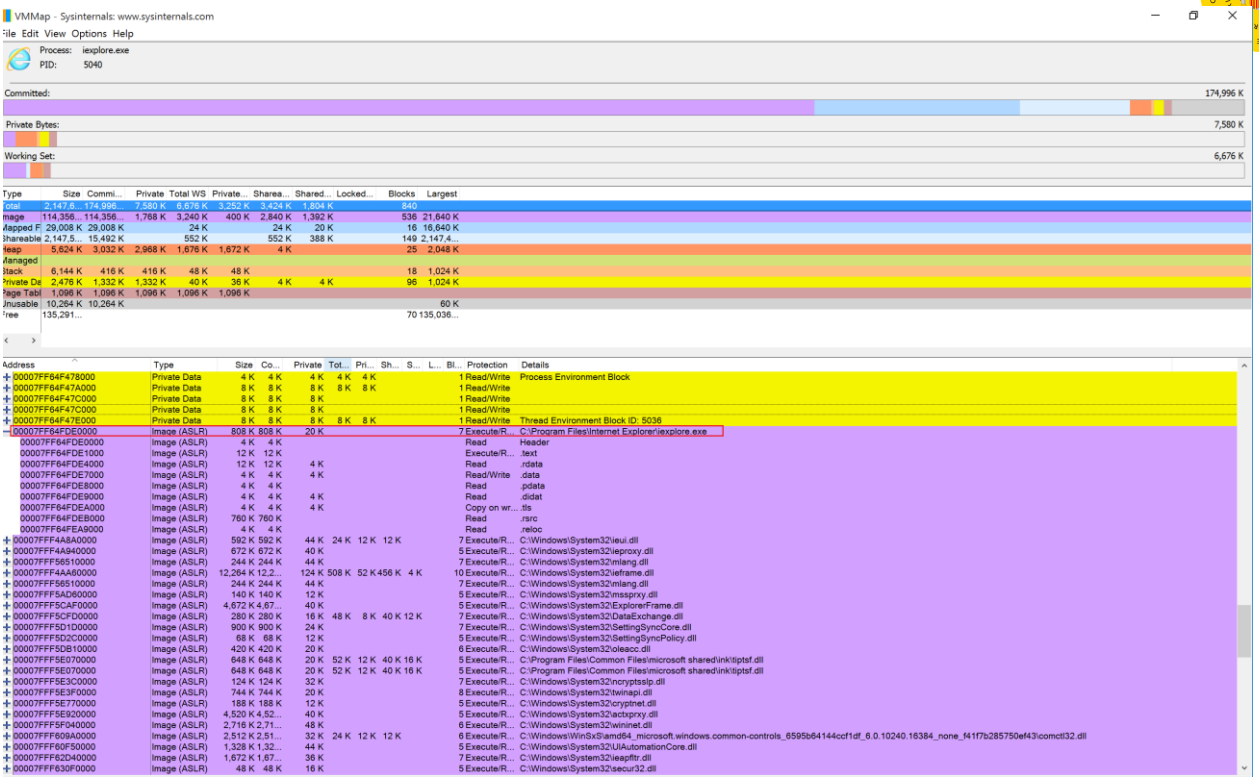
Type	Size	Commi...	Private	Total WS	Private...	Sharea...	Shared...	Locked...	Blocks	Largest
Total	2,147,4...	22,400 K	1,544 K	836 K	288 K	548 K	480 K		86	
Image	5,436 K	5,436 K	236 K	32 K		32 K			45	1,908 K
Mapped F	4,184 K	4,184 K		408 K		408 K	376 K		3	3,292 K
Shareable	2,147,4...	10,872 K		100 K		100 K	96 K		20	2,147,4...
Heap	1,360 K	172 K	108 K	92 K	88 K	4 K	4 K		9	1,024 K
Managed										
Stack	1,024 K	1,020 K	1,020 K	24 K	24 K				3	1,024 K
Private Da	148 K	24 K	24 K	24 K	20 K	4 K	4 K		6	64 K
Page Tabl	156 K	156 K	156 K	156 K	156 K					
Unusable	536 K	536 K								60 K
Free	135,291...								14	134,679...

Address	Type	Size	Co...	Private	Tot...	Pri...	Sh...	S...	L...	Bl...	Protection	Details
00000087CA900000	Heap (Private ...	104 K	4 K	4 K	4 K	4 K				2	Read/Write	Heap ID: 1 [LOW FRAGMENTATION]
00000087CA920000	Mapped File	132 K	132 K		32 K		32 K			1	Read	C:\Windows\System32\en-US\cmd.exe.mui
00000087CA960000	Heap (Private ...	1,024 K	68 K	68 K	52 K	52 K				2	Read/Write	Heap ID: 1 [LOW FRAGMENTATION]
00000087CAA60000	Mapped File	760 K	760 K		264 K		264 K	26...		1	Read	C:\Windows\System32\locale.nls
00000087CADC0000	Heap (Private ...	64 K	32 K	32 K	28 K	28 K				2	Read/Write	Heap ID: 3 [LOW FRAGMENTATION]
00000087CADD0000	Mapped File	3,292 K	3,29...		112 K		112 K	11...		1	Read	C:\Windows\Globalization\Sorting\SortDefault.nls
00007DF5FF490000	Shareable	2,147,...	10,6...							14	Read	
00007FF66B1E0000	Shareable	1,024 K	20 K							2	Read	
00007FF66B2E0000	Shareable	140 K	140 K							1	Read	
00007FF66B30D000	Private Data	4 K	4 K	4 K	4 K	4 K				1	Read/Write	Process Environment Block
00007FF66B30E000	Private Data	8 K	8 K	8 K	8 K	8 K				1	Read/Write	Thread Environment Block ID: 4968
00007FF66BF70000	Image (ASLR)	356 K	356 K	168 K	32 K		32 K			15	Execute/R...	C:\Windows\System32\cmd.exe
00007FF66BF70000	Image (ASLR)	4 K	4 K		4 K		4 K				Read	Header
00007FF66BF71000	Image (ASLR)	152 K	152 K		28 K		28 K				Execute/R...	.text
00007FF66BF97000	Image (ASLR)	32 K	32 K	4 K							Read	.rdata
00007FF66BF9F000	Image (ASLR)	24 K	24 K	24 K							Read/Write	.data
00007FF66BFA5000	Image (ASLR)	12 K	12 K	12 K							Copy on wr...	.data
00007FF66BFA8000	Image (ASLR)	4 K	4 K	4 K							Read/Write	.data
00007FF66BFA9000	Image (ASLR)	12 K	12 K	12 K							Copy on wr...	.data
00007FF66BFAC000	Image (ASLR)	4 K	4 K	4 K							Read/Write	.data
00007FF66BFAD000	Image (ASLR)	12 K	12 K	12 K							Copy on wr...	.data
00007FF66BFB0000	Image (ASLR)	4 K	4 K	4 K							Read/Write	.data
00007FF66BFB1000	Image (ASLR)	20 K	20 K	20 K							Copy on wr...	.data
00007FF66BFB6000	Image (ASLR)	4 K	4 K	4 K							Read/Write	.data
00007FF66BFB7000	Image (ASLR)	16 K	16 K	16 K							Copy on wr...	.data
00007FF66BFBB000	Image (ASLR)	4 K	4 K	4 K							Read/Write	.data
00007FF66BFBC000	Image (ASLR)	8 K	8 K								Read	.pdata
00007FF66BFBE000	Image (ASLR)	4 K	4 K	4 K							Read	.didat
00007FF66BFBF000	Image (ASLR)	36 K	36 K								Read	.rsrc
00007FFF608A0000	Image (ASLR)	56 K	56 K	16 K						5	Execute/R...	C:\Windows\System32\winbrand.dll
00007FFF6F1E0000	Image (ASLR)	1,908 K	1,90...	36 K						6	Execute/R...	C:\Windows\System32\KernelBase.dll
00007FFF70AD0000	Image (ASLR)	692 K	692 K	8 K						5	Execute/R...	C:\Windows\System32\kernel32.dll
00007FFF710F0000	Image (ASLR)	628 K	628 K	48 K						8	Execute/R...	C:\Windows\System32\msvcrt.dll
00007FFF71CF0000	Image (ASLR)	1,796 K	1,79...	64 K						6	Execute/R...	C:\Windows\System32\ntdll.dll
00007FFFFFFE0000	Private Data	64 K								1	Reserved	

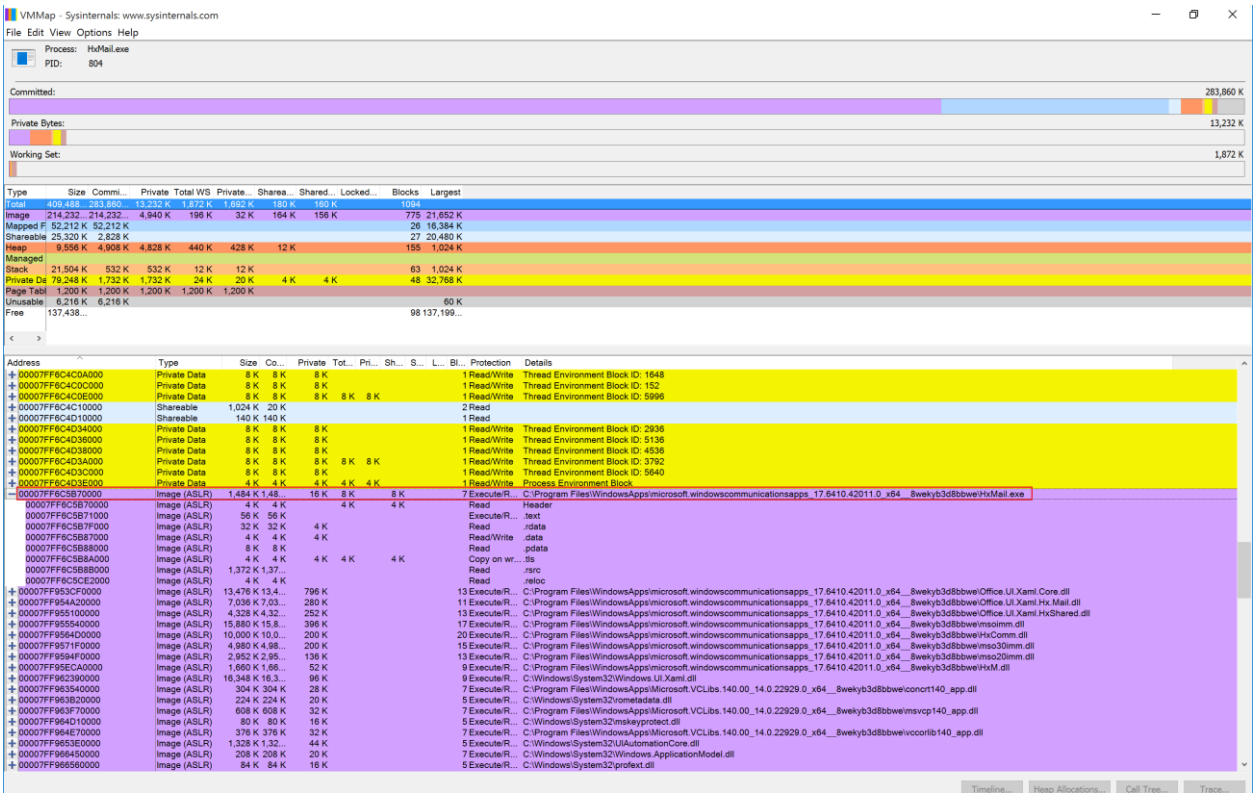
- Process: Internet Explorer 11 (iexplore.exe)
    - Platform A – Round 1 (Address 0x00007FF6901A0000):



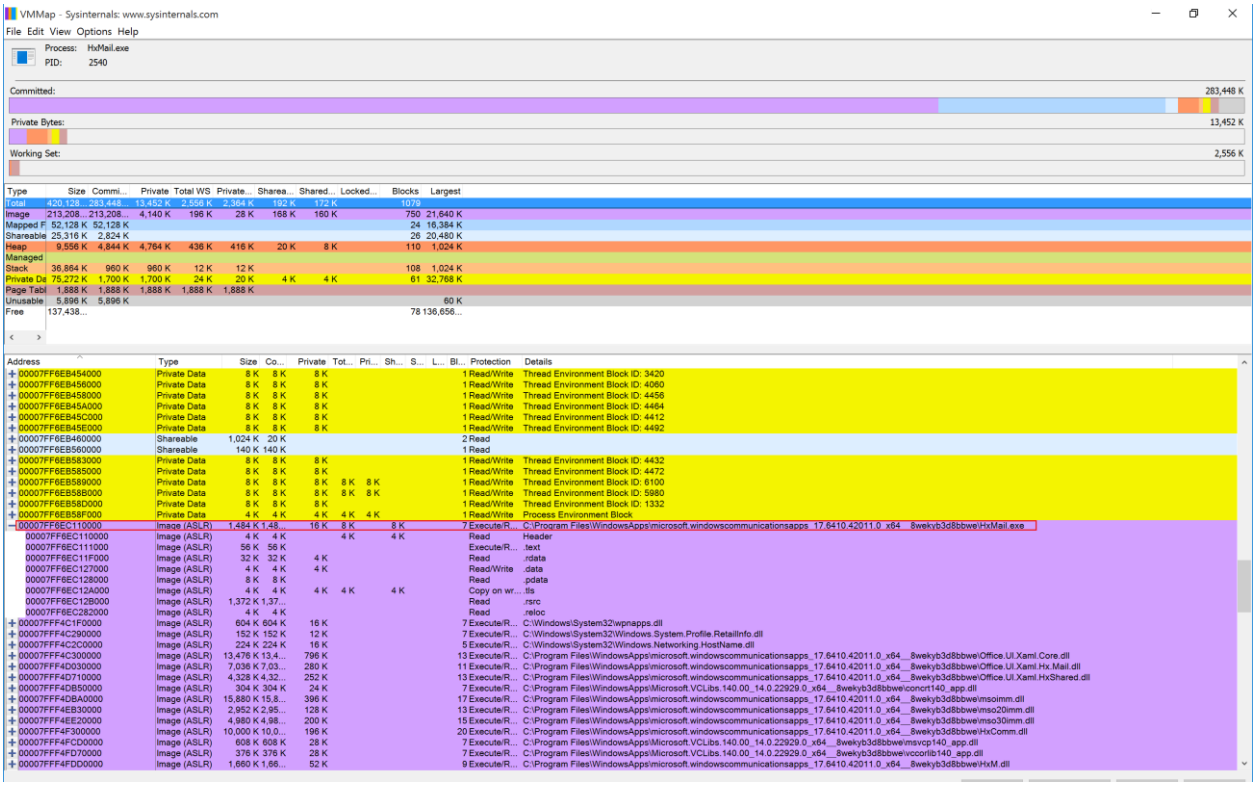
    - Platform A – Round 2 (Address 0x00007FF64FDE0000):



- Process: Mail App (HxMail.exe)
  - Platform A – Round 1 (Address 0x00007FF6C5B70000):



- Platform A – Round 2 (Address 0x00007FF6EC110000):



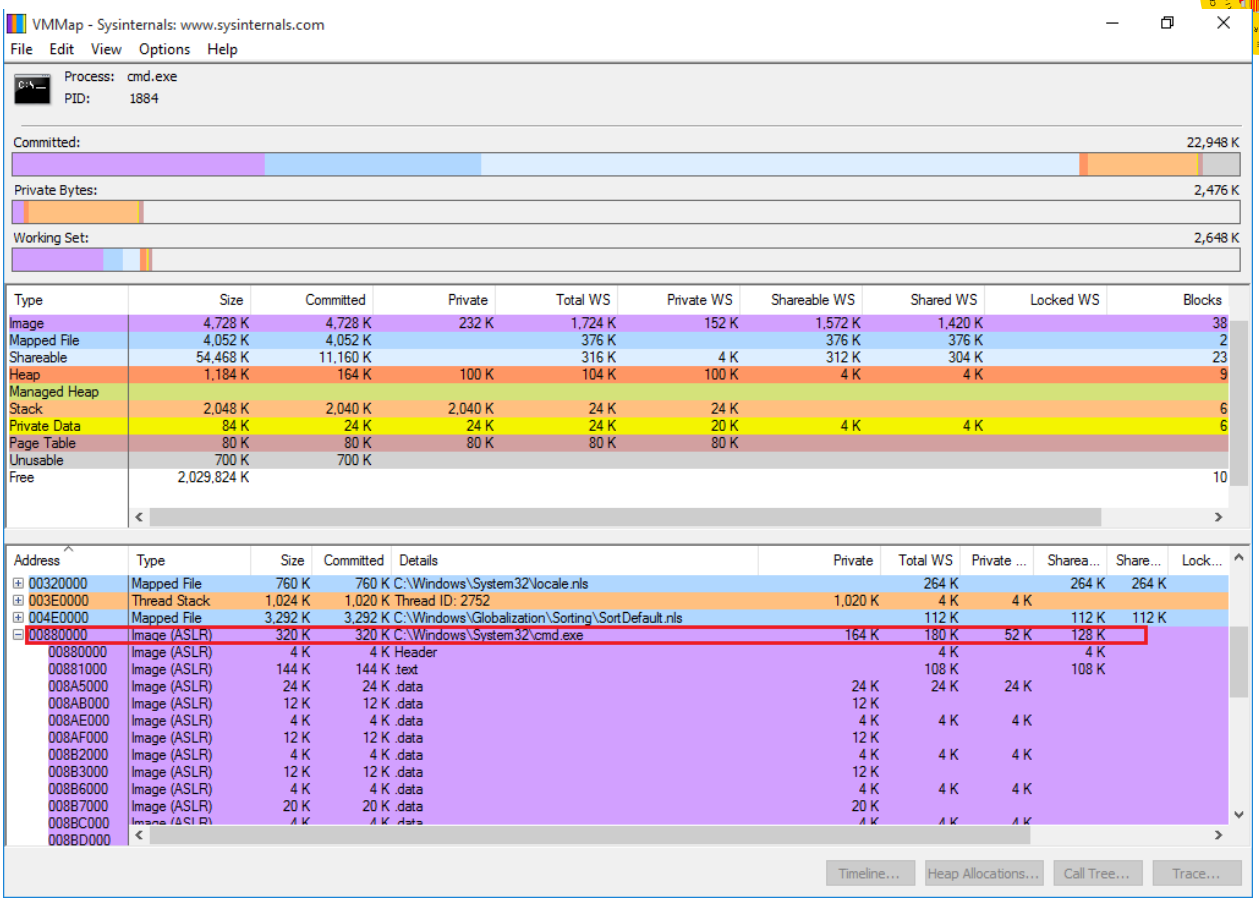
Testing platform:

- Windows Server 2012 R2 Hyper-V with Windows 10 x86 Home Edition

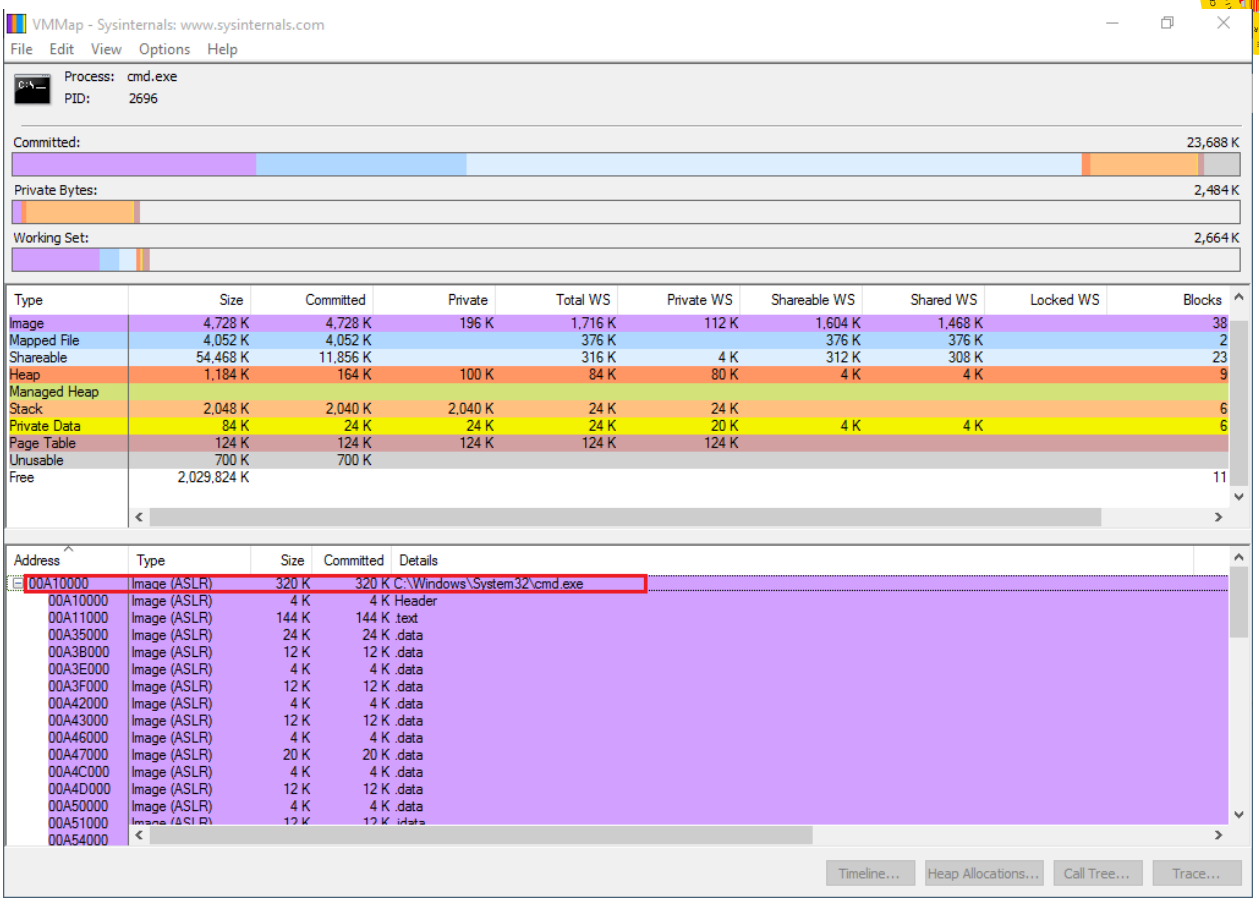
Obtained results:

Due to the fact that there was only one available platform, the evaluator has followed the steps defined in *Procedure* section, *Method 2*. The evaluator has executed one executable file in two different rounds over the same platform, and has observed that the memory address is different from each one. The following images demonstrate this fact:

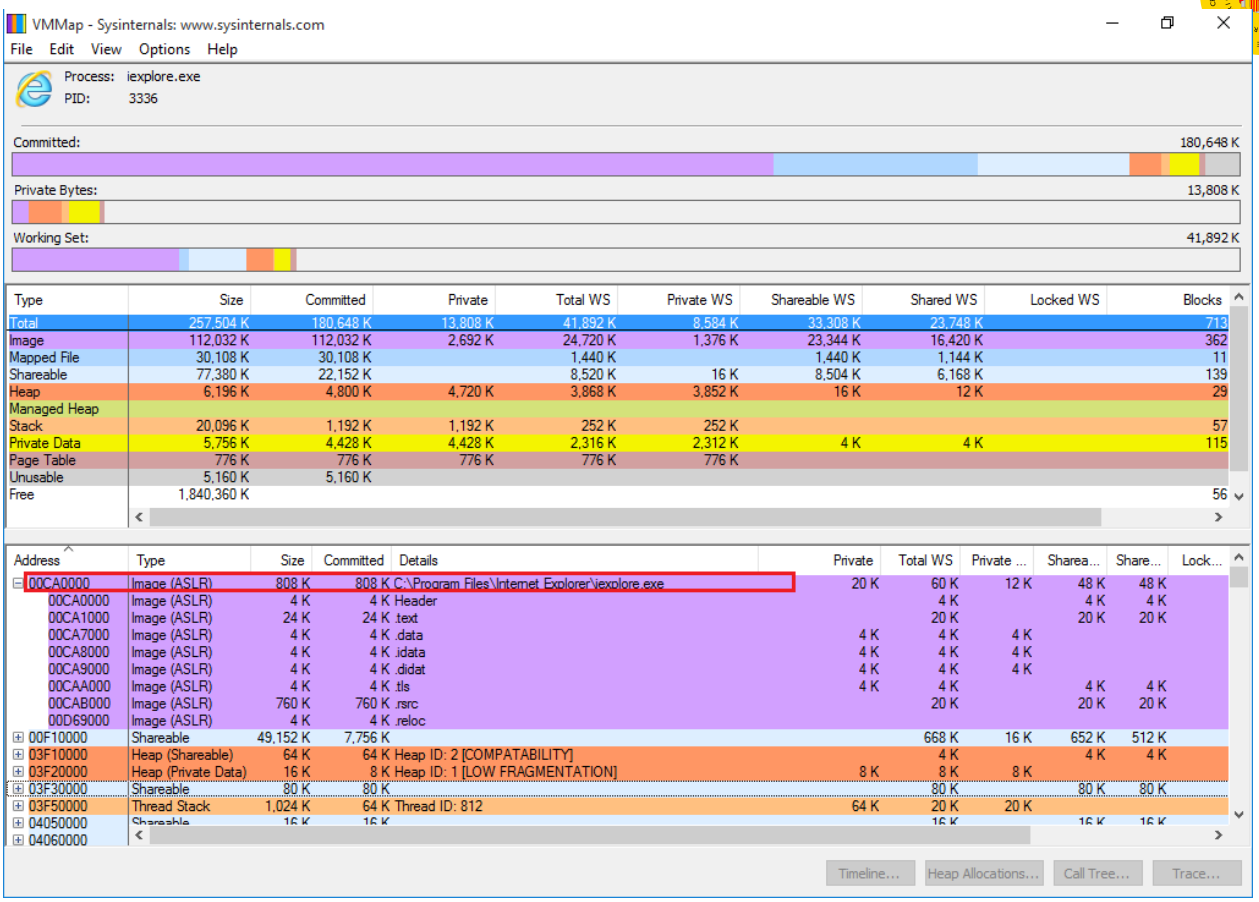
- Process: Command Prompt (cmd.exe)
  - Platform A - Round 1 (Address 0x00880000):



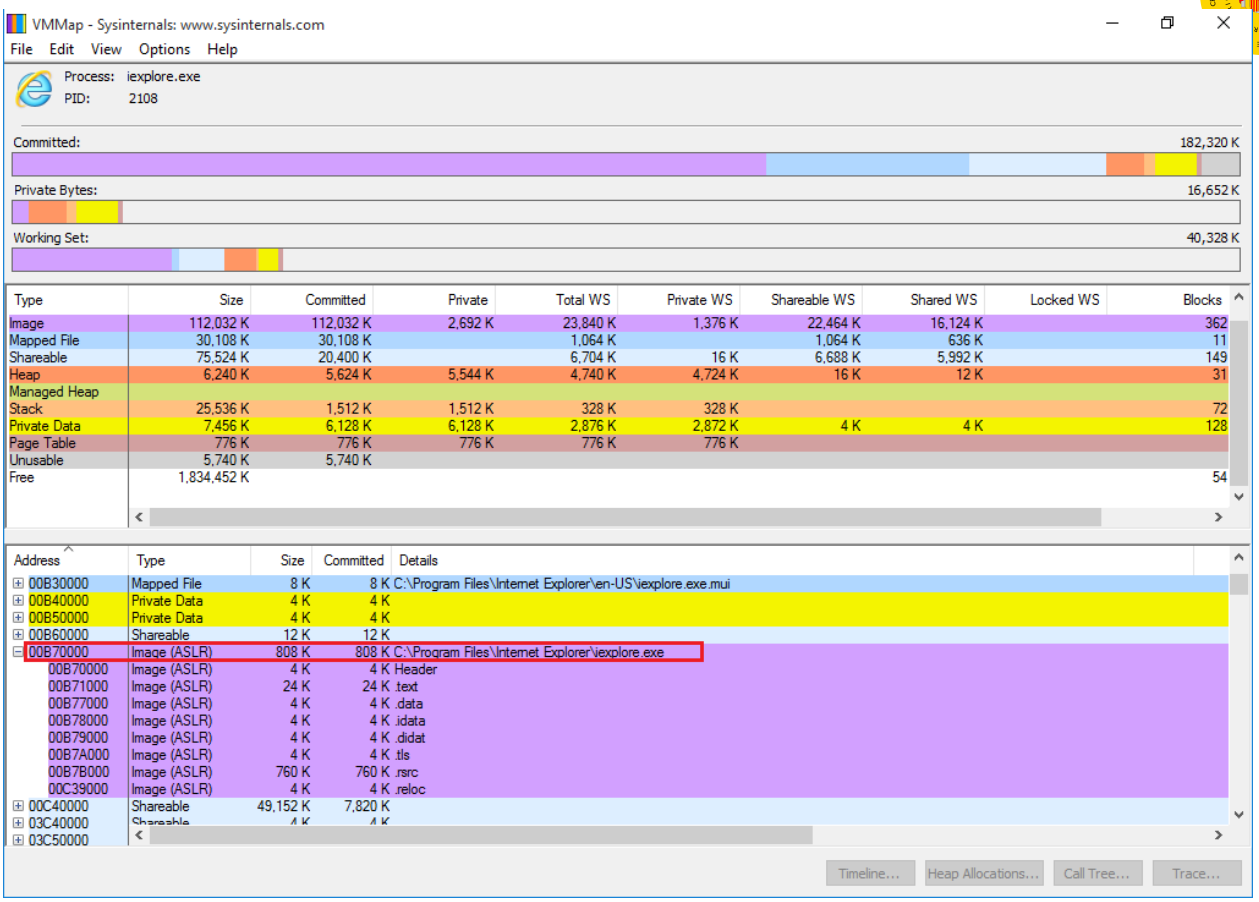
- Platform A - Round 2 (Address 0x00A10000):



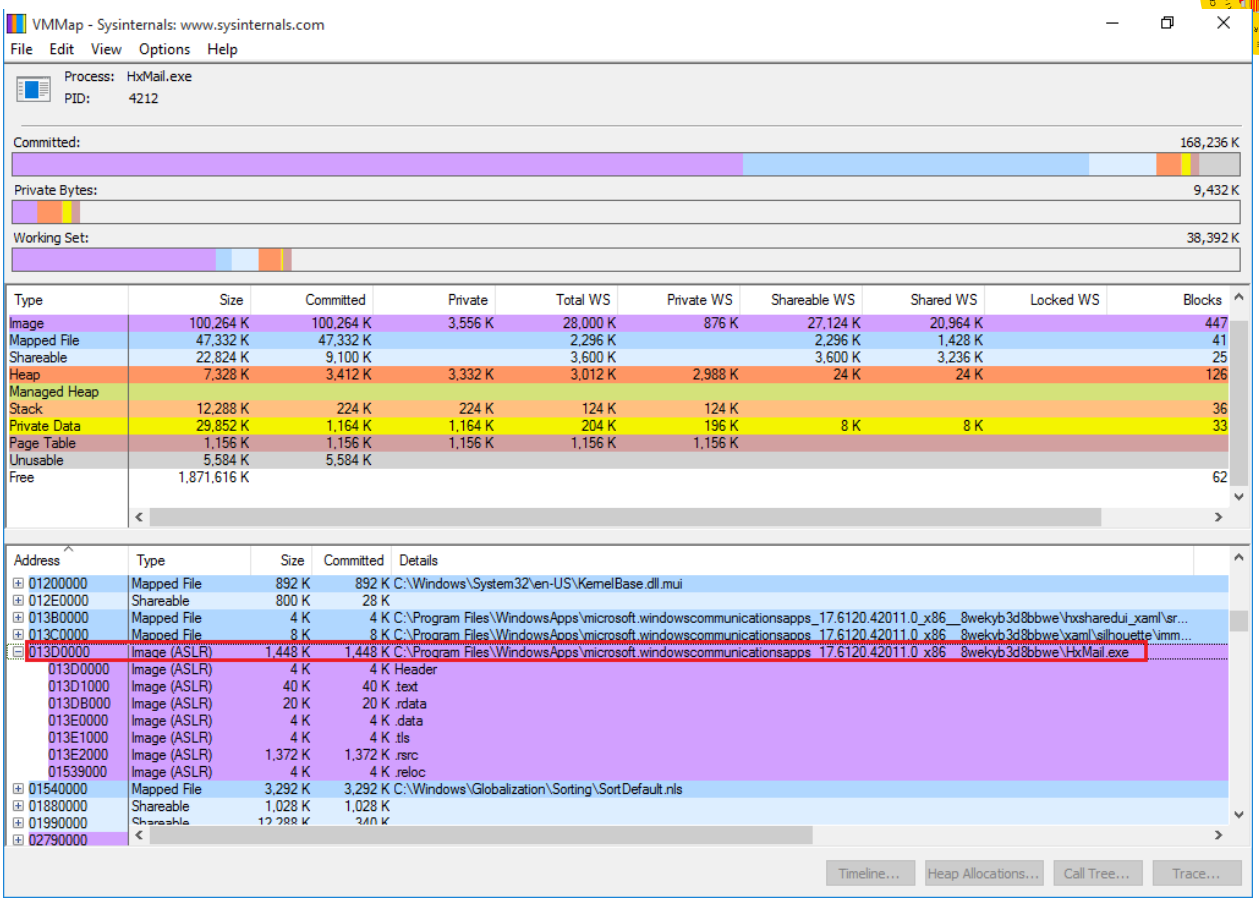
- Process: Internet Explorer 11 (iexplore.exe)
    - Platform A - Round 1 (Address 0x00CA0000):



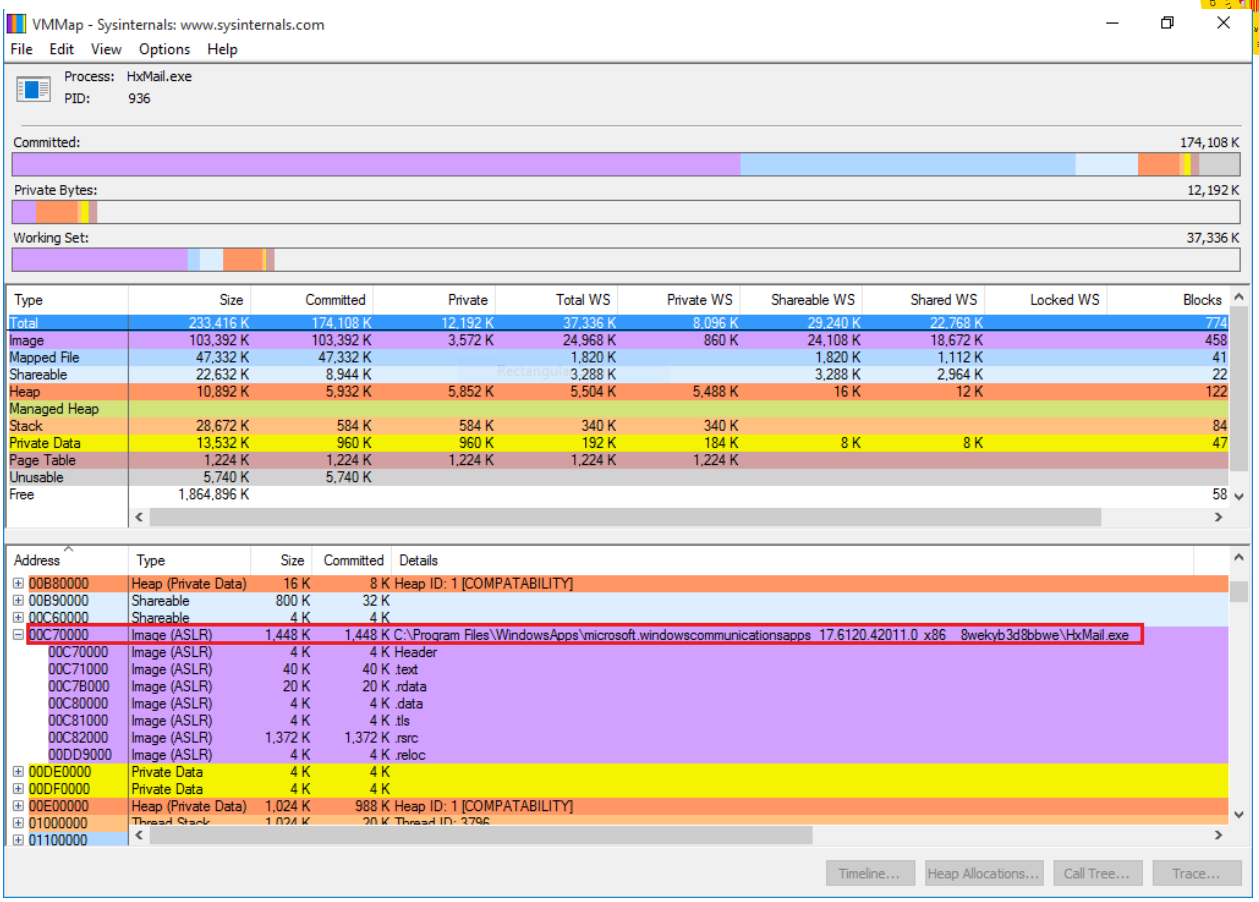
- Platform A - Round 2 (Address 0x00B70000):



- Process: Mail App (HxMail.exe)
    - Platform A - Round 1 (Address 0x013D0000):



- Platform A - Round 2 (Address 0x00C70000):



30.3.1.4. Verdict

As it can be appreciated in the image above in all the cases the memory assigned to one process is different. Therefore the evaluator considers that, the tests results obtained during the test activity demonstrate the fulfillment of the requirements established in the assurance activity section. So, the **PASS** verdict is assigned to the test activity.

30.4. Final Verdict

Due to all test activities 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 this FPT_ASLR_EXT.1.1.

31. FPT_SBOP_EXT.1.1

31.1. Assurance activity

The evaluator will determine that the TSS contains a description of stack-based buffer overflow protections used by the OS. Example implementations may be activated through compiler options such as "-fstack-protector-all", "-fstack-protector", and "/GS" flags. These are referred to by a variety of terms, such as stack cookie, stack guard, and stack canaries. The TSS must include a rationale for any binaries that are not protected in this manner.

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.

31.2. Documentation review activity

31.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.6.3 Protection From Implementation Weakness.** This section includes a list about the protections implemented by the TOE, e.g. Data Execution Prevention (DEP) or Address Space Layout Randomization (ASLR).

In addition, the TSS states that all Windows binaries and Windows Store Applications implement stack buffer overrun protection by being complied with the /GS option, and checking that all Windows Store Applications are compiled with buffer overrun protection before ingesting the Windows Store Application into the Windows Store except the following files:

- winload.exe
- winresume.exe
- ntoskrnl.exe for x64 architectures

winload.exe and winresume.exe are loaded before the stack buffer overrun protection mechanism is operational and therefore are not compiled with this option. The 64-bit version of ntoskrnl.exe Kernel Patch Protection mechanism is implemented in a static library, filter.obj. The implementation of the Patch Protection Mechanism is incompatible with /GS and therefore filter.obj is not compiled with /GS.

31.2.2. Verdict

The evaluator considers that the TSS provides enough information related to the stack-base overflow protections implemented by the TOE.

On the other hand the TSS has indicated that all windows binaries and Windows APPs have been complied with the /GS option except three files (winload.exe, winresume.exe and ntoskrnl.exe). A rationale explaining why these files have been not compiled with /GS option has been provided

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

31.3. Test Activity

31.3.1. Test 1

31.3.1.1. Setup

The evaluated platforms shall have installed the BinScope tool, in order to allow the evaluator analyzes the binaries to find out whether they have been compiled with stack-based overflow protections or not.

In order to avoid errors during the test execution, the evaluator shall have access to the private symbols. Due to this fact, this test shall be performed at Microsoft facilities.

Additionally, the evaluator shall use the following three tools:

- The first one is a PowerShell script developed by the evaluator. This script creates a file text with a list, including all system executables, all libraries, and all drivers stored in the system folder ***%windir%\System32*** directory. The source code of this script is as follows:

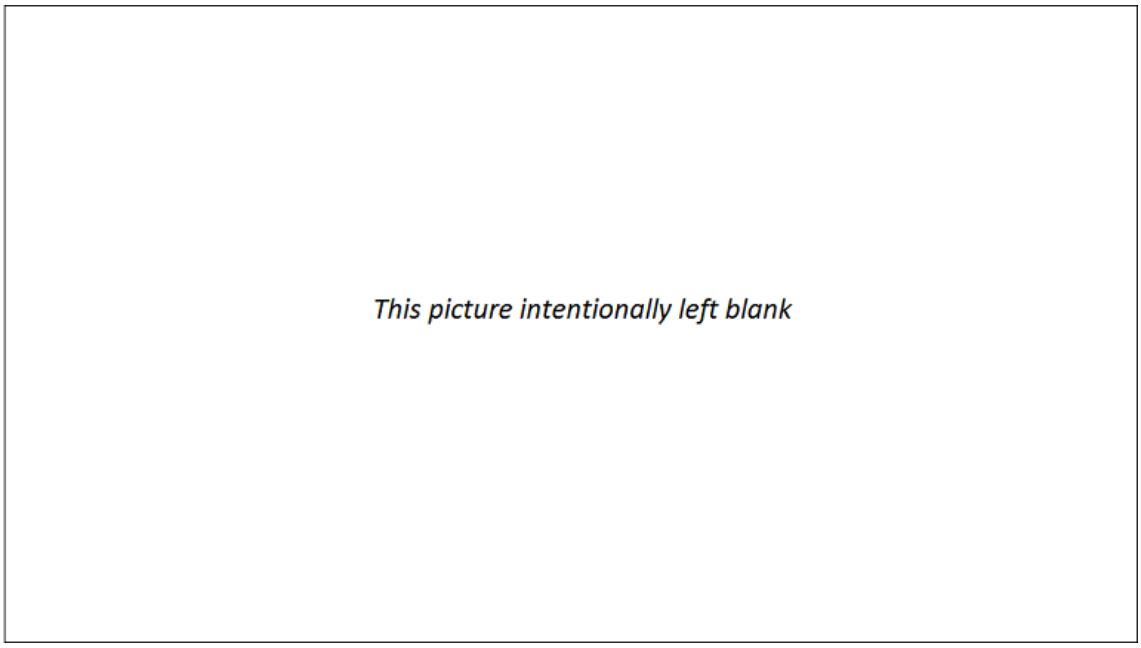
```
$kernelPath = "$env:SystemRoot\system32"
$kernelFiles = Get-ChildItem -Path $kernelPath -File -Recurse -Include *.com,*.exe,*sys,*dll -ErrorAction SilentlyContinue -ErrorVariable errors
foreach ($err in $errors){
    Write-Warning $err[0].Exception.Message
}

$file = New-Item -ItemType File -Path "C:\" -Name GSFiles.txt
foreach ($kernelFile in $kernelFiles){
    try{
        Write-Output $kernelFile.FullName | Out-File $file.FullName -Append
    } catch [Exception]{
        Write-Warning $Error[0].Exception.Message
    }
}
```

- The second tools is other PowerShell script, but this time provided by the vendor. This script allow the evaluator execute BinScope over a list of files. This list could be obtained using the above script. The script source code is as follows:

```
$startTime = get-date
$Env:Path += ";C:\Program Files (x86)\Microsoft\SDL BinScope"
$Env:_NT_SYMBOL_PATH = "SRV*http://symweb"
Get-Content SystemFileListSrvDataCenter.txt | Foreach-Object {binscope.exe /c GSCheck /d output /target bin\$_} > results.txt
$(get-date) - $startTime
```

- The last one is a Visual C# executable, which parses all the xml files generated for each analyzed binary. These reports contain too much information and the main purpose of this program is to extract the final verdict, indicating whether the file has been

compiled with /GS flag or not. After all the results are collected, it creates a CSV (Comma-Separated Values) with the final verdict for each file. The following screenshot show the main source code of this program:



31.3.1.2. Procedure

The following steps shall be performed in order to complete this testing assurance activity:

1. List all the kernel drivers and modules (*.sys extension), libraries (*.dll extension) and application binaries (*.exe and *.com extension) of the TOE. These files are stored under the system folder, and it could be done using the script developed by the evaluator (*Get-GSFiles.ps1*)

2. Check which of these files have not been compiled with stack-based buffer overflow protection. In order to do this, the evaluator shall use the script provided by the vendor, using as input the list obtained in the step 1. This script executes the BinScope command over the input list. In order to execute the BinScope command successfully, the script shall have access to the private symbols.

3. Once the BinScope analysis has finished, the evaluator shall execute the Visual C# file executable in order to parse all the generated report. The path folder where all partial reports are stored must be specified.

4. Open the .csv file generated as output in the step 3 and check the files which have not been compiled with /GS flag.

31.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- HP Pro X2 with Windows 10 x64 Pro Edition
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-v with Windows 10 x86 Home Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

After analyzing the report for each platform, the evaluator has observed that the files which have not been compiled with /GS flag are the same for all the evaluated platforms. These files are the following:

358	ntoskrnl.exe	ResultFAIL
359	winload.exe	ResultFAIL
360	winresume.exe	ResultFAIL

Except, for “Windows Server 2012 R2 Hyper-v with Windows 10 x86 Home Edition“ platform, where the result is as follows:

90	winload.exe	ResultFAIL
91	winresume.exe	ResultFAIL

31.3.1.4. Verdict

The obtained results have been the same as expected according to the TSS definition.

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

31.4. Final Verdict

Due to all activities 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 this FPT_SBOP_EXT.1.1 requirement.

32. FPT_SRP_EXT.1.1

32.1. Assurance activity

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.

32.2. Documentation review activity

Assurance activity does not state any documentation review activity.

32.3. Test Activity

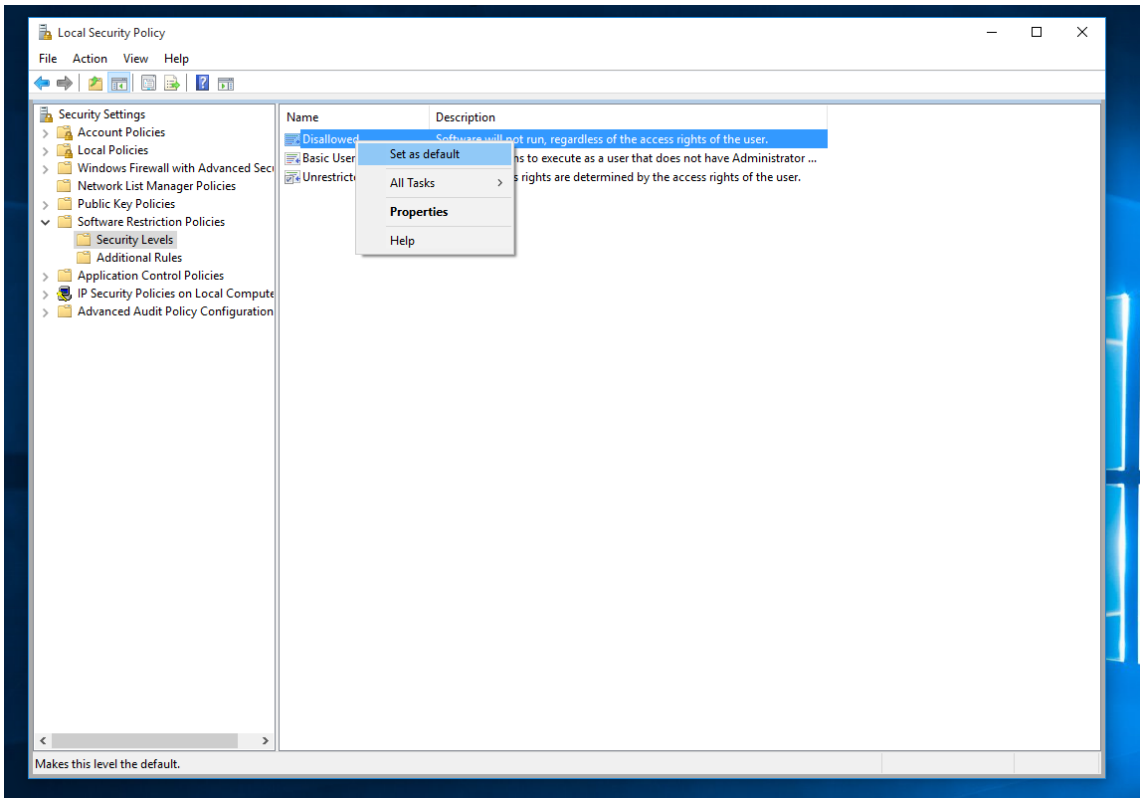
32.3.1. Test 1 & Test 2

32.3.1.1. Setup

Software restriction policy must be enabled. To do that, the evaluator shall carry out the next steps.

1. Open a command line terminal with administrator rights and type "secpol" to open the *Local Security Policy* window.

2. Then, go to *Security Settings -> Software Restriction Policies*. Right-click and select *New Software Restriction Policies*.



3. After that, go to Security Levels, right-click in Disallowed and select Set as default option.

4. Restart to apply the changes.

Once the software restriction policy is enabled, two default path rules are applied. These rules only allow code execution from the *%SystemRoot%* (system32) and *%ProgramFilesDir%* (ProgramFiles) directories.

32.3.1.2. Procedure

To ensure that the rules configured during the setup are applied properly, the evaluator shall perform the following steps.

1. Go to system32 directory and select one executable file, e.g. cmd.exe. Create a shortcut and send it to the desktop.

2. After that, attempt to execute the executable file stored in system32 directory. The file shall be executed properly.

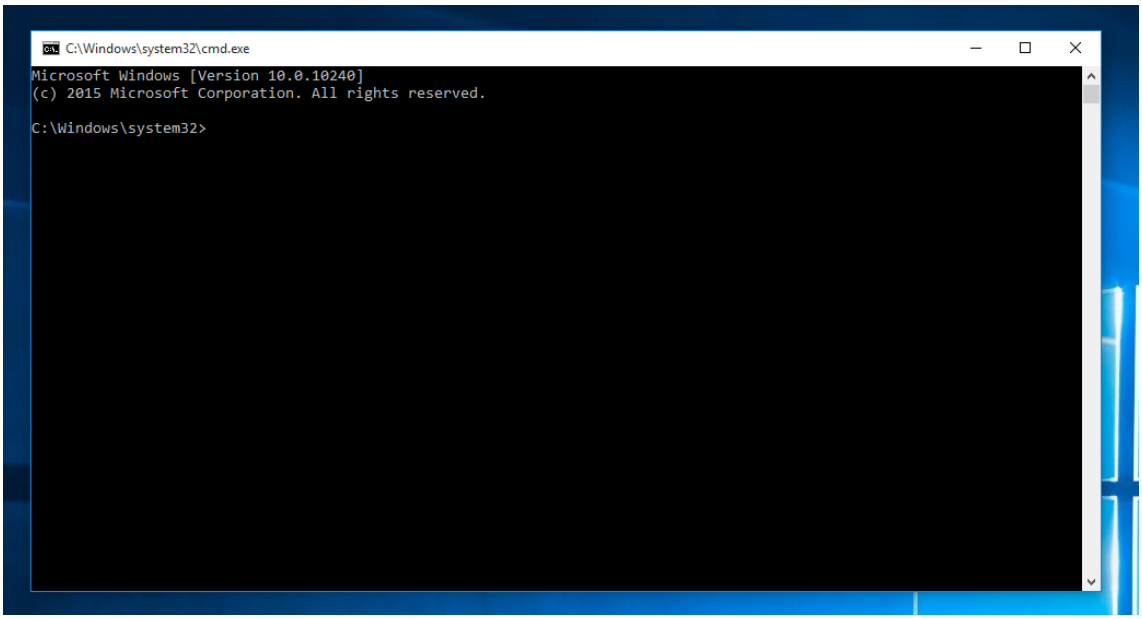
3. Finally, attempt to execute the shortcut created in the step 1. The file shall not be executed and an error message shall be shown.

32.3.1.3. Results

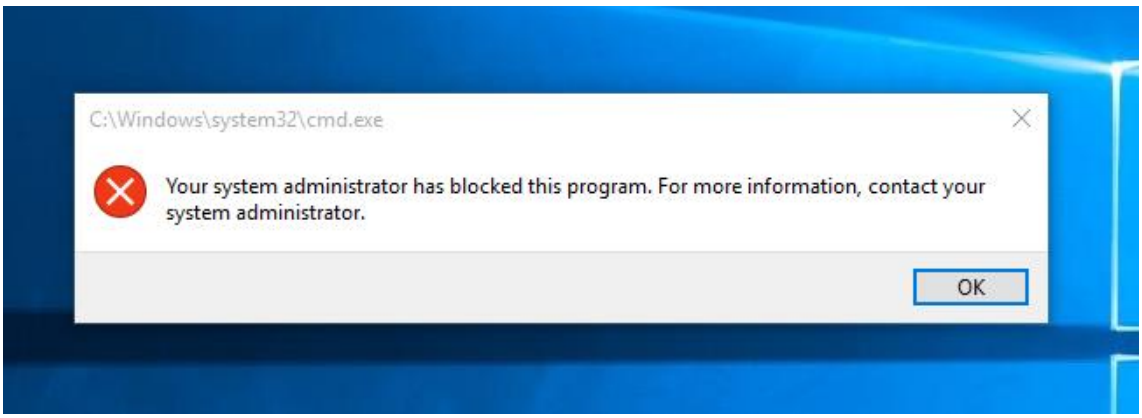
The evaluator has performed these tests in the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same results for all the tested platforms, which they are explained as follows. The evaluator has attempted to execute an executable file from the system32 directory, and it has been executed in a correct way.



On the other hand, the evaluator has attempted to execute the shortcut stored in the desktop, and it has not been executed. Moreover an error message has been shown, indicating that the system administrator has blocked the execution of this program.



32.3.1.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 1** and **Test 2** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 1** and **Test 2**.

32.3.2. Test 3 & Test 4

32.3.2.1. Setup

The applicable setup for this test is the same as for Test 1 and Test 2.

Additionally, the following executables files, which will be used during the tests execution, must be downloaded into the target machine:

- Firefox Setup 41.0.2.exe
- MediaCreationToolx64.exe

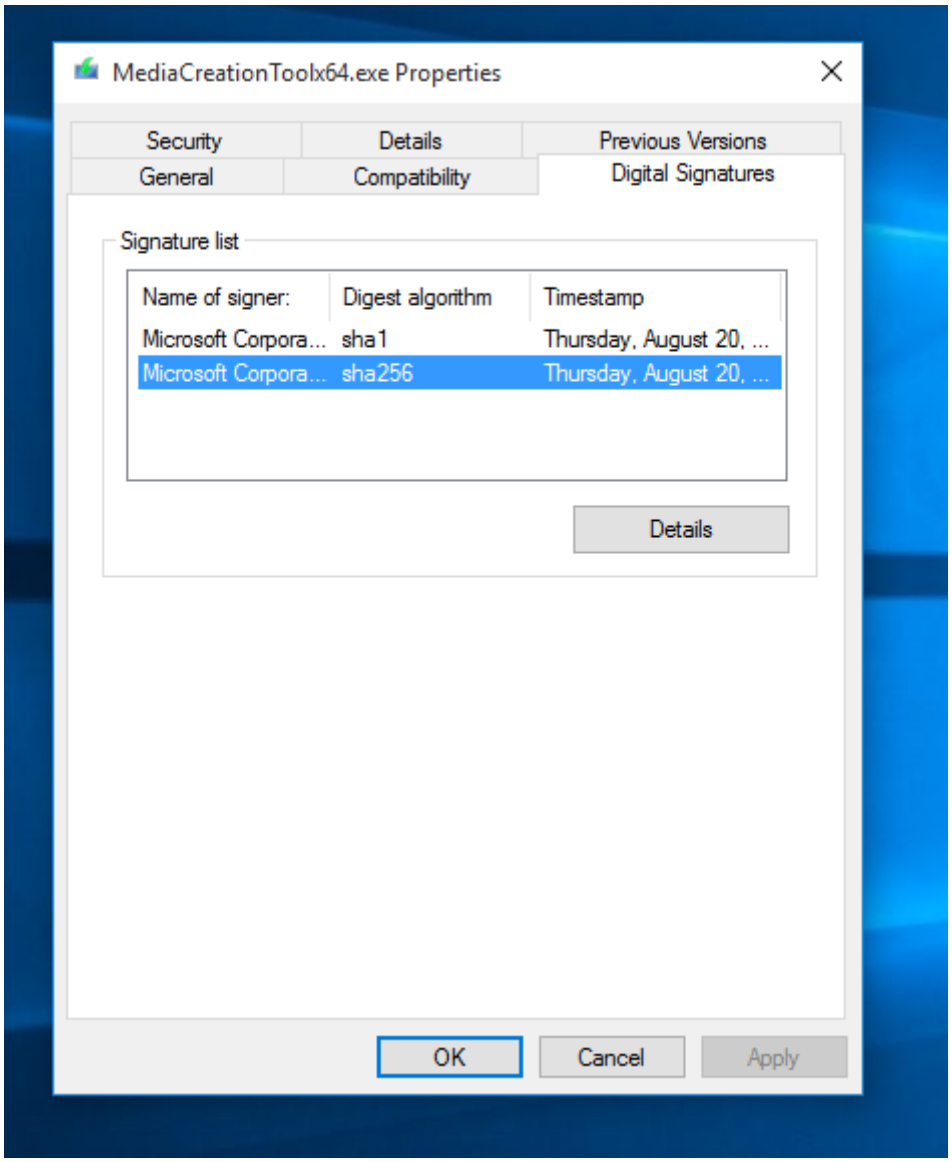
32.3.2.2. Procedure

The evaluator shall create a new software restriction rule, which only allow code execution that has been signed by the OS vendor.

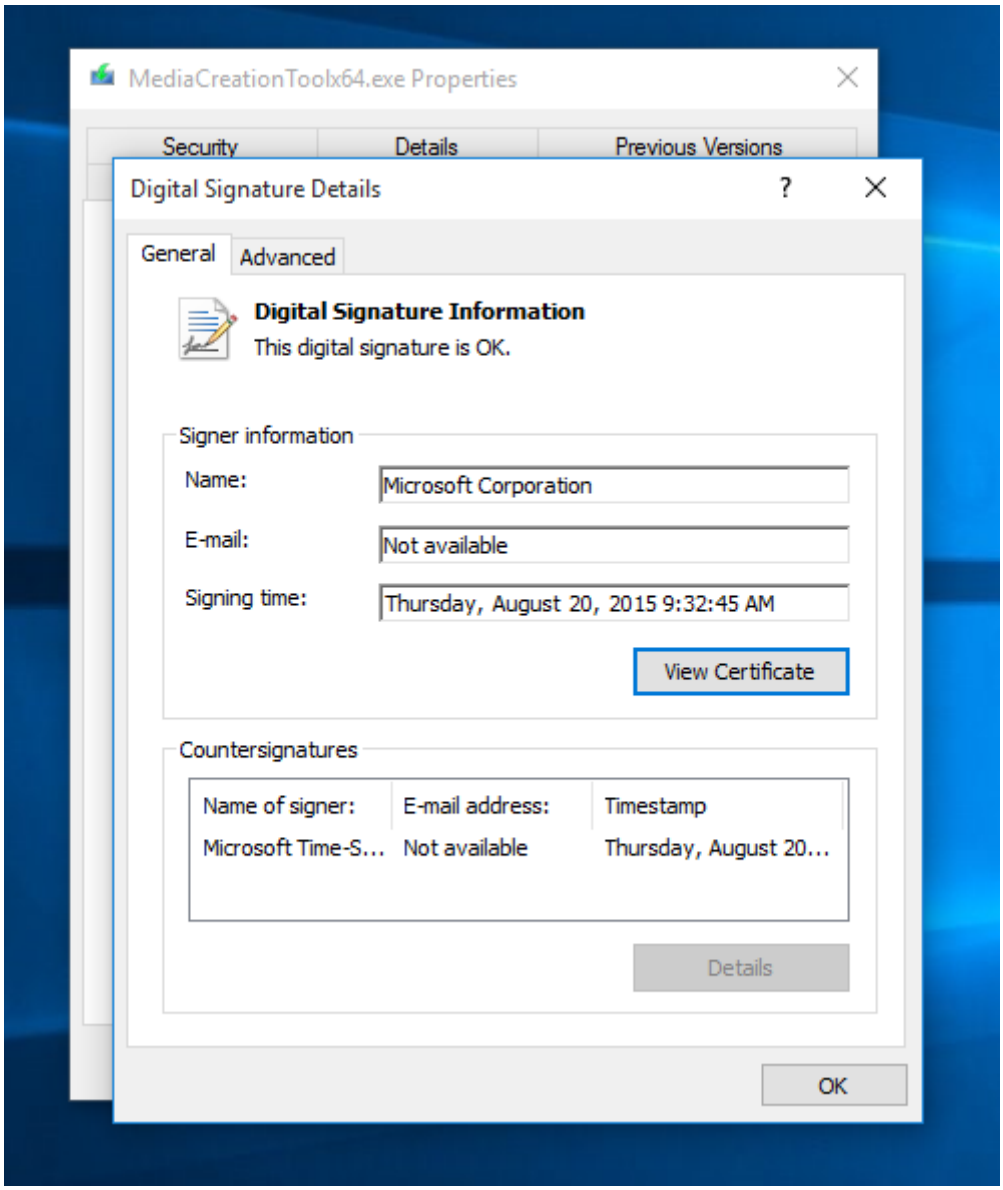
First of all, the evaluator shall obtain the certificate with which the file executable has been signed. To do that, the evaluator shall perform the following steps:

1. Go to the location where MediaCreationToolx64.exe is stored. Right-click over there and open the properties panel.

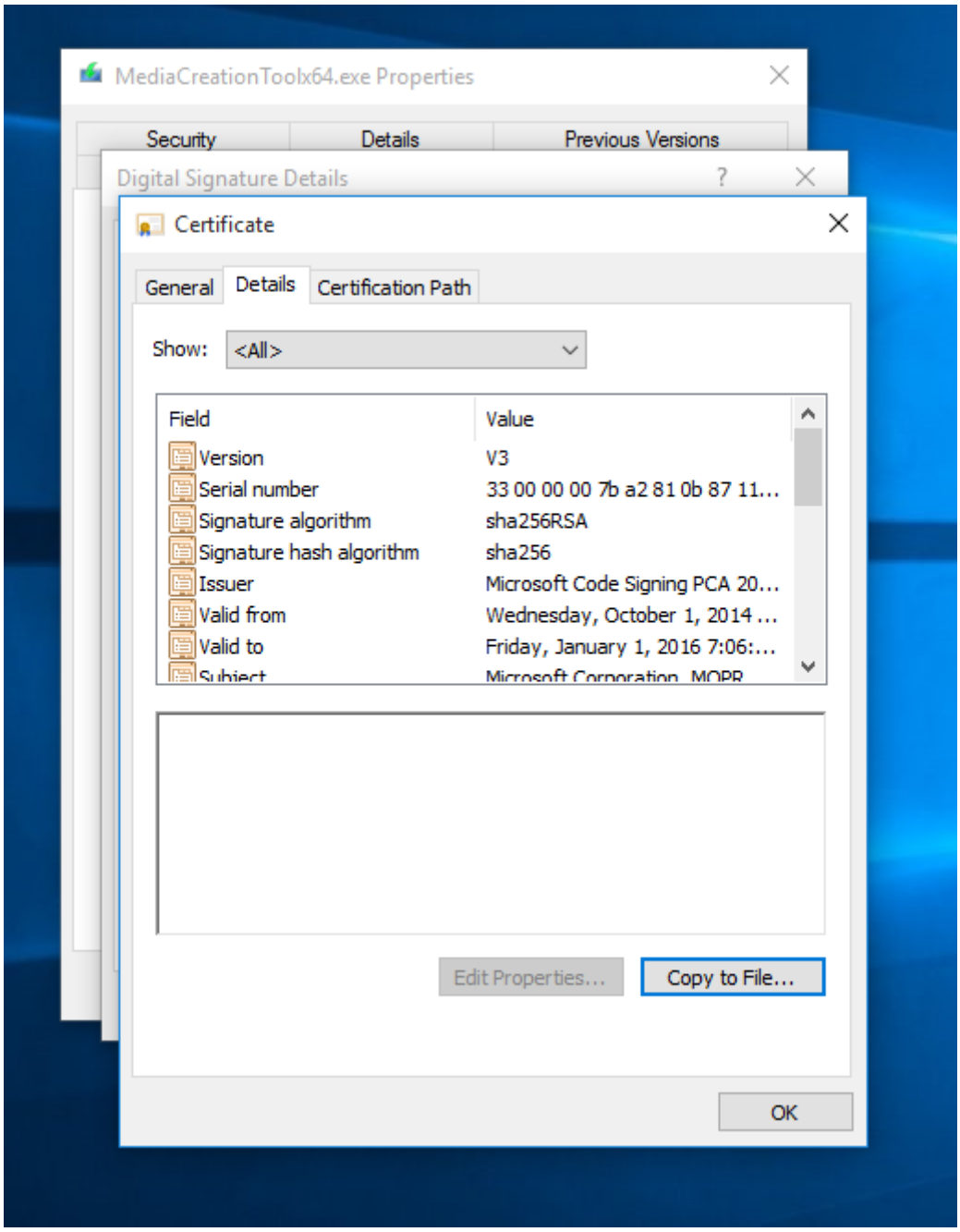
2. Select the *Digital Signatures* tab, and select one of the digital signatures of the file, e.g. SHA-256, and click over *Details* button.



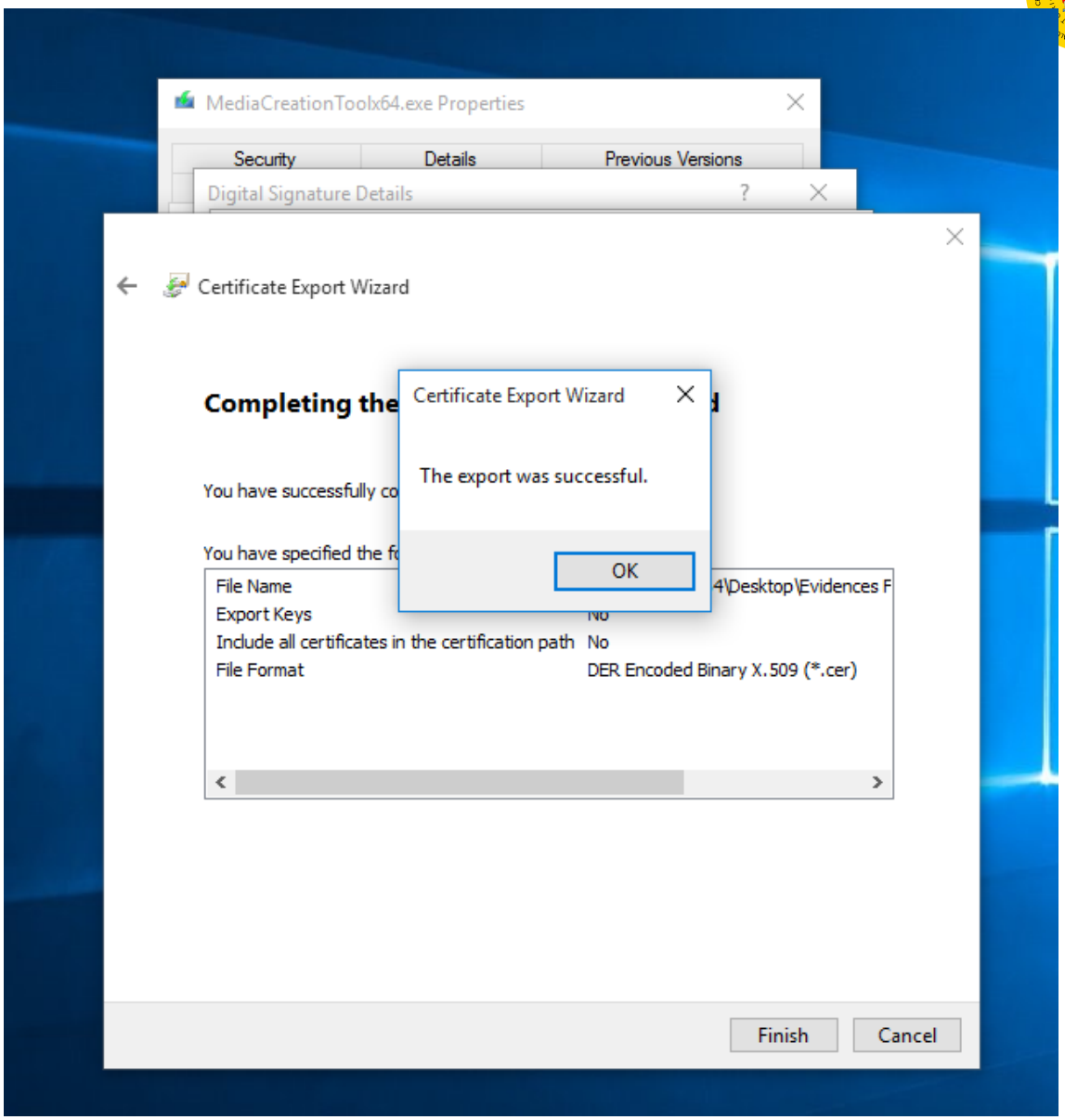
3. Click over *View Certificate* button, to view details about the certificate.



4. In the new opened window, select the *Details* tab and click over *Copy to file...* button.

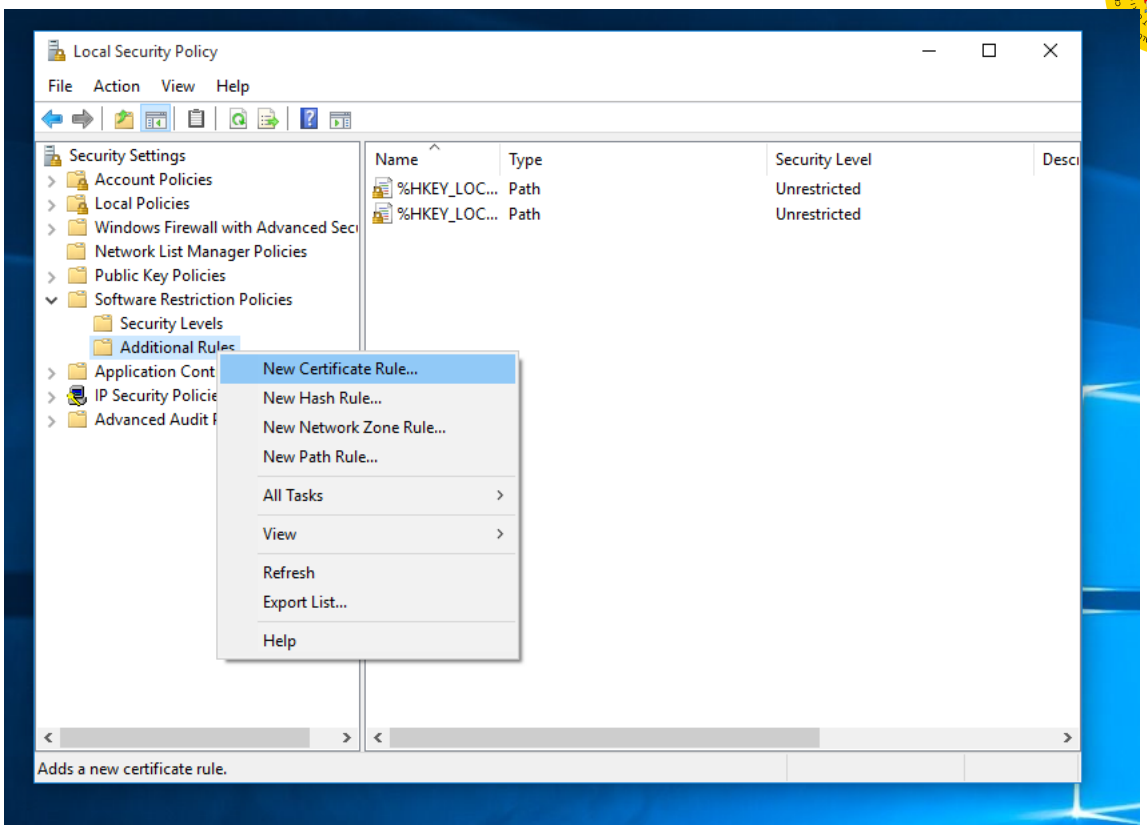


5. Follow the wizard and save the certificate using a DER encoded format.

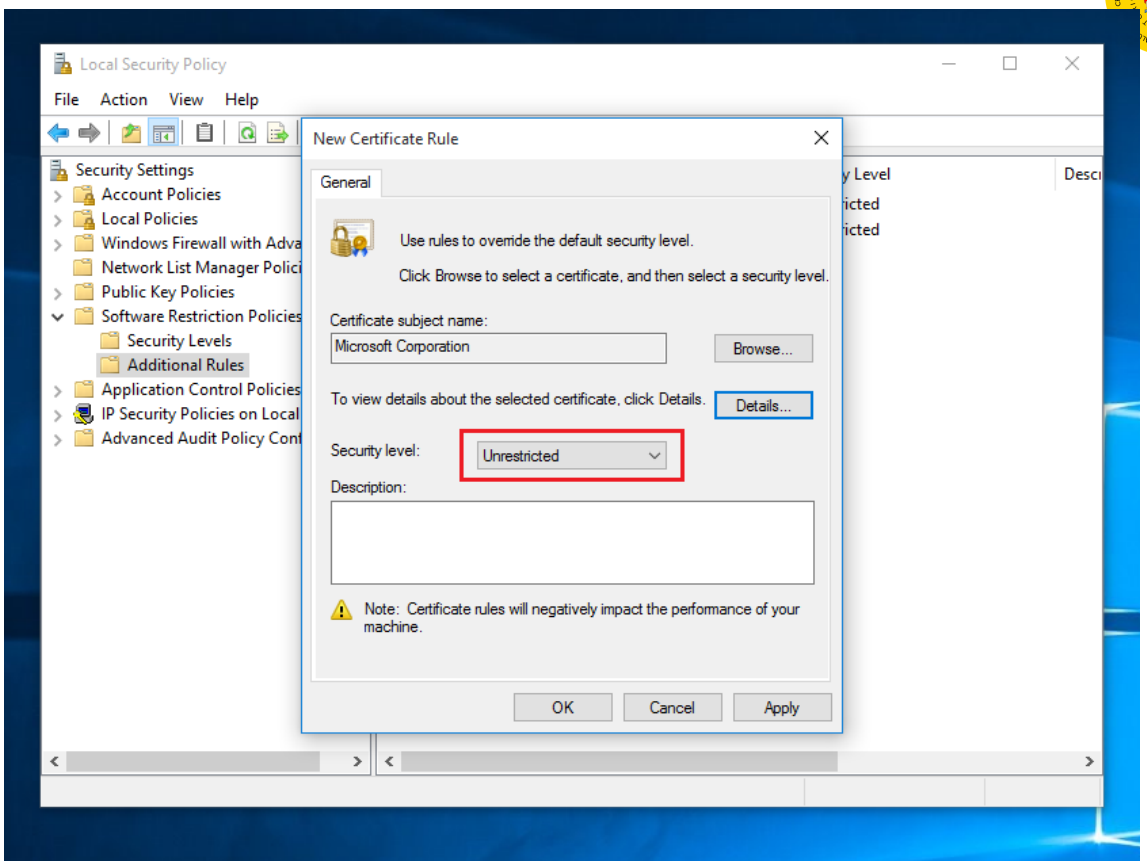


Once the certificate has been obtained, the evaluator shall carry out the next steps in order to configure a new certificate-based software restriction rule.

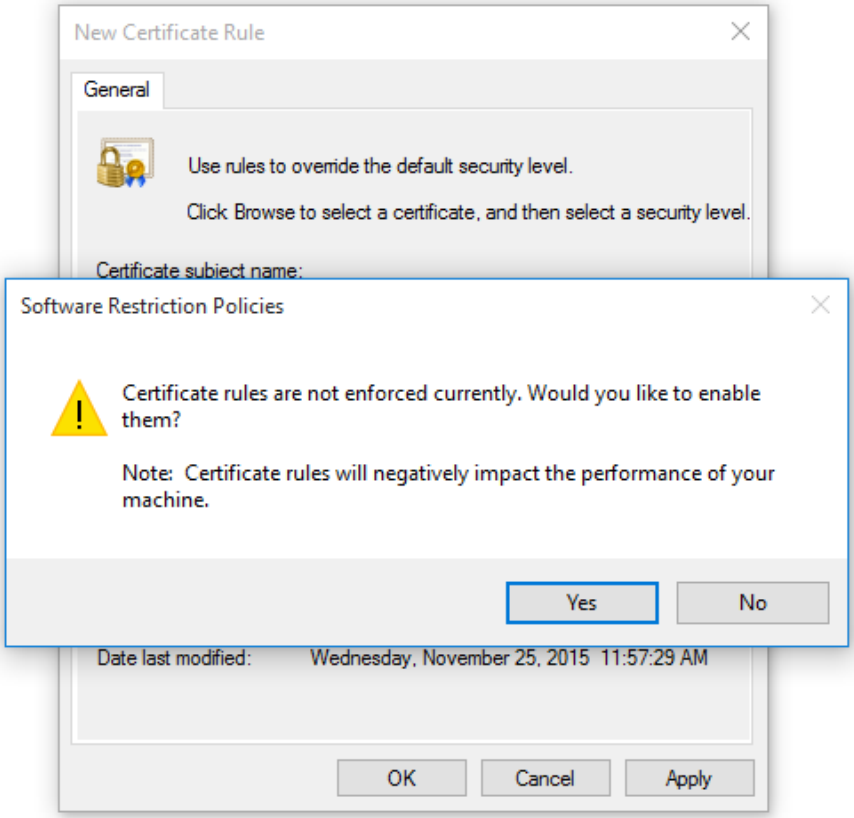
6. Open the *Local Security Policy*. Go to *Secure Settings -> Software Restriction Policies -> Additional Rules*. Right-click and select *New Certificate Rule...*

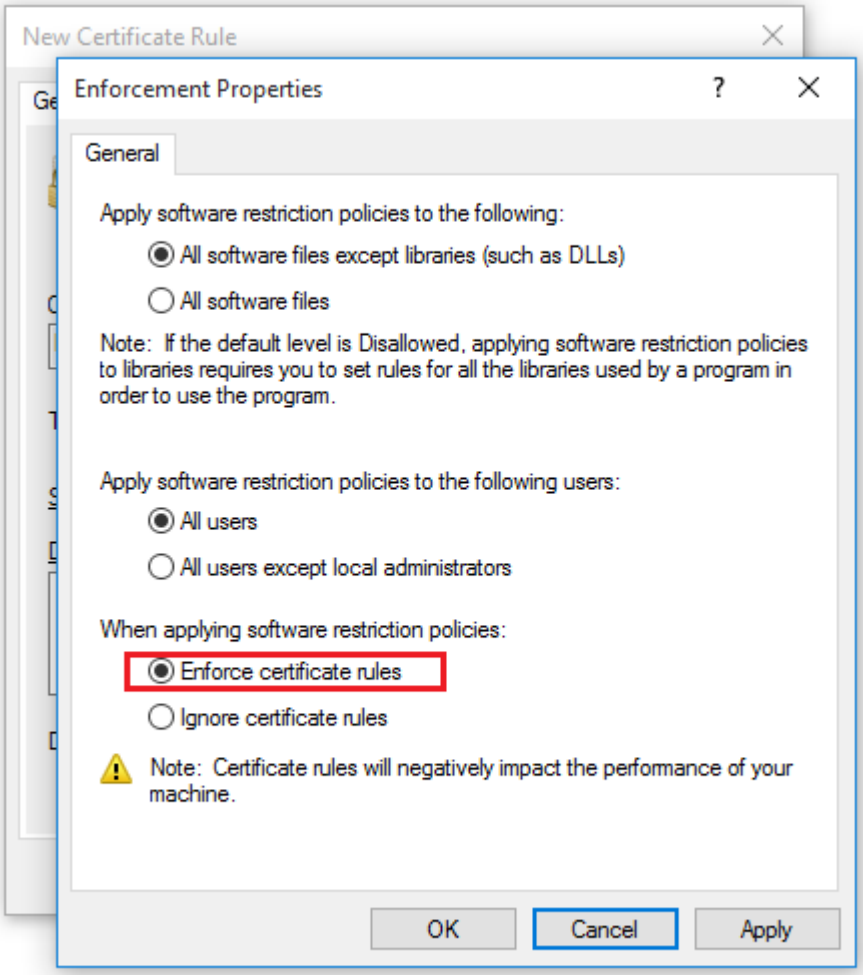


7. After that, select the certificate obtained in the step 8 and configure the *Security Level* option as *Unrestricted*.



8. Click over *Apply* button, and in the new opened window select the option *Enforce certificates rules* and select *Yes* option when the following prompt is shown.





9. Restart to apply the changes.

After the reboot, the evaluator shall perform the following steps to ensure that the created rule is applied properly.

10. Go to the location where the executable files are stored and attempt to execute MediaCreationToolx64.exe. The file shall be executed correctly.

11. After that, attempt to execute Firefox Setup 41.0.2.exe file directory. The file shall not be executed and an error message shall be shown.

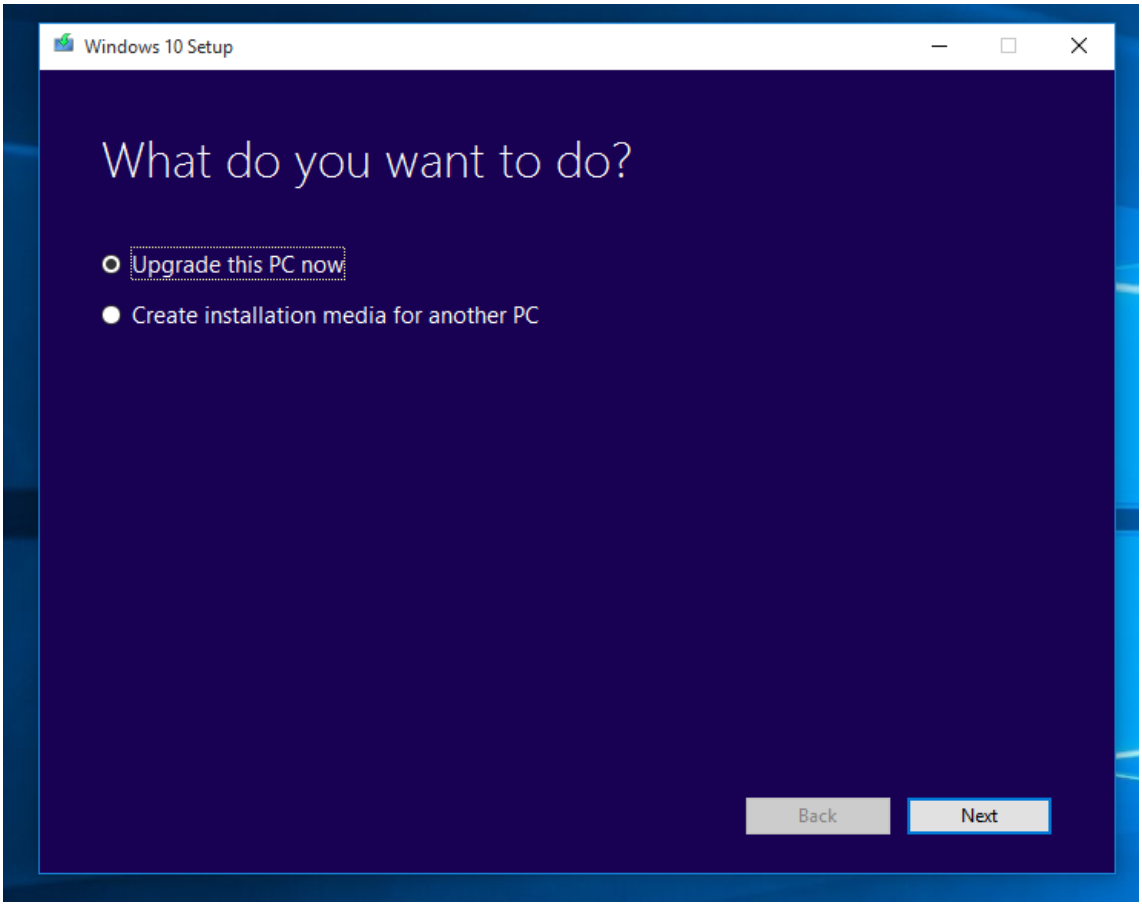
32.3.2.3. Results

The evaluator has performed these tests in the following evaluated platforms:

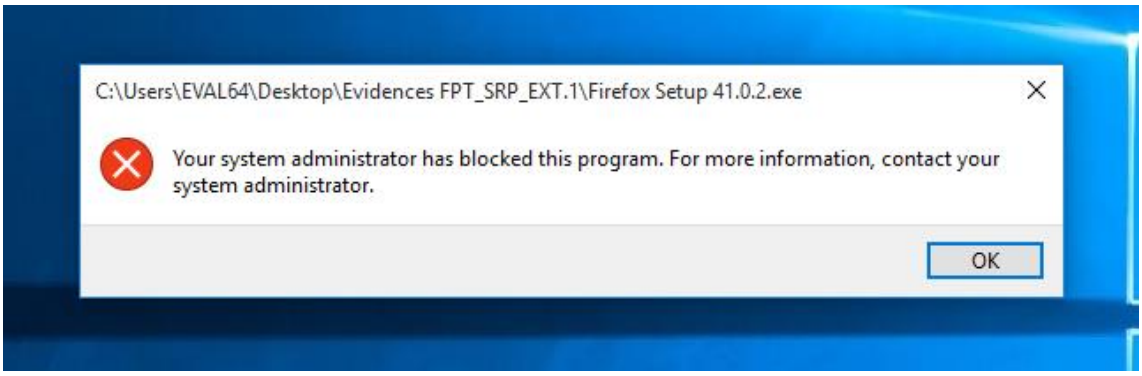
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.

- Surface 3 with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same results for all the tested platforms, which they are explained as follows. The evaluator has attempted to execute an executable file, which has been signed by the OS vendor, and it has been executed in a correct way.



On the other hand, the evaluator has attempted to execute an executable file, which has not been signed by the vendor, and it has not been executed. Moreover an error message has been shown, indicating that the system administrator has blocked the execution of this program.



32.3.2.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 3** and **Test 4** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 3** and **Test 4**.

32.3.3. Test 5 & Test 6

32.3.3.1. Setup

The security target includes the following footnote in the SFR definition which states that the software restriction version-based rules are only applicable for Windows 10 Enterprise Edition and Windows Server 2012 R2.

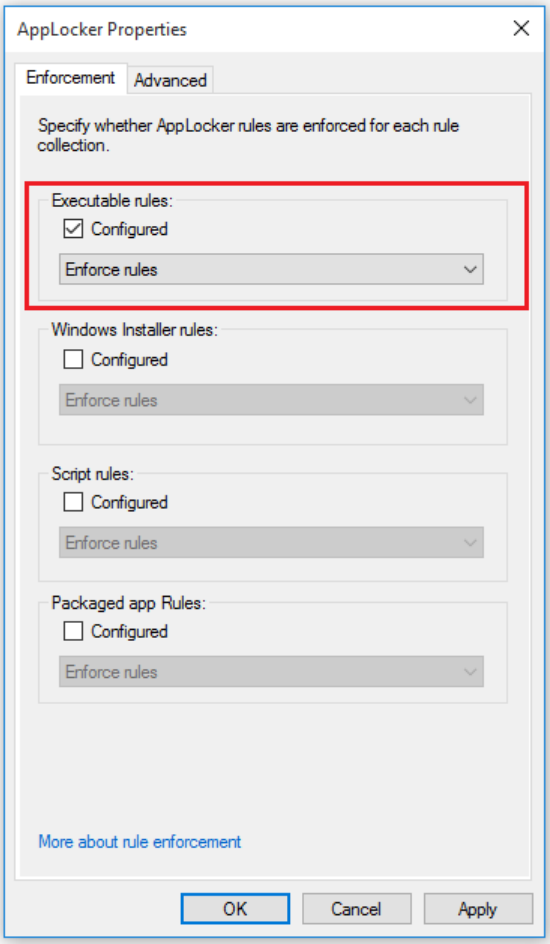
[7] Windows 10 Enterprise and Windows Server 2012 R2 can restrict program execution based on a version; Windows 10 Pro and Windows 10 Home editions cannot.

Therefore these tests are only applicable for:

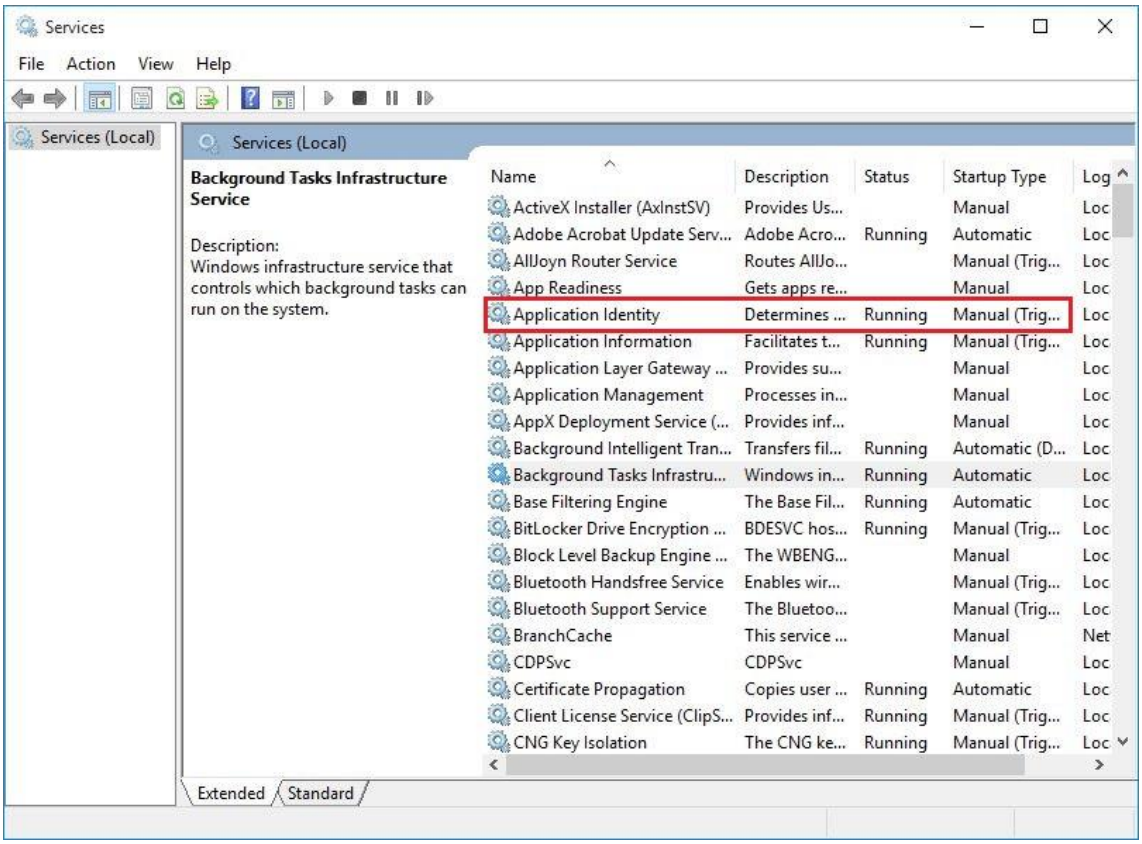
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 Datacenter Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Pro 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

Due to these tests shall be performed using AppLocker, the next setup conditions must be fulfilled:

- AppLocker shall be configured in order to enforce the executable rules. To do that open Local Security Policy tool and go to *Application Control Policies->AppLocker*. Right-click in it, select the *Properties* option and configure it as follows:



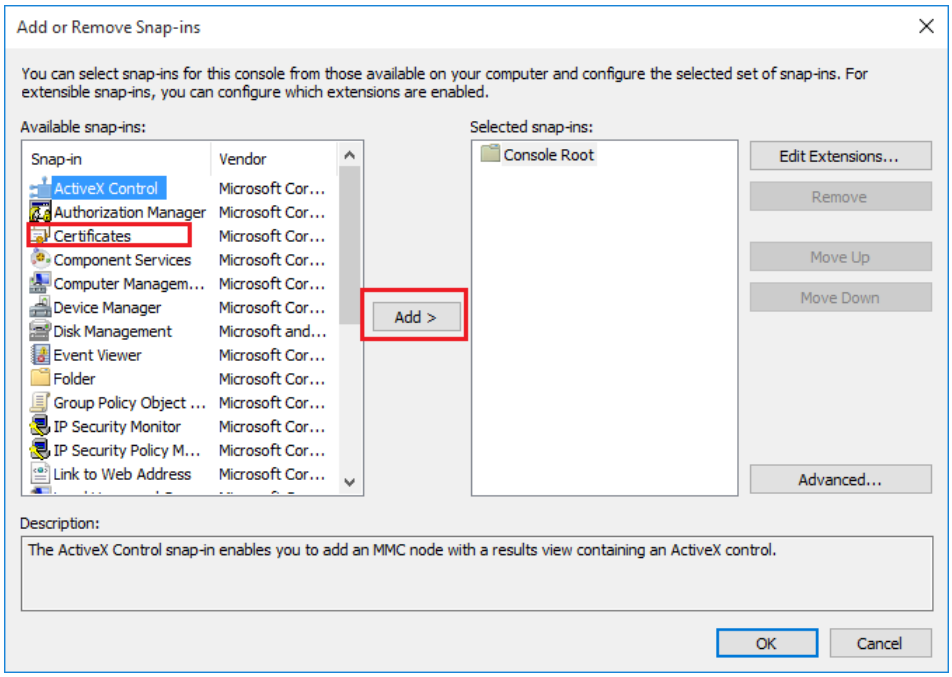
- Application Identity service must be running. The evaluator shall the next steps in order to ensure that the service is already running or start it in case of the service is stopped:
    - Right-click in Start button and select Run option. Then type "Services".
    - Check that the Application Identity service is running. In case of the service is stopped, right-click over it and then select Start option

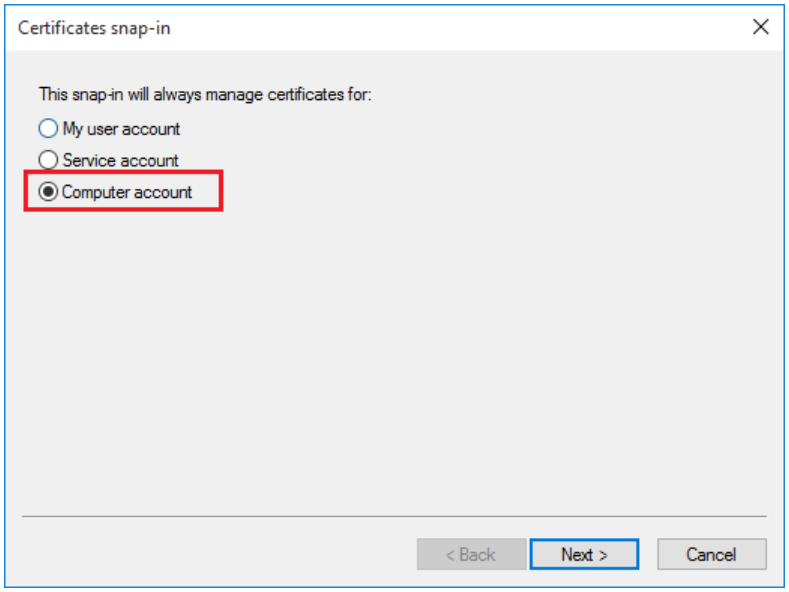


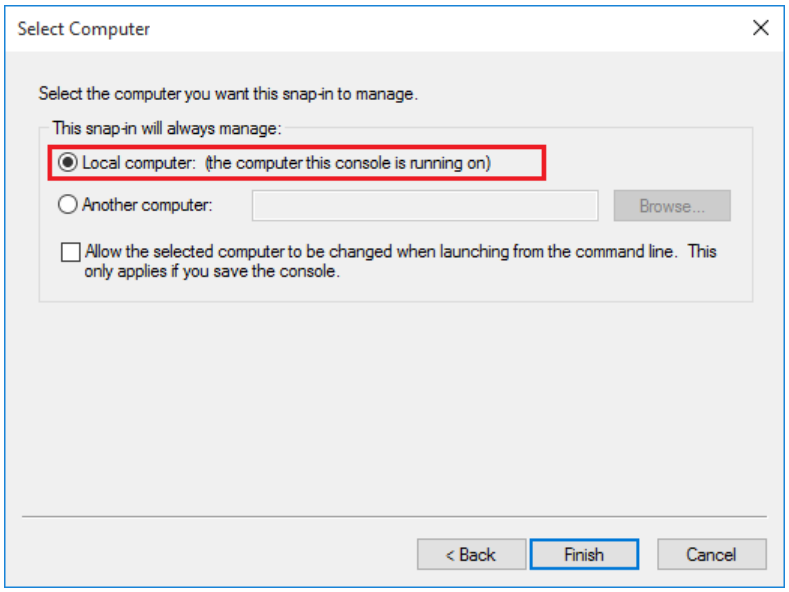
- During the test execution a computer rebooting may be required and due to this the Application Identity service shall be launched again. To avoid this, the Application Identity service can be configured to start automatically. To configure it, right-click in the services, select the Properties option and configure the *Startup type* as "Automatic"

- The following executables provided by the vendor shall be available in the evaluated platform. These executable files are signed by a test certificate issued by a test root:
  - TestConsolev10.exe
  - TestConsolev11.exe

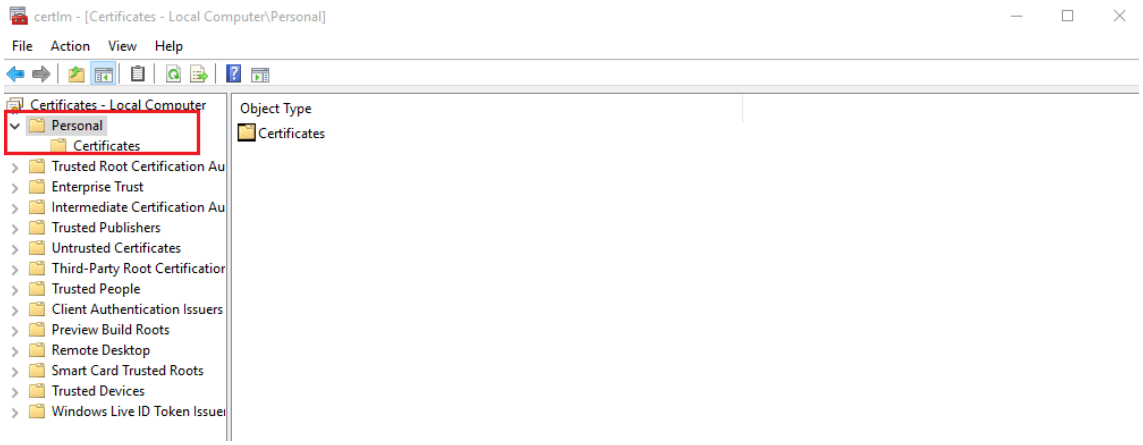
- In order to the version-based rule can be applied correctly, the evaluator shall include both test certificate and test root into the Personal certificate store and the Trusted Root Certification Authorities store. To do that, the evaluator shall carry out the following steps:
  - Right-click in Start button and select Run option. After that type "mmc" in order to open the Microsoft Management Console tool.
  - Go to *File->Add/Remove Snap-in...* in order to add the Certificates snap-in to the viewer. To do this configure the wizard as the following images show:

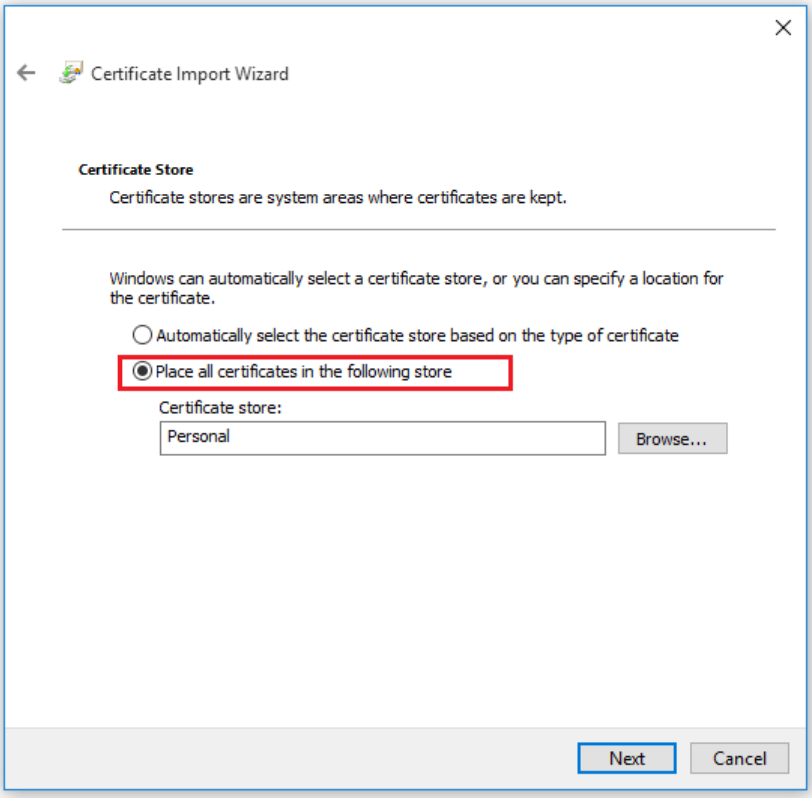




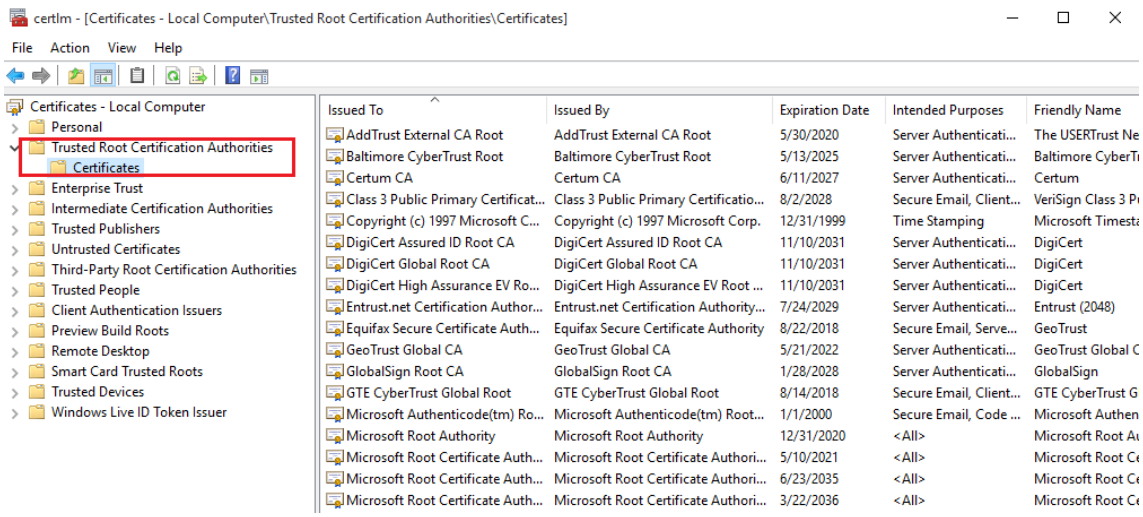


- Go to *Personal->Certificates*. Then, right-click in *Actions->All Tasks->Import...* and follow the wizard in order to import the test certificate.





- After that, go to *Trusted Root Certification Authorities->Certificates* and repeat the above test to import the test root.



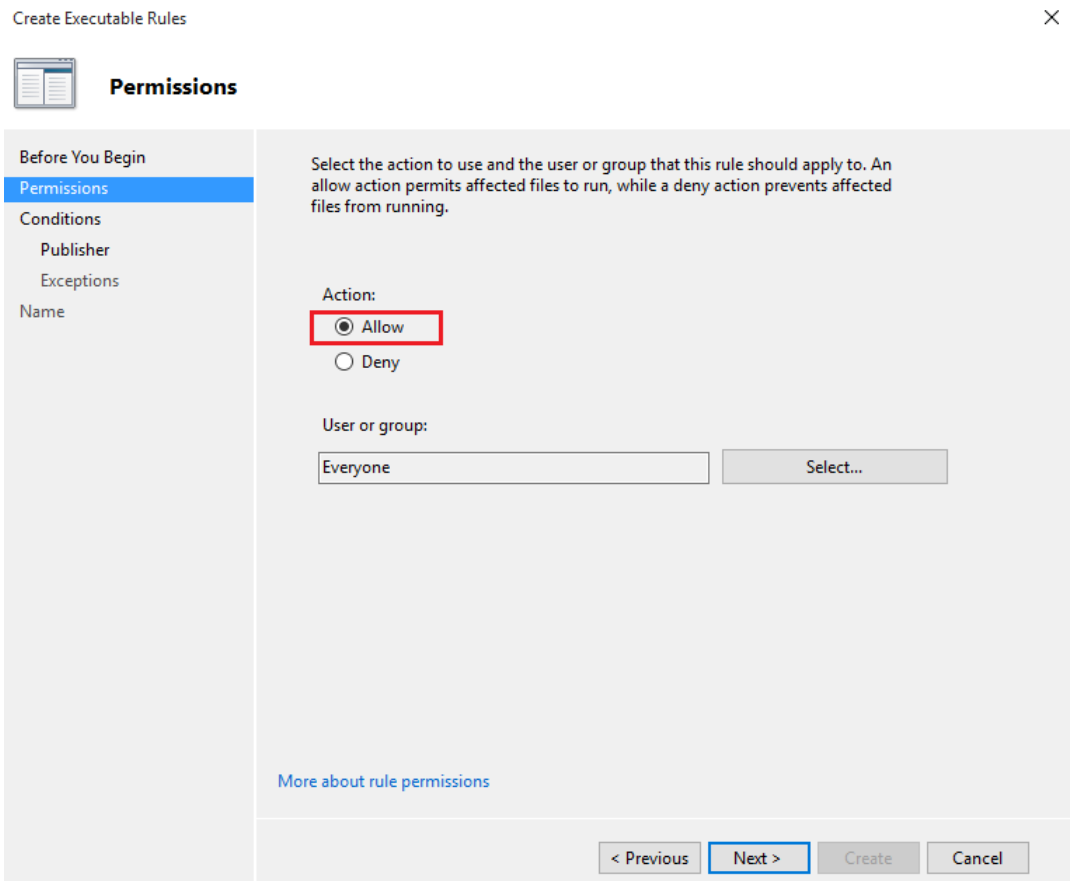
32.3.3.2. Procedure

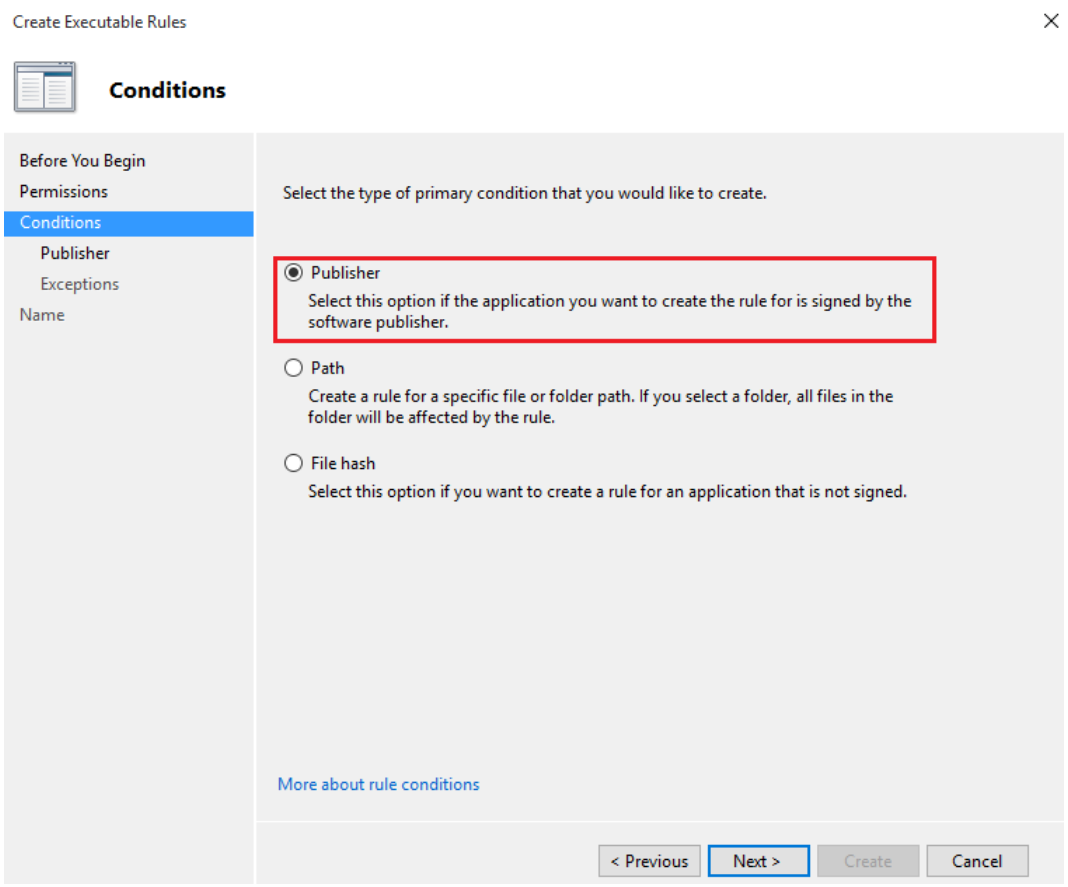
The evaluator shall create a new version-based software restriction rule. To do that, the evaluator shall carry out the following steps.

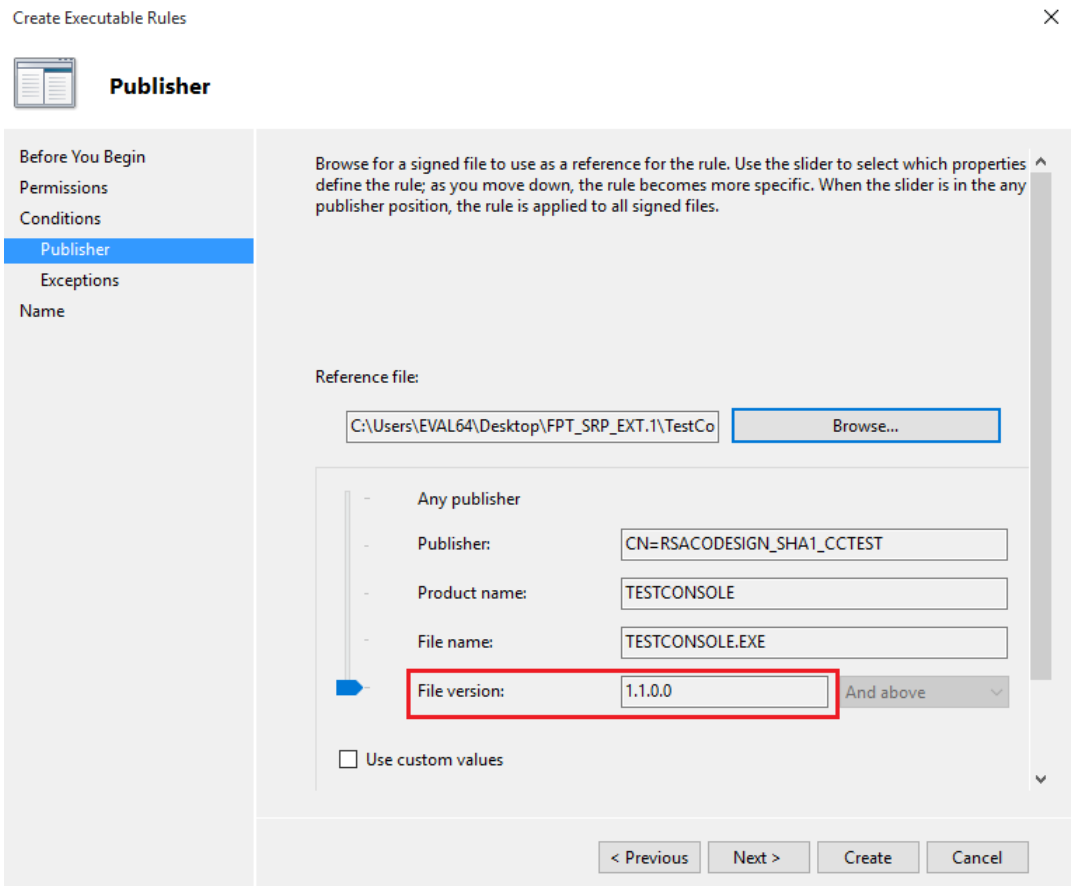
1. Open the Local Security Policy tool and go to *Application Control Policies->AppLocker->Executable Rules*.

2. Right-click over it and select Create New Rule... option. A new wizard window shall be opened.

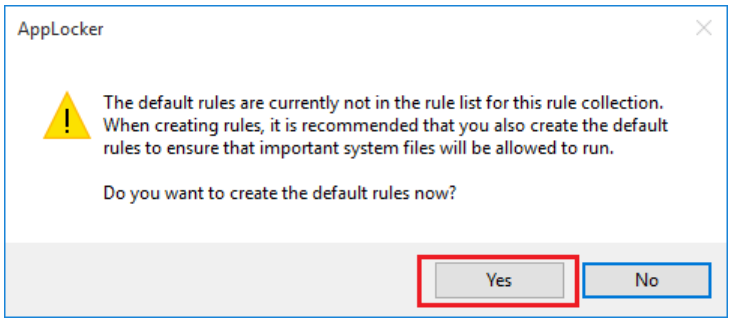
3. Configure the new version-rule to allow the execution of the executable file with version 1.1 as the following images show:



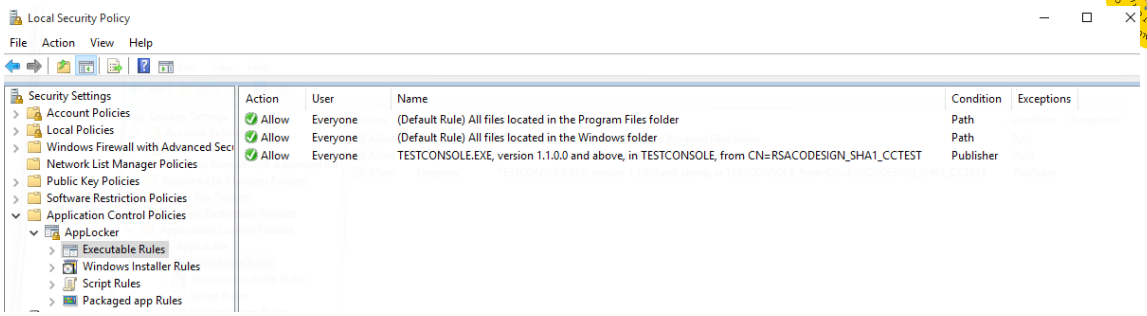




4. Before creating the rule, the following dialog shall be shown asking whether the default rules shall be created or not. Select the option *Yes*.



5. Finally, the defined executable rules in AppLocker shall be the following:



6. After that, attempt to execute TestConsolev11.exe. The file shall execute successfully.

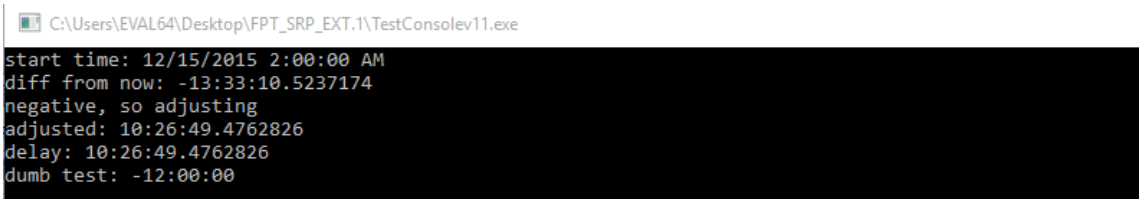
7. Finally, attempt to execute TestConsolev10.exe. AppLocker shall block the program execution and a message shall be shown indicating this information.

32.3.3.3. Results

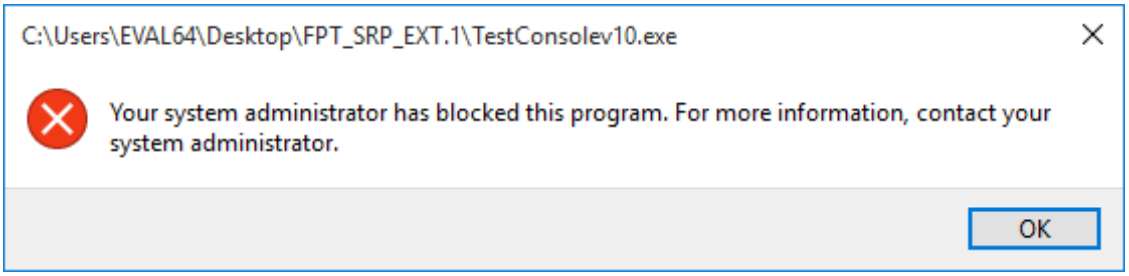
The evaluator has performed these tests in the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition

The evaluator has obtained the same results for all the tested platforms, which they are explained as follows. The evaluator has attempted to execute TestConsolev11.exe, which its version matches with the defined rule, and it has been executed in a correct way.



On the other hand, the evaluator has attempted to execute TestConsolev10.exe, which its version does not match with the defined rule, and it has not been executed. Moreover an error message has been shown, indicating that the system administrator has blocked the execution of this program.



32.3.3.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 5** and **Test 6** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 5** and **Test 6**.

32.3.4. Test 7 & Test 8

32.3.4.1. Setup

The applicable setup for this test is the same as for Test 1 and Test 2.

A hexadecimal editor must be installed in order to modify the original executable file.

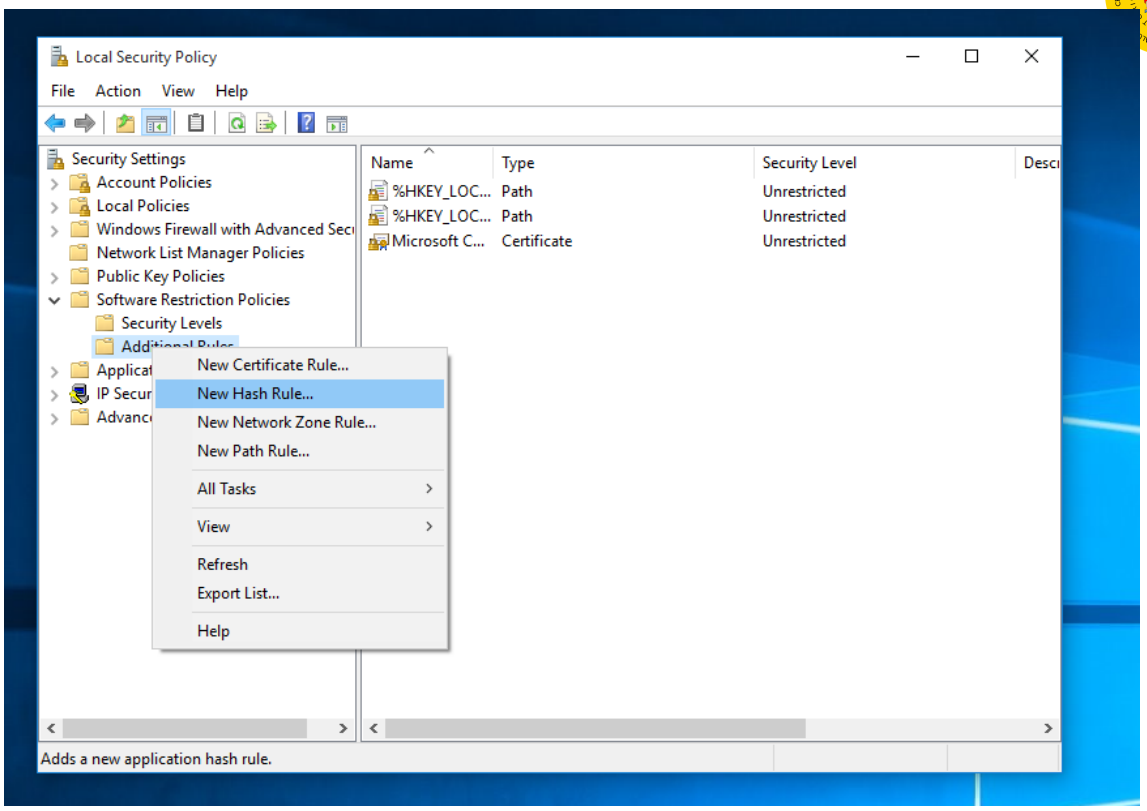
Additionally, the following batch script created by the evaluator will be used during the tests execution:

- test7&8.bat

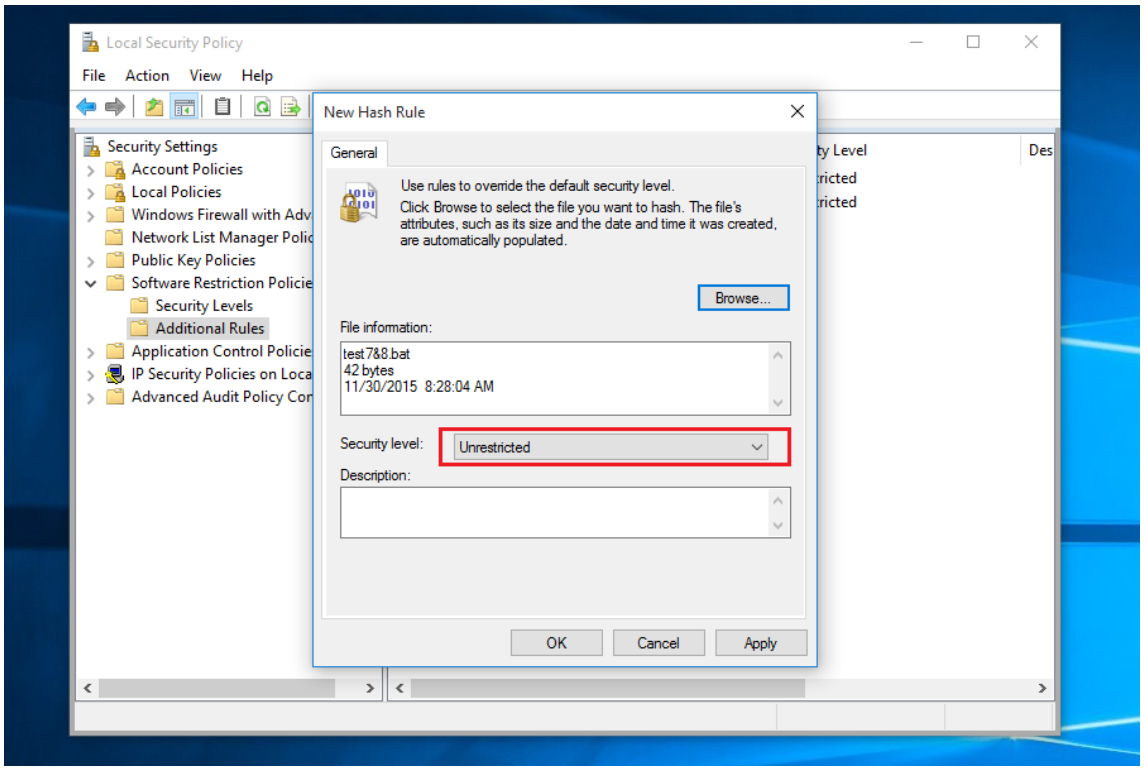
32.3.4.2. Procedure

The evaluator shall create a new hash-based software restriction rule. To do that, the evaluator shall carry out the following steps.

1. Create a copy of test7&8.bat, open it with a hexadecimal editor and modify some bytes.
2. Open the *Local Security Policy*. Go to *Secure Settings -> Software Restriction Policies -> Additional Rules*. Right-click and select *New Hash Rule...*



3. After that, select the original file test7&8.bat and configure the *Security Level* option as *Unrestricted*.



4. Click over *Apply* button.

5. Restart to apply the changes.

After the reboot, the evaluator shall perform the following steps to ensure that the created rule is applied properly.

6. Go to the location where the scripts files are stored and attempt to execute the original script. The script shall be executed correctly.

7. After that, attempt to execute the modified one. The file shall not be executed and an error message shall be shown.

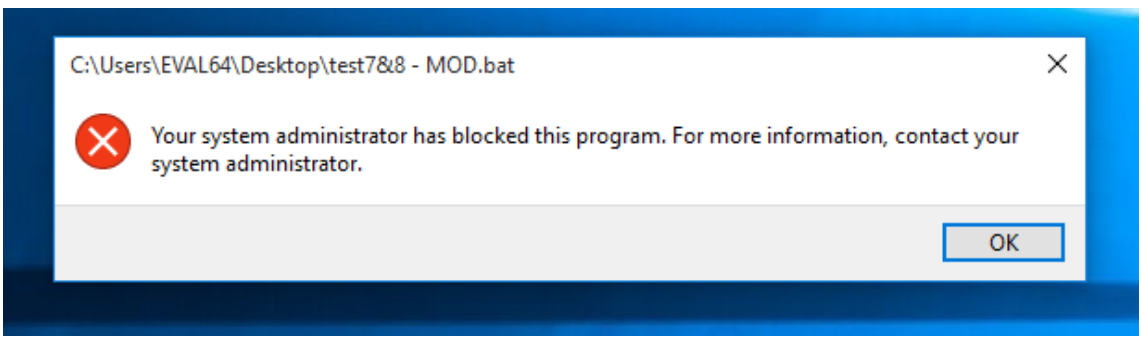
32.3.4.3. Results

The evaluator has performed these tests in the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition

The evaluator has obtained the same results for all the tested platforms, which they are explained as follows. The evaluator has attempted to execute a script file, which its hash matches with the defined rule, and it has been executed in a correct way.

On the other hand, the evaluator has attempted to execute a script file, which its hash does not match with the defined rule, and it has not been executed. Moreover an error message has been shown, indicating that the system administrator has blocked the execution of this program.



32.3.4.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 7** and **Test 8** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 7** and **Test 8**.

32.4. Final Verdict

Due to all test activities 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 FPT_SRP_EXT.1.1.

33. FPT_TST_EXT.1.1

33.1. Assurance activity

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. The evaluator will digitally sign the TSF executable with a certificate that does not have the Code Signing purpose in the extendedKeyUsage field and verify that an integrity violation is triggered. The evaluator shall repeat the test using a certificate that contains the Code Signing purpose and verify that the integrity verification succeeds. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

33.2. Documentation review activity

33.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.6.4 Windows Platform Integrity and Code Integrity**. This section describes the different stages of the boot process, providing information about what are the main files loaded in each step of the bootchain.

The evaluator has analyzed the information provided in the TSS section. This information has allowed the evaluator design a testing plan, which it will be used during the test activity. The evaluator has identified four different stages during the boot process, which are described as follows:

- 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 in during the test activity.

- 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 applications:

    - *winload.exe* or *winload.efi*

    - *winresume.exe* or *winresume.efi.* A footnote is included in the security target, which states that the hibernation is disabled in the evaluated configuration and so this boot application will not be used during the evaluation.

    - *memtest.exe.* A footnote is included in the security target, which states that memtest is considered to be a non-operational mode.

- 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 list of these files is included in the security target:

    - ntoskrnl.exe
    - bootvid.dll
    - ci.dll
    - clfs.dll
    - cng.sys
    - fvevol.sys
    - hall.dll
    - kdcom.dll
    - ksecdd.sys
    - pshed.dll
    - tpm.sys

- Fourth stage: After the critical device drivers and libraries have been loaded, the Windows kernel continues to boot the rest of the operating system.

Moreover, along this section the integrity validation mechanism is explained, including information about how the TOE validates each piece of software using a hash based signature and an embedded public key. In addition, the following information is included regarding the

Code Integrity, which verifies the integrity of the kernel driver 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. On the other hand, for 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 other unsigned-driver is detected during the boot process, the operating system may be loaded normally.

Finally, this section also states that in case of integrity failure during the boot process, the TOE may have the ability to restore the corrupt files from a recovery partition.

33.2.2. Verdict

The evaluator has reviewed the information provided in the security target and consider that, the information in TSS section is enough to allow the evaluator understand how the boot process works, the main files loaded during the bootchain or how the integrity of these files are checked depending on the platform arquitecture. Moreover, all the stages in the bootchain are clearly described, indicating which ones are out of the evaluation scope taking into account the evaluated configuration.

Therefore, 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.

33.3. Test Activity

33.3.1. Test 1

33.3.1.1. Setup

No special setup is required in order to perform this test.

33.3.1.2. Procedure

The evaluator shall attempt to perform any action (power on the machine) in order to cause the operating system boot. After that, the evaluator shall observe that the system boot successfully and there was not any integrity error.

33.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Pro Edition.
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.

- HP Pro x2 612 with Windows 10 x64 Pro Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition.

The evaluator has obtained the same result for all the tested platforms, the operating system boot properly and it does not flag any integrity errors.

33.3.1.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 1** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 1**.

33.3.2. Test 2

33.3.2.1. Setup

A hexadecimal editor, e.g. WinHex, must be installed in the evaluated platforms in order to modify the binary files loaded during the boot process.

A WinPE USB for both architectures (x64 and x86) must be available.

33.3.2.2. Procedure

During the documentation review activity, the evaluator has identified four different stages during the boot process. Due to the first one involves the TPM, which is part of the external TOE environment, it will not be tested.

The evaluator shall modify one binary file for each stage, and shall attempt to boot. The target files for each stage are the following:

- Second stage: *winload.exe* or *winload.efi*, depends on the tested platform.
- Third stage: *ntoskrnl.exe*.
- Fourth stage: *win32k.sys*

The evaluator shall carry out the following steps for each stage in order to perform this test activity:

1. Boot the TOE and create a copy of the stage target file, e.g. *ntoskrnl.exe* if the third stage is being tested. This file will be modified in below steps.

2. Modify any bytes of the file using a hexadecimal editor, e.g. WinHex.

3. Boot the TOE into the Windows Pre Environment using the WinPE USB and replace the original file with the modified one.

4. Restart the TOE and observe the behavior. The system shall not boot properly and an error screen shall be shown.

5. Boot the TOE into the Windows Pre Environment using the WinPE USB drive, and restore the corrupt file using the backup created at step 1.

6. Restart the TOE and observe that the system boot properly.

33.3.2.3. Results

The evaluator has modified the files defined in the procedure section for each stage. The results obtained for the second stage and the third stage are the same. However, the results obtained for the fourth stage are different. These behaviors are describing as follows:

Second stage and third stage: Modifying winload.exe (or winload.efi) and ntoskrnl.exe
The evaluator has performed this test on the following evaluated platforms:

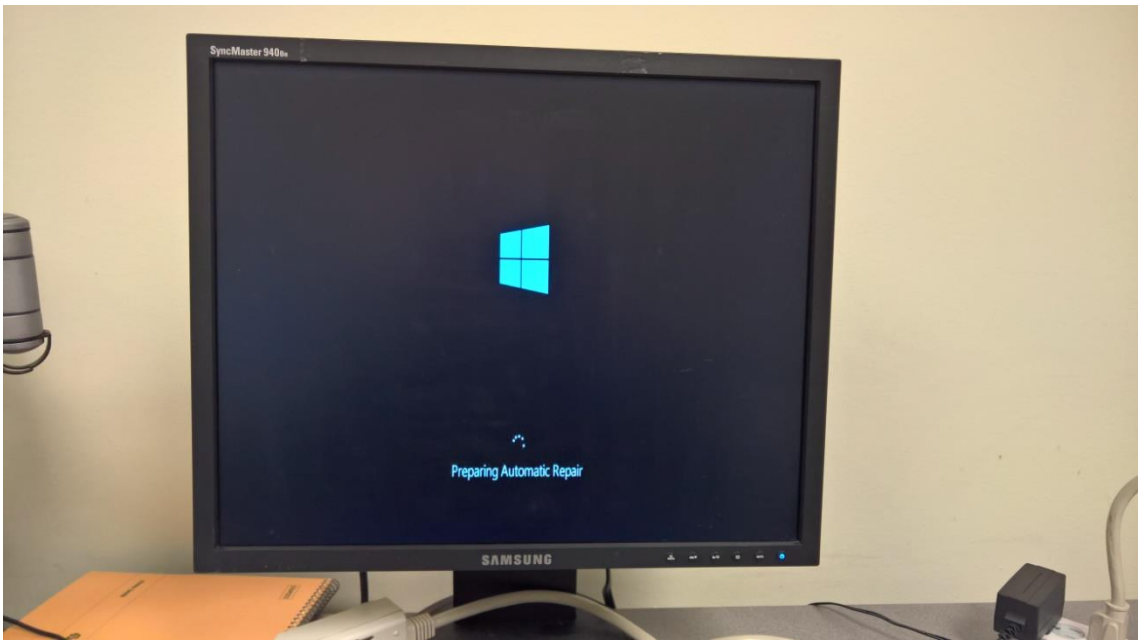
- Dell Optiplex 755 with Windows 10 x86 Pro Edition.
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition.
- Surface 3 with Windows 10 x64 Enterprise Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition.

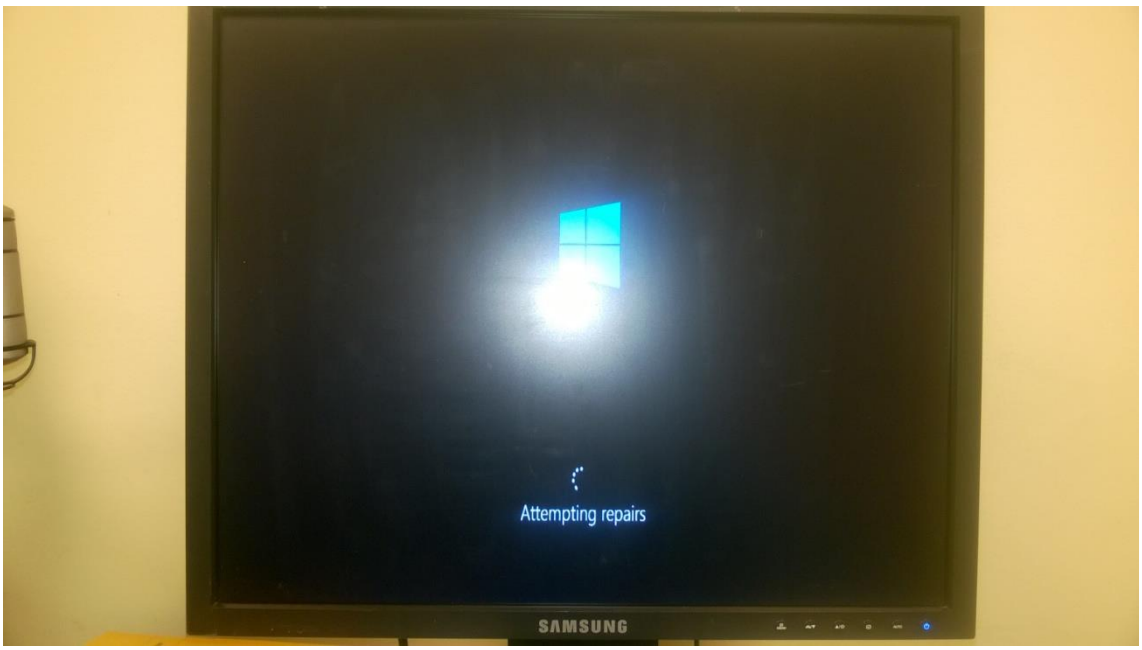
The evaluator has modified the determined file for each stage (winload.exe and ntoskrnl.exe) and has obtained the following behaviors depends on the operating system installed in the tested platform:

Testing platforms:
- Dell Optiplex 755 with Windows 10 x86 Pro Edition.
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- HP Pro x2 with Windows 10 x64 Pro Edition
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface 3 Pro with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition

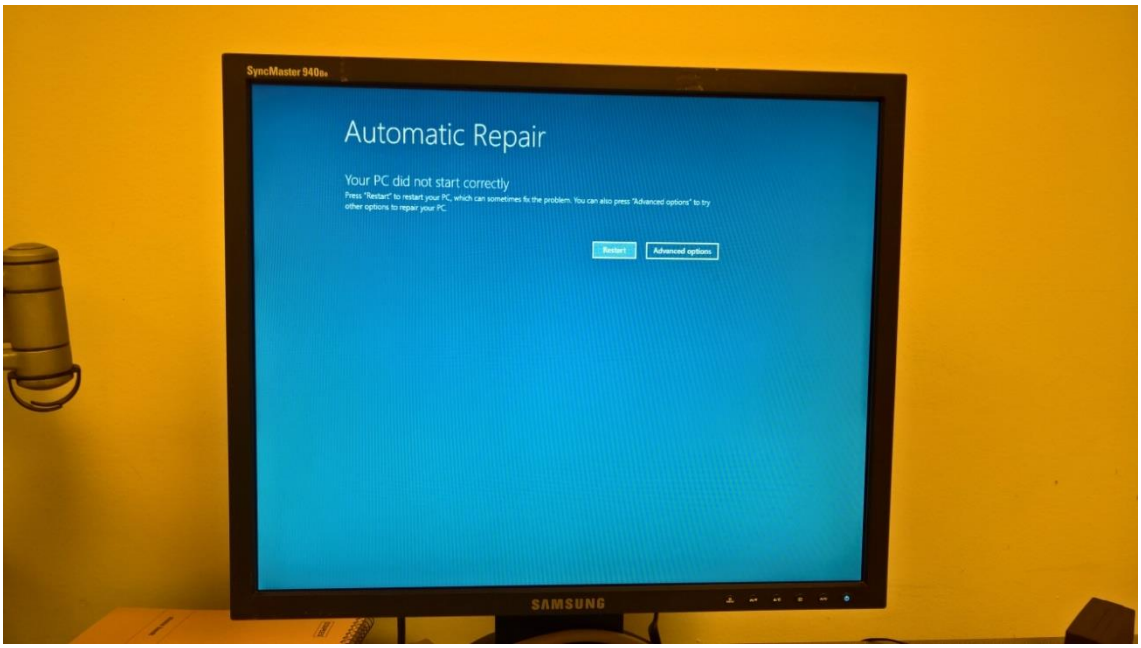
Obtained results:
The evaluator has obtained the same results for all the tested platforms defined above.

The following results were obtained during the testing at MS facilities. The operating system detected that there was a corrupt file during the boot process. After that, the operating system attempted to auto-repair itself, showing the following screens indicating that the computer is preparing to automatic repair.

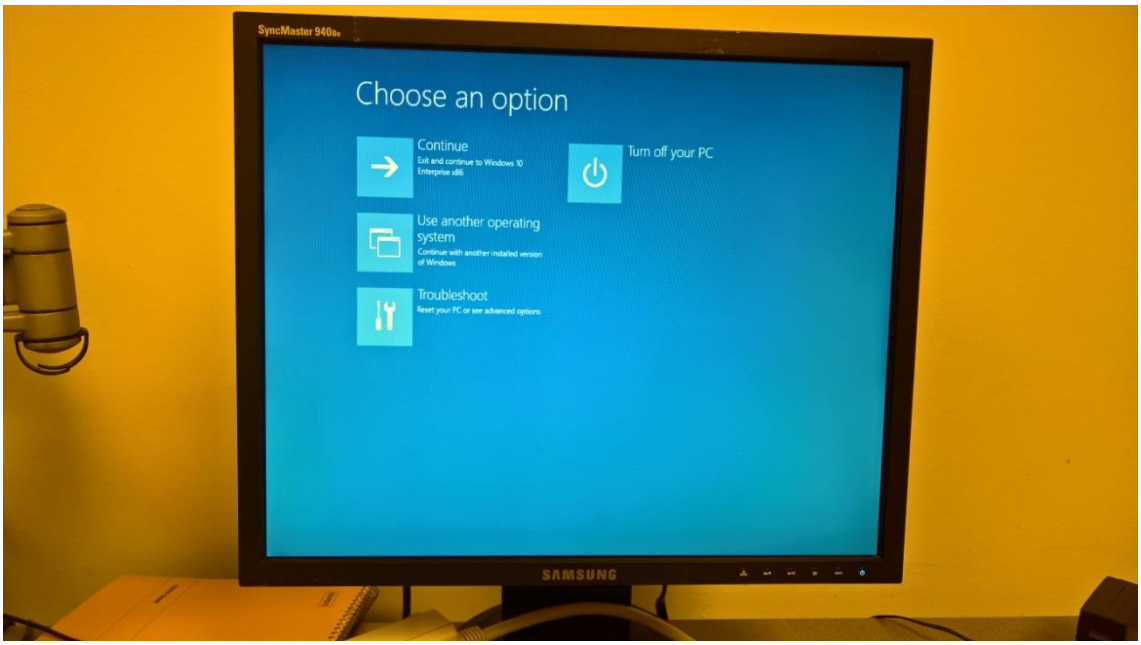




Finally, the operating system was not able to auto-repair itself and the following error screen was shown indicating that the operating system did not start properly.



This screen offers two options, restart the TOE or go to the Advanced Options, which allow turn off the PC, use another operating system or open the troubleshoot menu. The following screen show the options included in the Advanced Options screen:



Testing platform:

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition

Obtained result:

The following result was obtained during the testing at MS facilities. The operating system detected that there was a corrupt file during the boot process. After that, the operating

system showed a screen indicating that is preparing to perform an automatic repair. Finally the automatic repair failed and the Advanced Options screen was shown. The screens showed by the operating system during the boot process are similar to the ones included above.

Fourth stage: Modifying win32k.sys

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Pro Edition.
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.

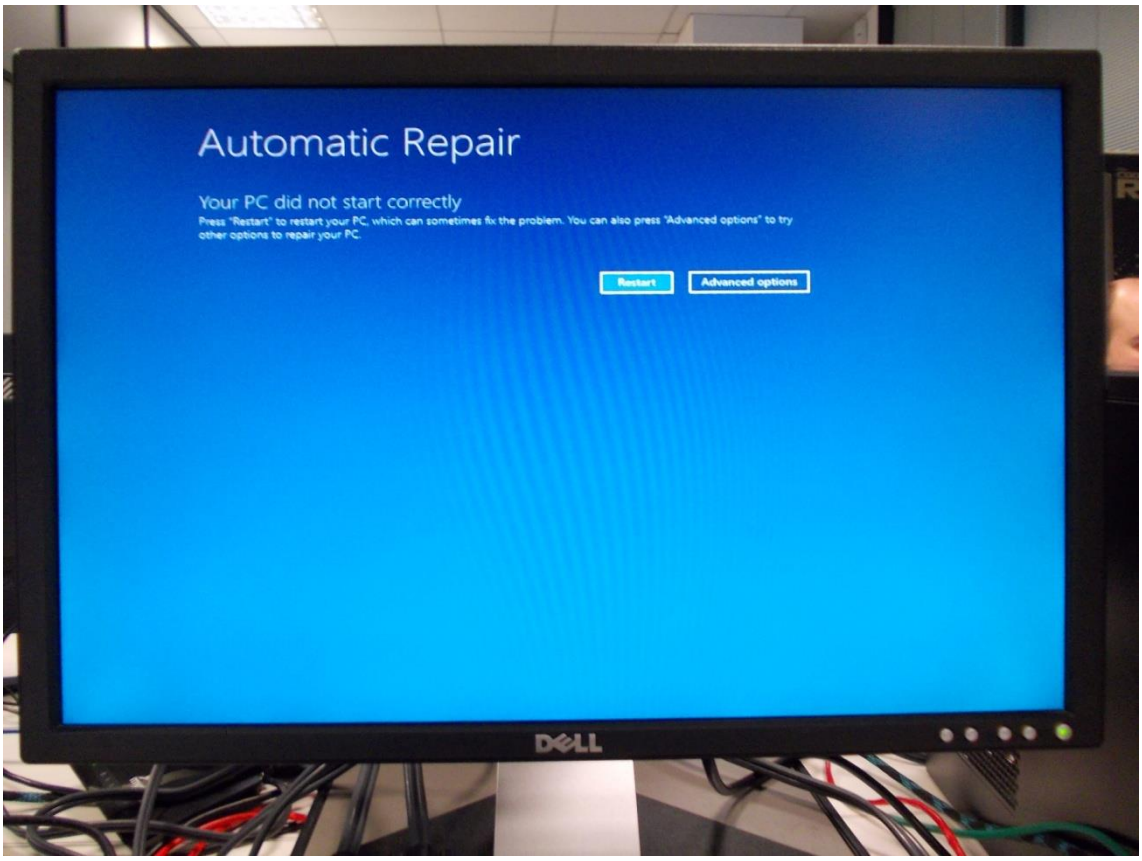
The evaluator has modified the determined file for this stage (win32k.sys) and has obtained the following behaviors depends on the tested platform:

Testing platform:

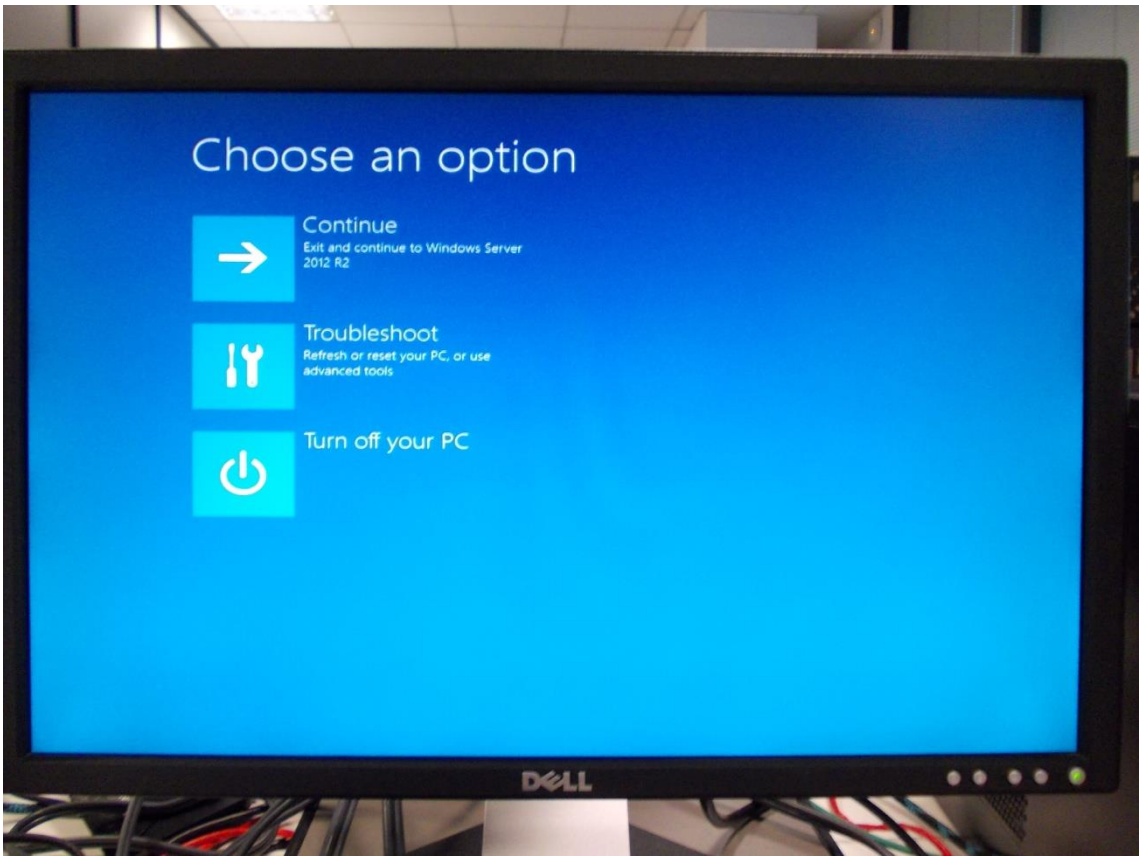
- Dell Optiplex 755 with Windows x86 Enterprise Edition
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- Surface Pro 3 with Windows 10 x64 Enterprise Edition

Obtained result:

For these platforms, the operating system detected that there was a corrupt file during the boot process and the following screen was showed indicating that the system did not boot properly.



This screen offers two options, restart the TOE or go to the Advanced Options, which allow turn off the PC, use another operating system or open the troubleshoot menu. In case of the Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition platform, the Advanced Option screen is showed instead of the above image:



Testing platform:

- Dell Optiplex 755 with Windows 10 x86 Pro Edition

Obtained result:

The evaluator has modified win32k.sys file and has attempted to boot the computer. The system has booted properly without detecting any integrity error. Additionally and after the boot process has been completed, the evaluator has analyzed the Windows Code Integrity/Operational log, and the following audit has been generated:

```
Message               : Code Integrity determined an unsigned kernel module
                        \Device\HarddiskVolume2\Windows\System32\win32k.sys is loaded into the system. Check with the
                        publisher to see if a signed version of the kernel module is available.
Id                    : 3001
Version               : 0
Qualifiers            :
Level                 : 3
Task                  : 1
Opcode                : 101
Keywords              : -9223372036854775808
RecordId              : 5
ProviderName          : Microsoft-Windows-CodeIntegrity
ProviderId            : 4ee76bd8-3cf4-44a0-a0ac-3937643e37a3
LogName               : Microsoft-Windows-CodeIntegrity/Operational
ProcessId             : 344
ThreadId              : 348
MachineName           : DESKTOP-3MUDNVP
UserId                : S-1-5-18
TimeCreated           : 12/15/2015 1:41:24 PM
ActivityId            :
RelatedActivityId     :
ContainerLog          : microsoft-windows-codeintegrity/operational
MatchedQueryIds       : {}
Bookmark              : System.Diagnostics.Eventing.Reader.EventBookmark
LevelDisplayName      : Warning
OpcodeDisplayName     :
TaskDisplayName       :
KeywordsDisplayNames  : {}
Properties            : {System.Diagnostics.Eventing.Reader.EventProperty,
                        System.Diagnostics.Eventing.Reader.EventProperty}
```

This audit log states that the operating system has detected that there was an unsigned kernel module, but in spite of this fact, the system has boot properly.

33.3.2.4. Verdict

As the obtained results demonstrate, the evaluator has found one case in which the TOE boot properly in spite of there is a modified file, and therefore there is an integrity error.

However, taking into account the information provided in the TSS section related to the loaded files during the bootchain, and how their integrity are checked depending on the platform architechture, the evaluator considers as follows:

During the test case execution, the evaluator performed a modification over the win32k.sys file in order to cause an integrity failure during the boot process. This file is not one of the critical kernel-mode drivers loaded during the bootchain. So, it can be assumed that the win32k.sys is loaded after the boochain is completed. On the other hand, this special behavior was identified in the following tested platform: Dell Optiplex 755 with Windows 10 x86 Pro Edition. Taking into account the information provided related to the integrity verification performed by Code Integrity in 32-bits platforms, win32k.sys is not one of the files for which the digital signature is mandatory. Due to this, the operating system booted properly in spite of the digital signature for win32k.sys was no longer valid (this fact was registered by the TOE, including a new entry in the Code Integrity audit log as it is stated in the Result section).

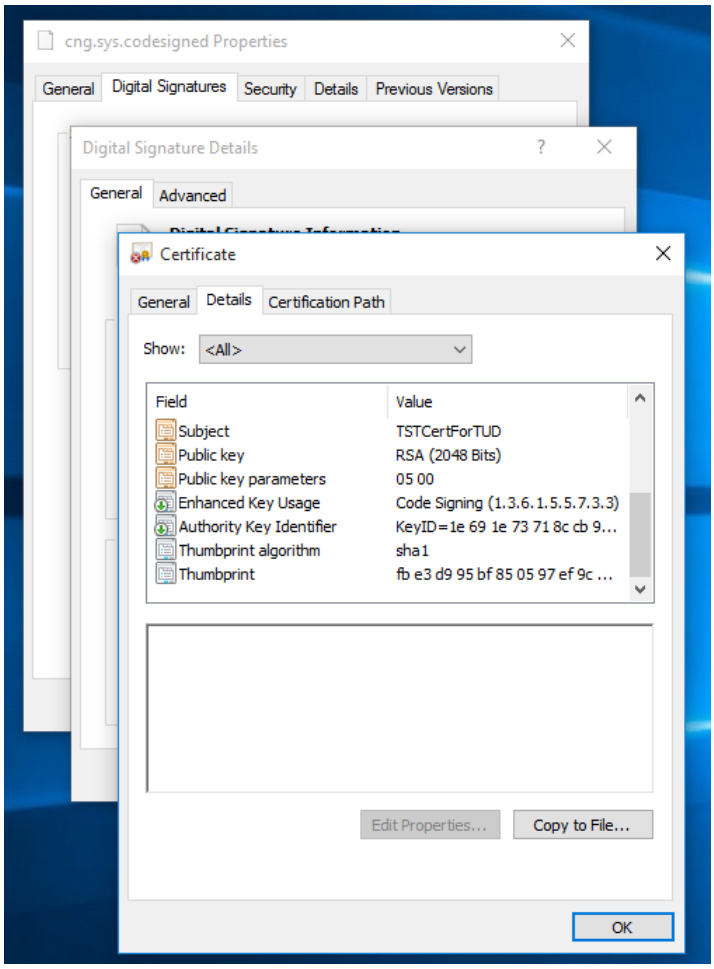
Concluding, the evaluator considers that, the tests results obtained during this test activity demonstrate that **Test 2** requirements established in the assurance activity section are not fulfilled. So, the **PASS** verdict is assigned to **Test 2**.

33.3.3. Test 3

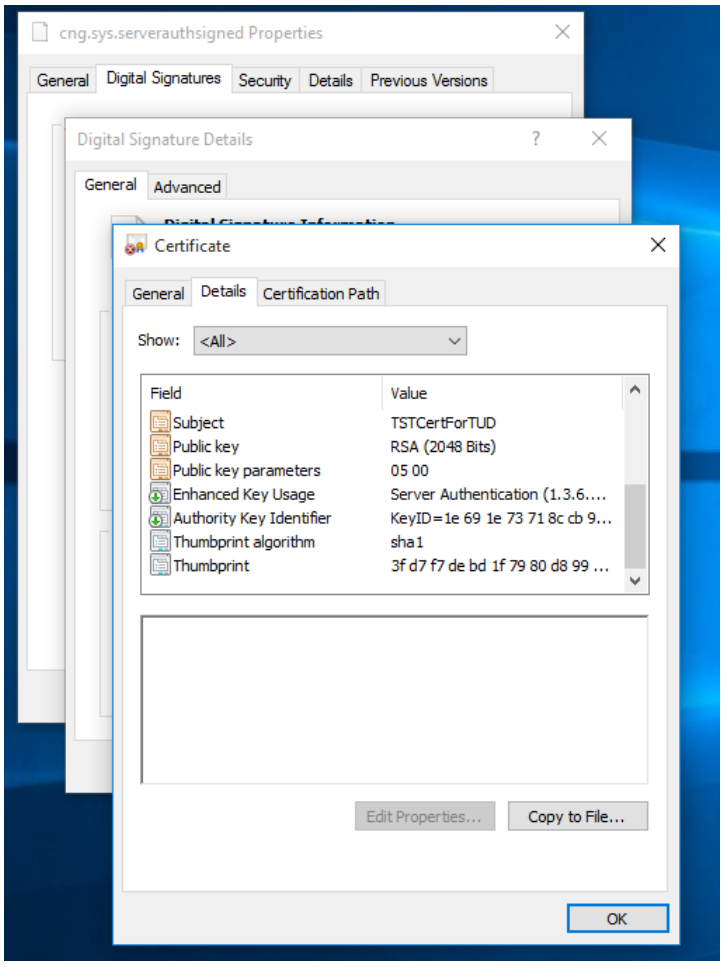
33.3.3.1. Setup

In order 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 is signed using a certificate with *Code Signing* purpose in the *extendedKeyUsage* field. The following picture shows this fact:



- cng.sys.serverauthsigned: This file is signed using a certificate with *Server Authentication* purpose in the *extendedKeyUsage* field. The following picture shows this fact:



Additionally, a WinPE USB for both architectures (x64 and x86) must be available.

33.3.3.2. 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.

1. Run a cmd terminal as administrator and type the following command in order to enable the test signing mode: *bcdedit /set testsigning on*

   Enabling this mode allow the TOE load binaries which have not been signed by a trusted certification authority during the boot process.

2. After that, create a copy of the valid binary. This copy will be restored in next steps.

3. Boot the TOE into the WinPE and replace the valid binary with the signed one with the *Server Authentication* purpose in the *extendedKeyUsage* field. Restart the TOE and observe that an integrity violation is triggered and the system does not boot properly.

4. Boot the TOE again into the WinPE, and replace the bad binary with the signed one with the *Code Signing* purpose in the *extendedKeyUsage* field. Restart the TOE and observe that the boot process is completed successfully.

5. Finally in order to restore the previous state of the tested platform, boot the TOE into the WinPE and replace the binary with the backup created at step 2. After that, restart the TOE, open a cmd terminal as administrator and type the following command in order to disable the test signing mode: *bcedit /set testsigning off*

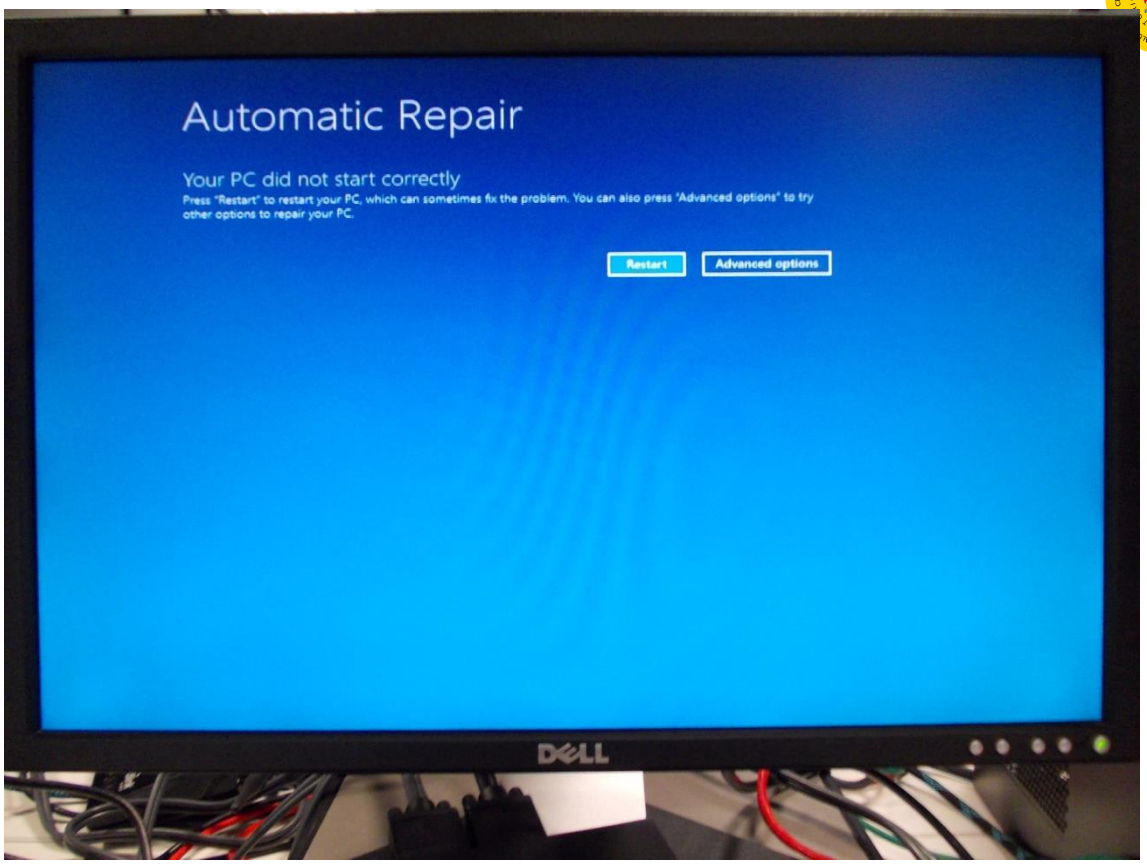
33.3.3.3. Results

The evaluator has performed this test on the following evaluated platforms:

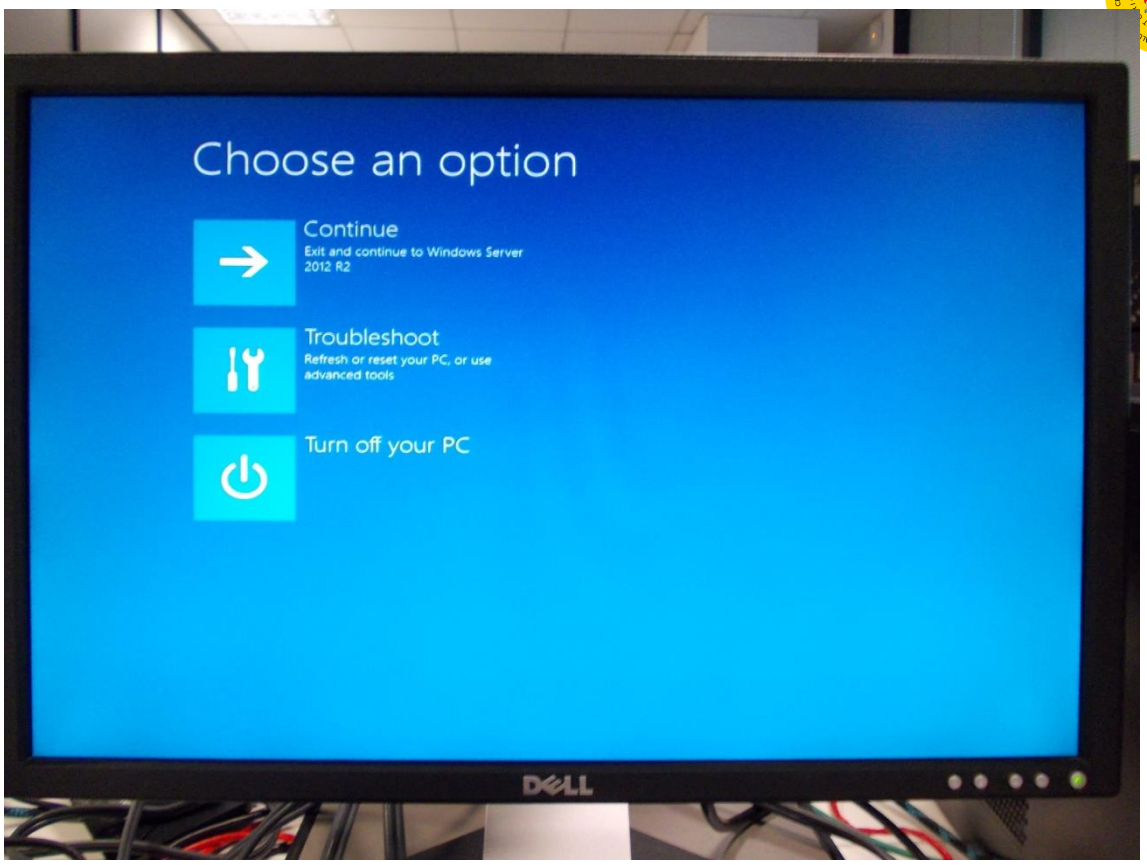
- Dell Optiplex 755 with Windows 10 x86 Pro Edition
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- HP Pro x2 612 with Windows 10 x64 Pro Edition

The evaluator has replaced a binary file loaded during the boot process with one signed using a certificate with the *Server Authentication* value in the *extendedKeyUsage* field. In these cases, the TOE did not boot properly, because an integrity error was triggered. The evaluator obtained the following behaviors depends on the operating system in the tested platforms:

- Windows 10: 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 has been shown:



- Windows Server 2012 R2: The operating system has detected the integrity error and the following screen has been shown, which include different options.



On the other hand, the boot process has finished successfully after replacing the bad binary with the one signed by a certificate with the *Code Signing* value in the *extendedKeyUsage* field. The evaluator has obtained this behavior for all the tested platforms.

33.3.3.4. Verdict

As the above results state, an integrity error is triggered if a binary file which has been signed using a certificate with a *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 tests results obtained during this test activity demonstrate the fulfillment of **Test 3** requirement established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 3**.

33.4. Final Verdict

Due to all test activities 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 FPT_TST_EXT.1.1.

34. FPT_TUD_EXT.1.1

34.1. Assurance activity

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.)

34.2. Documentation review activity

34.2.1. Findings

The evaluator has reviewed the operational guidance, which includes the following information in its section **12. Managing Updates.**

The following steps shall be performed in order to check for updates for Windows 10:

- Open **Settings**
- Click **Update & Security**
- Under Windows Update, click **Check for updates**

The following steps shall be performed in order to check for updates for Windows Server 2012 R2:

- Open **Control Panel**
- Click **System and Security**
- Under **Windows Update**, click **Check for updates**

As the image above states, the operational guidance provides the steps which the evaluator should follow in order to check for updates.

34.2.2. Verdict

The operational guidance includes in its section **12.Managing Updates** enough information to allow the evaluator determines how to check for new updates. Additionally, this information is provided for both operating systems, Windows 10 and Windows Server 2012 R2.

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.

34.3. Test Activity

34.3.1. Test - Checking update over a trusted channel

34.3.1.1. Setup

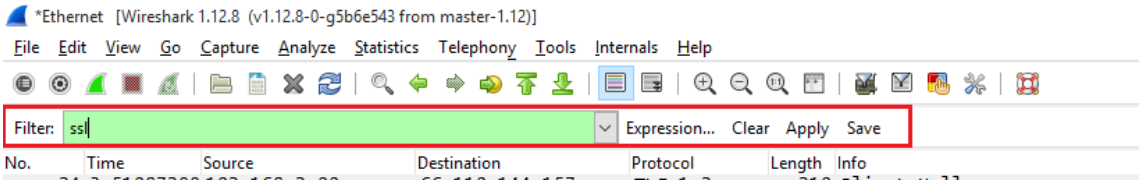
The following tool must be installed in the platform in order to allow the evaluator perform this test:

- A network protocol analyzer, e.g. Wireshark.

34.3.1.2. Procedure

The following steps must be performed in order to ensure that the update checking process is performed over a trusted channel as described in FTP_ITC_EXT.1.

1. Open Wireshark and configure it to listen through the active Ethernet interface.
2. The following step depends on the operating system which is being tested:
    a. For Windows 10: Go to *Settings->Update & Security ->Windows Update*. Then click in *Check for updates.*
    b. For Windows Server 2012 R2: Go to *Control Panel -> System and Security -> Windows Update.* Then click in *Check for updates.*
3. 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. In order to make easier the traffic analysis, the evaluator can apply a filter over the capture, 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.



34.3.1.3. Results

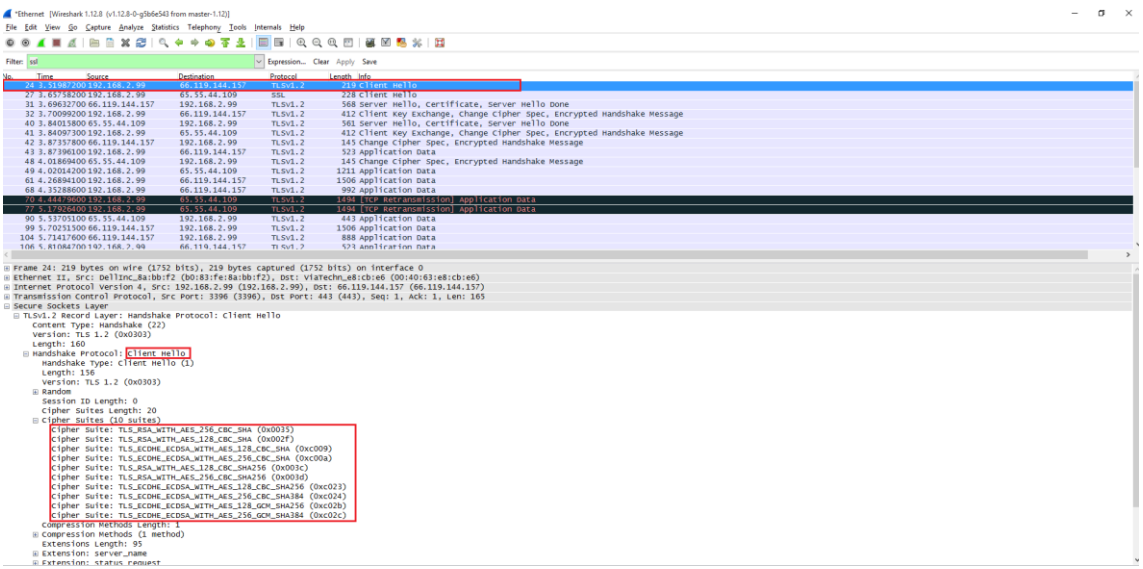
The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition.

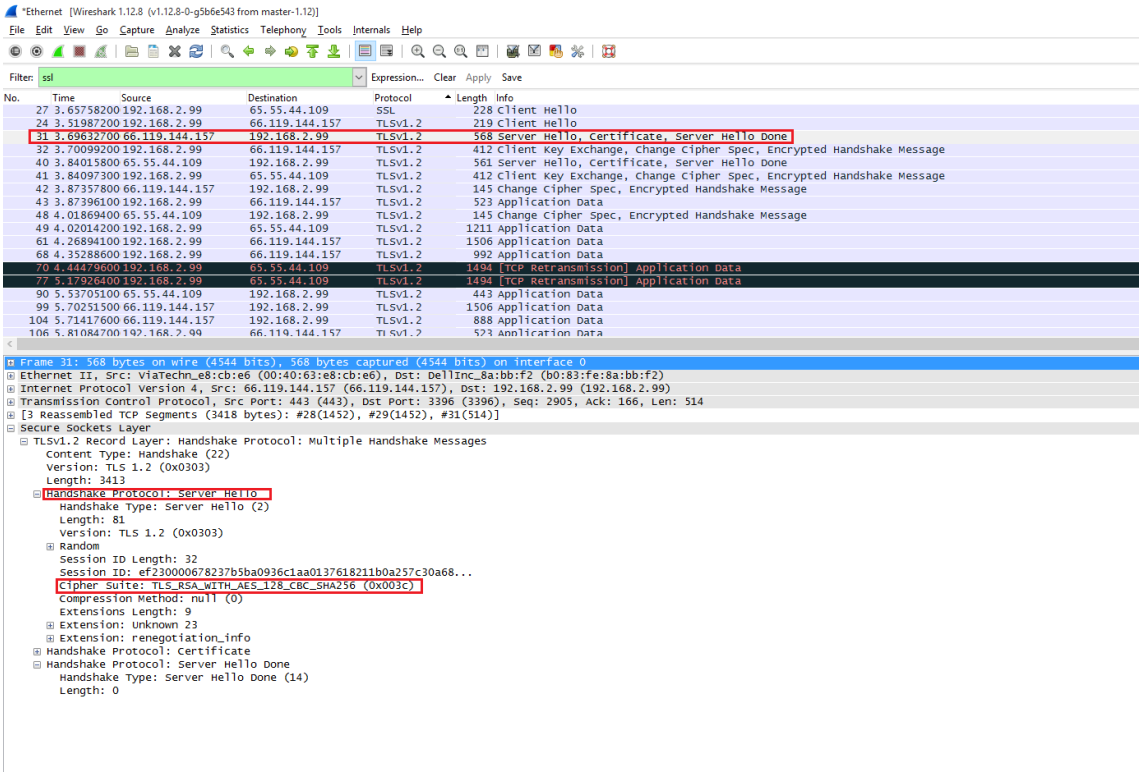
The evaluator has initiated the update checking process and has checked that the connection between the TOE and the update server is established over a secure channel. The obtained results have been the same in all tested platforms.

The following screenshots, which are taken from a Wireshark capture in one of the tested platform, show this fact.

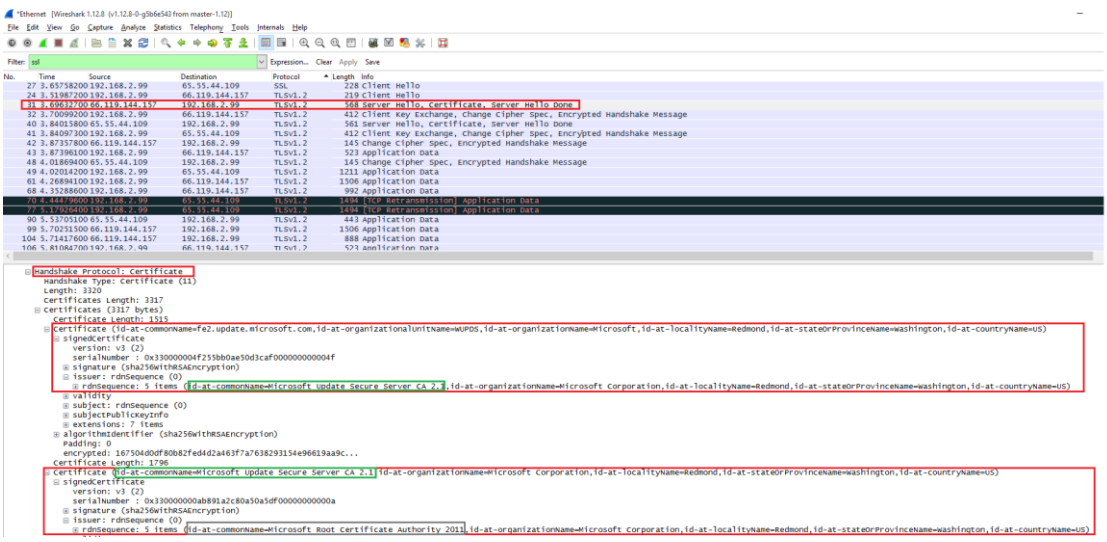
First of all, the evaluator has identified the cipher suites in the *Client Hello* packet. These cipher suites match with the cipher suites defined in FCS_TLSC_EXT.1.

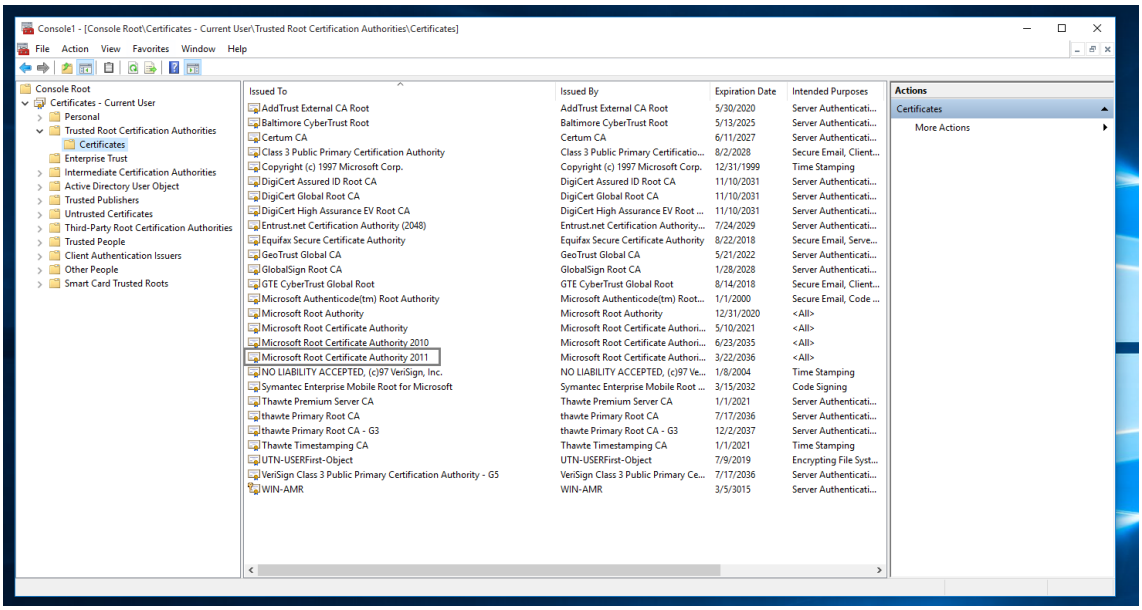


After that, the server selects one of the cipher suites offered by the client machine. It could be observed in the *Server Hello* packet.



Additionally, the server sends the certificate and its certificate chain. The root CA of the certificate chain sent by the server match with one of the root certificate stored in the client machine.





34.3.1.4. Verdict

As the above results state, 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

The evaluator considers that, the test results obtained during this test activity demonstrate the fulfillment of the test requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to this test.

34.4. Final Verdict

Due to both the documentation review activity and the test activity 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 FPT_TUD_EXT.1.1.

35. FPT_TUD_EXT.1.2

35.1. Assurance activity

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.

35.2. Documentation review activity

35.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.6.5 Windows and Application Updates,** which is related to how the updates are provided by the vendor.

This section states that the updates to Windows are delivered through the Windows Update capability, which is enabled by default. Additionally, this section also states that the user can obtain update files visiting the following vendor website:

- http://catalog.update.microsoft.com

35.2.2. Verdict

The TSS includes in its section **6.6.5 Windows and Application Updates** enough information to allow the evaluator determines how the updates are provided by the vendor and how can obtain a update files from the vendor website.

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.

35.3. Test Activity

35.3.1. Test 1

35.3.1.1. Setup

The following tools must be installed in the testing platform in order to allow performing this test:

- 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. WinHex.
- WinCab, a tool provided by the vendor to pack and unpack MSU files.

35.3.1.2. Procedure

Obtain a update file

The following steps must be performed to download an update file:

1. Open Internet Explorer and browse to *http://catalog.update.microsoft.com*
2. In the search box type "Windows 10" or "Windows Server 2012 R2" depending on the operating system of the platform which is being tested. A list of available update will be shown.
3. 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, add the update to the basket.
4. Finally, click in *View Basket* and after that, click in *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).

Invalidating digital signature

In order to invalidate the digital signature two approaches are followed.

A. **Approach 1 - Digital signature modification**:

    1. Create a copy of the update file, which is going to be modified.
    2. Right-click over it and select the Properties option.

3. Go to Digital Signatures tab and observer that the update file contains two valid digital signatures (SHA1 and SHA256).

4. Open a cmd terminal and go to the folder where SignTool is installed (typical path is ***%programfiles(x86)%\WindowsKits\10\bin\x86***).

5. Obtain the digital signatures typing the following command: "*signtool.exe verify /all /pa /v updatefile.msu*". The signatures will be shown in *Hash of file (sha1)* and *Hash of file (sha256)* fields.

6. On the other hand, open the update file in WinHex editor and search the value obtained in the previous step.

7. Modify one byte of the digital signature and save the file.

8. Repeat the step 6, and modify the other digital signature.

9. To ensure that the digital signatures have been modified correctly, the step 5 could be repeated. At this moment the digital signatures are no longer valid. Other way to check whether the digital signatures have been modified is using the following PowerShell command: *Get-AuthenticodeSignature updatefile.msu.*

10. Finally, attempt to install the modified update file and observe that the operating system rejects the operation.

B. Approach 2 - Testing using vendor tool:

The vendor has provided WinCab tool, which allow the evaluator pack and unpack the MSU file and extract its content.

To extract the content of an 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 package 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>.

Using this tool, the following test must be performed by the evaluator

1. Test 1: Unpack the MSU file, modify the content of the *pkgProperties* file, pack the files and attempt to install the update.

2. Test 2: Unpack the MSU file, modify the content of the xml file, pack the files and attempt to install the update.

3. Test 3: Unpack the MSU file, delete the *pkgProperties* file, pack the files and attempt to install the update.

4. Test 4: Unpack the MSU file, delete the xml file, pack the files and attempt to install the update.

5. Test 5: Unpack the MSU file, modify the content of the *Windows10.0-KB9999991-x64.cab* file, pack the files and attempt to install the update.

6. Test 6: Unpack the MSU file, modify the content of the *WSUSSCAN.cab* file, pack the files and attempt to install the update.

35.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has attempted to install a modified update file and has obtained different results depending on the used approach and the modified bytes offset. These results have been the same in all tested platforms.

The results obtained during the test execution are described as follows:

A. Approach 1 – Digital signature modification:

The evaluator has executed the *signtool* command and has obtained the SHA-1 and SHA-256 hash from the downloaded file:

```
Verifying: C:\Users\Evaluador\Desktop\Windows10.0-KB9999991-x64-realsigned.msu
Signature Index: 0 (Primary Signature)
Hash of file (sha1): E3133419C1177F5EB23AC6273649FA6E236440F8

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority
    Issued by: Microsoft Root Certificate Authority
    Expires:   Mon May 10 00:28:13 2021
    SHA1 hash: CDD4EEAE6000AC7F40C3802C171E30148030C072

        Issued to: Microsoft Code Signing PCA
        Issued by: Microsoft Root Certificate Authority
        Expires:   Mon Aug 31 23:29:32 2020
        SHA1 hash: 3CAF9BA2DB5570CAF76942FF99101B993888E257

            Issued to: Microsoft Corporation
            Issued by: Microsoft Code Signing PCA
            Expires:   Sun Sep 04 18:42:45 2016
            SHA1 hash: 3BDA323E552DB1FDE5F4FBEE75D6D5B2B187EEDC

The signature is timestamped: Tue Jul 28 22:18:11 2015
Timestamp Verified by:
    Issued to: Microsoft Root Certificate Authority
    Issued by: Microsoft Root Certificate Authority
    Expires:   Mon May 10 00:28:13 2021
    SHA1 hash: CDD4EEAE6000AC7F40C3802C171E30148030C072

        Issued to: Microsoft Time-Stamp PCA
        Issued by: Microsoft Root Certificate Authority
        Expires:   Sat Apr 03 14:03:09 2021
        SHA1 hash: 375FCB825C3DC3752A02E34EB70993B4997191EF

            Issued to: Microsoft Time-Stamp Service
            Issued by: Microsoft Time-Stamp PCA
            Expires:   Mon Jun 20 18:32:03 2016
            SHA1 hash: 0731E6E5631C8EC056E121B4DF6832DD460D90EF

Signature Index: 1
Hash of file (sha256): 1ADBD9CD5BEAF90923DDDD6D482B37DFD339FB62DD50A6AFD2FFFDAE66CF0EE0
```

These hashes are in the offset 0x0005D390 (SHA-1) and 0x0005EEA0 (SHA-256). The original hashes are as follows:

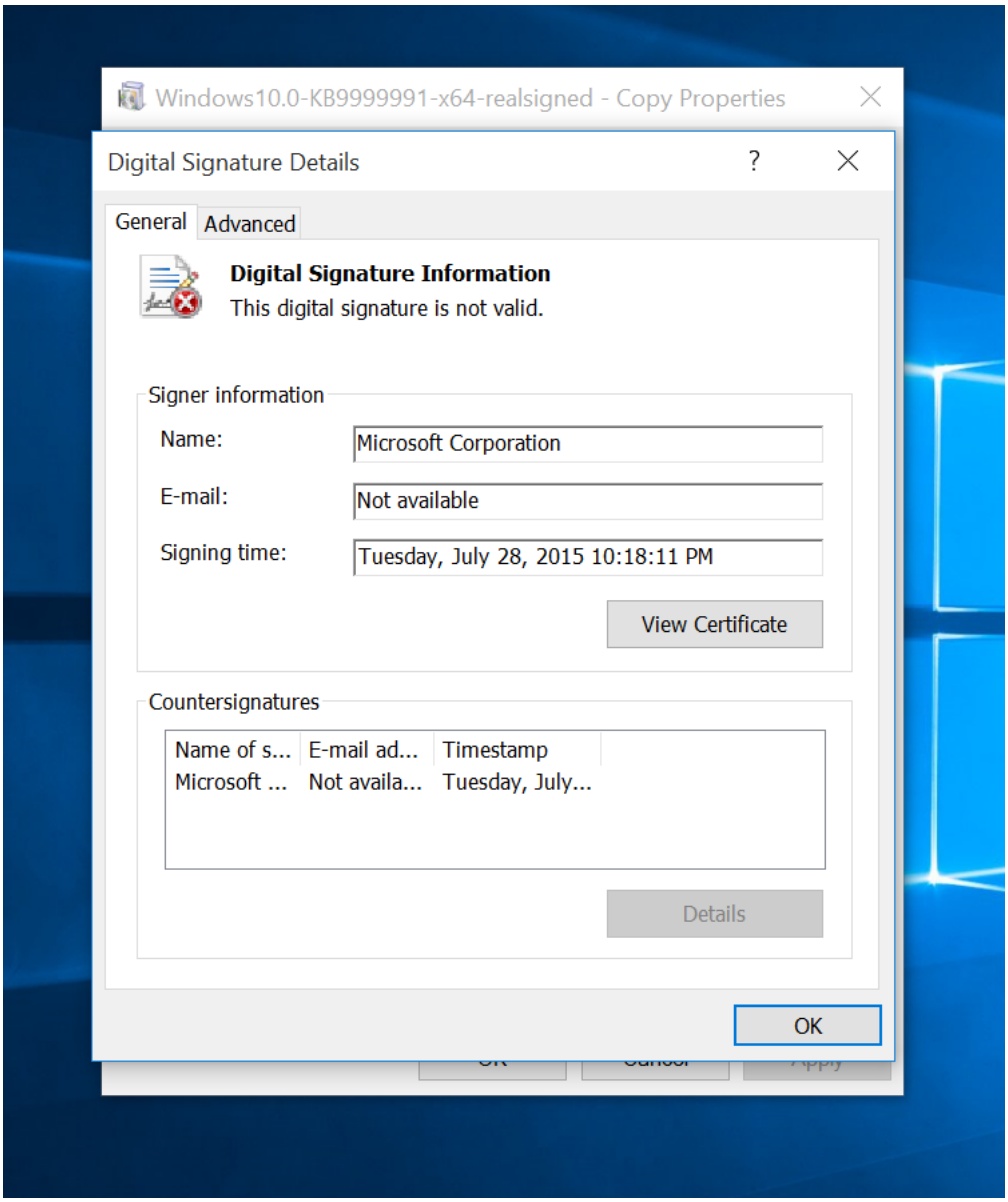
```
0005D330  02 01 04 A0 71 30 6F 30  4A 06 0A 2B 06 01 04 01  ... q0o0J..+....
0005D340  82 37 02 01 1C A1 3C 04  10 A6 B5 86 D5 B4 A1 24  ▌7...¡<..¦µ▌Õ´¡$
0005D350  66 AE 05 A2 17 DA 8E 60  D6 04 28 31 26 30 24 06  f®.¢.Ú▌`Ö.(1&0$.
0005D360  0A 2B 06 01 04 01 82 37  02 05 01 31 16 04 14 E5  .+....▌7...1...å
0005D370  78 E9 E6 C7 5B 33 B5 39  A3 71 E0 1B 5F EA ED A1  xéæÇ[3µ9£qà._êí¡
0005D380  AA 3F 37 30 21 30 09 06  05 2B 0E 03 02 1A 05 00  ª?70!0...+......
0005D390  04 14 E3 13 34 19 C1 17  7F 5E B2 3A C6 27 36 49  ..ã.4.Á..^²:Æ'6I
0005D3A0  FA 6E 23 64 40 F8 A0 82  15 82 30 82 04 C3 30 82  ún#d@ø ▌.▌0▌.Ã0▌
0005D3B0  03 AB A0 03 02 01 02 02  13 33 00 00 00 71 B3 2E  .« ......3...q³.
0005D3C0  8A 6B 82 AA 1F 4E 00 00  00 00 00 71 30 0D 06 09  ▌k▌ª.N.....q0...
0005D3D0  2A 86 48 86 F7 0D 01 01  05 05 00 30 77 31 0B 30  *▌H▌÷......0w1.0
0005D3E0  09 06 03 55 04 06 13 02  55 53 31 13 30 11 06 03  ...U....US1.0...
0005D3F0  55 04 08 13 0A 57 61 73  68 69 6E 67 74 6F 6E 31  U....Washington1
0005D400  10 30 0E 06 03 55 04 07  13 07 52 65 64 6D 6F 6E  .0...U....Redmon
0005D410  64 31 1E 30 1C 06 03 55  04 0A 13 15 4D 69 63 72  d1.0...U....Micr
0005D420  6F 73 6F 66 74 20 43 6F  72 70 6F 72 61 74 69 6F  osoft Corporatio

0005EE50  48 04 10 A6 B5 86 D5 B4  A1 24 66 AE 05 A2 17 DA  H..¦µ▌Õ´¡$f®.¢.Ú
0005EE60  8E 60 D6 04 34 31 32 30  30 06 0A 2B 06 01 04 01  ▌`Ö.41200..+....
0005EE70  82 37 02 05 01 31 22 04  20 5C BD 2F E3 4E 22 1D  ▌7...1". \½/ãN".
0005EE80  8B 58 77 01 75 2D 31 F6  4A 08 69 0C 4D 77 70 F7  ▌Xw.u-1öJ.i.Mwp÷
0005EE90  06 F4 9D 42 44 31 6F AE  20 30 31 30 0D 06 09 60  .ô.BD1o® 010...`
0005EEA0  86 48 01 65 03 04 02 01  05 00 04 20 1A DB D9 CD  ▌H.e..... .ÛÙÍ
0005EEB0  5B EA F9 09 23 DD DD 6D  48 2B 37 DF D3 39 FB 62  [êù.#ÝÝmH+7ßÓ9ûb
0005EEC0  DD 50 A6 AF D2 FF FD AE  66 CF 0E E0 A0 82 0B 83  ÝP¦¯Òÿý®fÏ.à ▌.▌
0005EED0  30 82 05 0B 30 82 03 F3  A0 03 02 01 02 02 13 33  0▌..0▌.ó ......3
0005EEE0  00 00 00 7B A2 81 0B 87  11 AB E7 FC 00 00 00 00  ...{¢..▌.«çü....
0005EEF0  00 7B 30 0D 06 09 2A 86  48 86 F7 0D 01 01 0B 05  .{0...*▌H▌÷.....
0005EF00  00 30 7E 31 0B 30 09 06  03 55 04 06 13 02 55 53  .0~1.0...U....US
0005EF10  31 13 30 11 06 03 55 04  08 13 0A 57 61 73 68 69  1.0...U....Washi
0005EF20  6E 67 74 6F 6E 31 10 30  0E 06 03 55 04 07 13 07  ngton1.0...U....
0005EF30  52 65 64 6D 6F 6E 64 31  1E 30 1C 06 03 55 04 0A  Redmond1.0...U..
0005EF40  13 15 4D 69 63 72 6F 73  6F 66 74 20 43 6F 72 70  ..Microsoft Corp
0005EF50  6F 72 61 74 69 6F 6E 31  28 30 26 06 03 55 04 03  oration1(0&..U..
0005EF60  13 1F 4D 69 63 72 6F 73  6F 66 74 20 43 6F 64 65  ..Microsoft Code
```

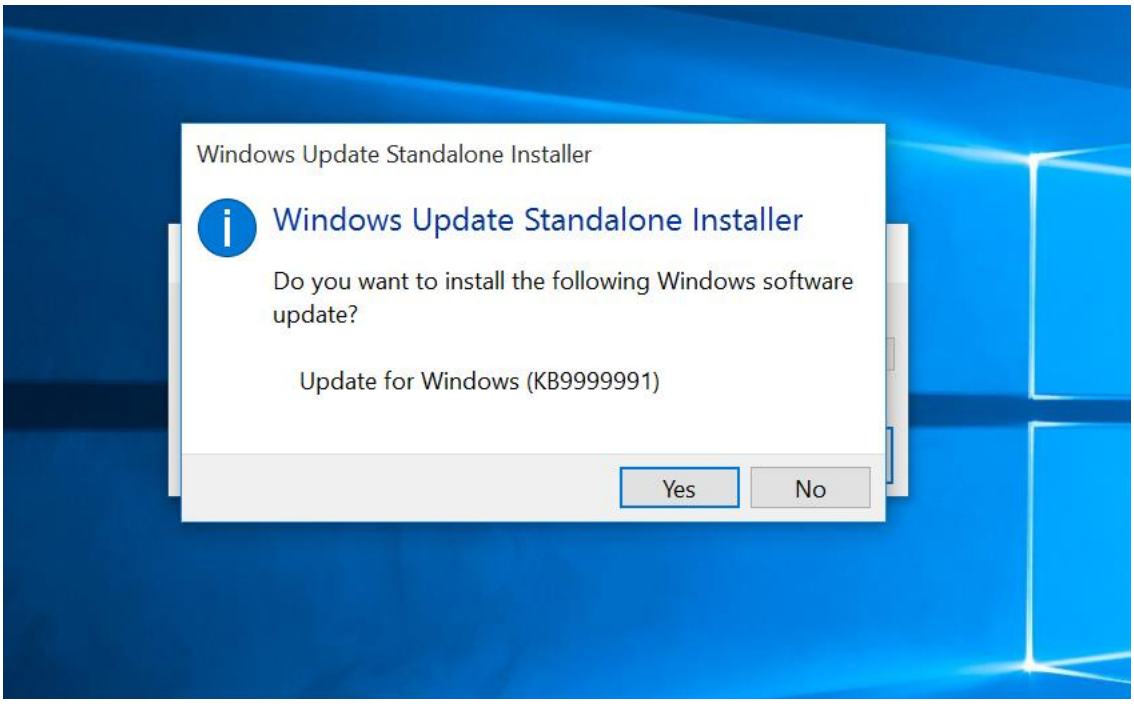
Once the hashes have been modified:

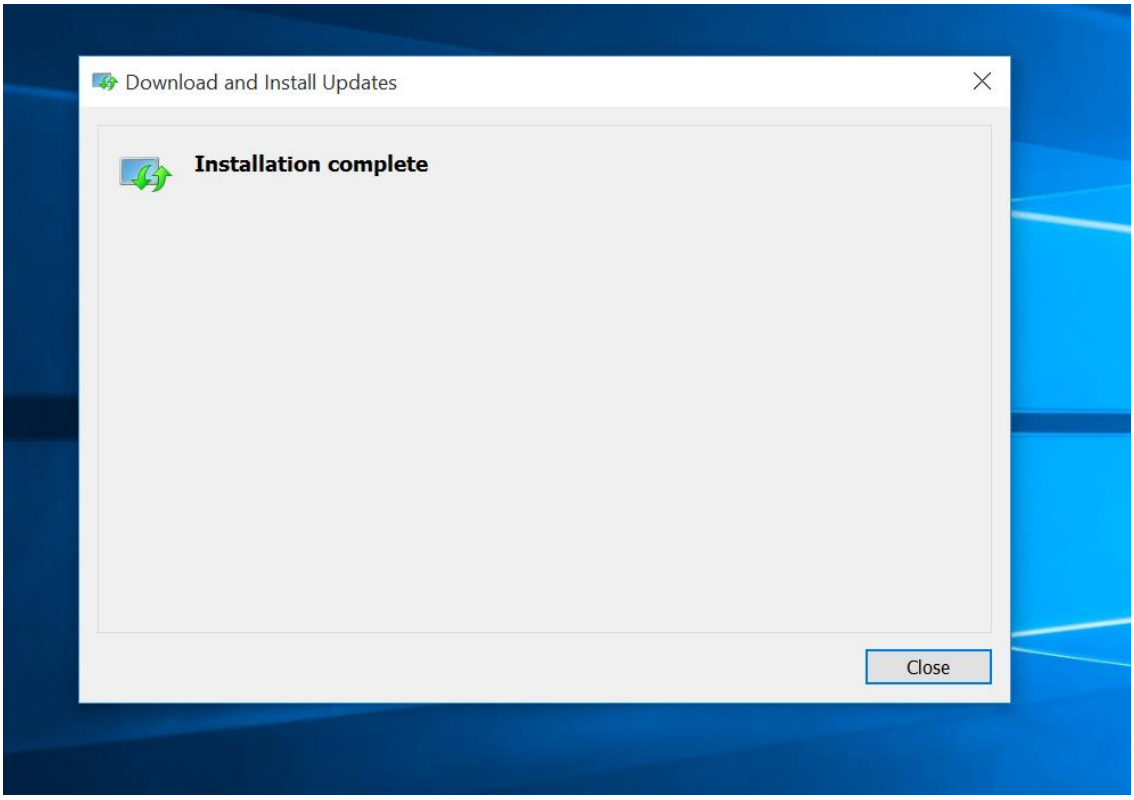
```
0005D330  02 01 04 A0 71 30 6F 30  4A 06 0A 2B 06 01 04 01  ... q0o0J..+....
0005D340  82 37 02 01 1C A1 3C 04  10 A6 B5 86 D5 B4 A1 24  ▌7...¡<..¦µ▌Õ´¡$
0005D350  66 AE 05 A2 17 DA 8E 60  D6 04 28 31 26 30 24 06  f®.¢.Ú▌`Ö.(1&0$.
0005D360  0A 2B 06 01 04 01 82 37  02 05 01 31 16 04 14 E5  .+....▌7...1...å
0005D370  78 E9 E6 C7 5B 33 B5 39  A3 71 E0 1B 5F EA ED A1  xéæÇ[3µ9£qà._êí¡
0005D380  AA 3F 37 30 21 30 09 06  05 2B 0E 03 02 1A 05 00  ª?70!0...+......
0005D390  04 14 AA AA AA AA 01 17  7F 5E B2 3A C6 27 36 49  ..ªªªªÁ..^²:Æ'6I
0005D3A0  FA 6E 23 64 40 F8 A0 82  15 82 30 82 04 C3 30 82  ún#d@ø ▌.▌0▌.Ã0▌
0005D3B0  03 AB A0 03 02 01 02 02  13 33 00 00 00 71 B3 2E  .« ......3...q³.
0005D3C0  8A 6B 82 AA 1F 4E 00 00  00 00 00 71 30 0D 06 09  ▌k▌ª.N.....q0...
0005D3D0  2A 86 48 86 F7 0D 01 01  05 05 00 30 77 31 0B 30  *▌H▌÷......0w1.0
0005D3E0  09 06 03 55 04 06 13 02  55 53 31 13 30 11 06 03  ...U....US1.0...
0005D3F0  55 04 08 13 0A 57 61 73  68 69 6E 67 74 6F 6E 31  U....Washington1
0005D400  10 30 0E 06 03 55 04 07  13 07 52 65 64 6D 6F 6E  .0...U....Redmon
0005D410  64 31 1E 30 1C 06 03 55  04 0A 13 15 4D 69 63 72  d1.0...U....Micr
0005D420  6F 73 6F 66 74 20 43 6F  72 70 6F 72 61 74 69 6F  osoft Corporatio
```

```
0005EE50  48 04 10 A6 B5 86 D5 B4  A1 24 66 AE 05 A2 17 DA  H..¦µ▮Õ´¡$f®.¢.Ú
0005EE60  8E 60 D6 04 34 31 32 30  30 06 0A 2B 06 01 04 01  ▮`Ö.41200..+....
0005EE70  82 37 02 05 01 31 22 04  20 5C BD 2F E3 4E 22 1D  ▮7...1". \½/ãN".
0005EE80  8B 58 77 01 75 2D 31 F6  4A 08 69 0C 4D 77 70 F7  ▮Xw.u-1öJ.i.Mwp÷
0005EE90  06 F4 9D 42 44 31 6F AE  20 30 31 30 0D 06 09 60  .ô.BD1o® 010...`
0005EEA0  86 48 01 65 03 04 02 01  05 00 04 20 AA AA AA AA  ▮H.e..... ªªªª
0005EEB0  5B EA F9 09 23 DD DD 6D  48 2B 37 DF D3 39 FB 62  [êù.#ÝÝmH+7ßÓ9ûb
0005EEC0  DD 50 A6 AF D2 FF FD AE  66 CF 0E E0 A0 82 0B 83  ÝP¦¯Òÿý®fÏ.à ▮.▮
0005EED0  30 82 05 0B 30 82 03 F3  A0 03 02 01 02 02 13 33  0▮..0▮.ó ......3
0005EEE0  00 00 00 7B A2 81 0B 87  11 AB E7 FC 00 00 00 00  ...{¢..▮.«çü....
0005EEF0  00 7B 30 0D 06 09 2A 86  48 86 F7 0D 01 01 0B 05  .{0...*▮H▮÷.....
0005EF00  00 30 7E 31 0B 30 09 06  03 55 04 06 13 02 55 53  .0~1.0...U....US
0005EF10  31 13 30 11 06 03 55 04  08 13 0A 57 61 73 68 69  1.0...U....Washi
0005EF20  6E 67 74 6F 6E 31 10 30  0E 06 03 55 04 07 13 07  ngton1.0...U....
0005EF30  52 65 64 6D 6F 6E 64 31  1E 30 1C 06 03 55 04 0A  Redmond1.0...U..
0005EF40  13 15 4D 69 63 72 6F 73  6F 66 74 20 43 6F 72 70  ..Microsoft Corp
0005EF50  6F 72 61 74 69 6F 6E 31  28 30 26 06 03 55 04 03  oration1(0&..U..
0005EF60  13 1F 4D 69 63 72 6F 73  6F 66 74 20 43 6F 64 65  ..Microsoft Code
```

The evaluator checks the validity of the digital signature and attempts to install the update obtaining the following result:







Therefore, after modifying the hashes values and even considering that the digital signature is no longer valid, the update is installed correctly.

B. Approach 2 – Testing using vendor tool:

The evaluator has performed the test defined for this approach and has obtained the following results:

- Test 1: Unpack the MSU file, modify the content of the *pkgProperties* file, pack the files and attempt to install the update.
    - Obtained result: The update was installed successfully, so modifying of this file does not affect installation.
- Test 2: Unpack the MSU file, modify the content of the *xml* file, pack the files and attempt to install the update.
    - Obtained result: The update was installed successfully, so modifying of this file does not affect installation.
- Test 3: Unpack the MSU file, delete the *pkgProperties* file, pack the files and attempt to install the update.
    - Obtained result: The update was installed successfully, so deleting this file does not affect installation.
- Test 4: Unpack the MSU file, delete the *xml* file, pack the files and attempt to install the update.
    - Obtained result: The update was installed successfully, so deleting this file does not affect installation.
- Test 5: Unpack the MSU file, modify the content of the *Windows10.0-KB9999991-x64.cab* file, pack the files and attempt to install the update.
    - Obtained result: The update was not installed successfully because the digital signature is not valid, so modifying of this file affect installation.
- Test 6: Unpack the MSU file, modify the content of the *WSUSSCAN.cab* file, pack the files and attempt to install the update.
    - Obtained result: The update was not installed successfully because the digital signature is not valid, so modifying of this file affect installation.

35.3.1.4. Verdict

Analyzing the obtained results, the evaluator considers that the update file is not installed properly, when the modified file is used during the installation. In case of the modified file is not used during the installation, their integrity is not validated and therefore the update is applied properly.

Therefore, the test results obtained during this test activity demonstrate that the Test 1 requirements established in the assurance activity section are properly fulfilled. Due to this, the **PASS** verdict is assigned to Test 1.

35.3.2. Test 2

35.3.2.1. Setup

The original update file downloaded in Test 1 is available.

No additional tools are needed to perform this test.

35.3.2.2. 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.

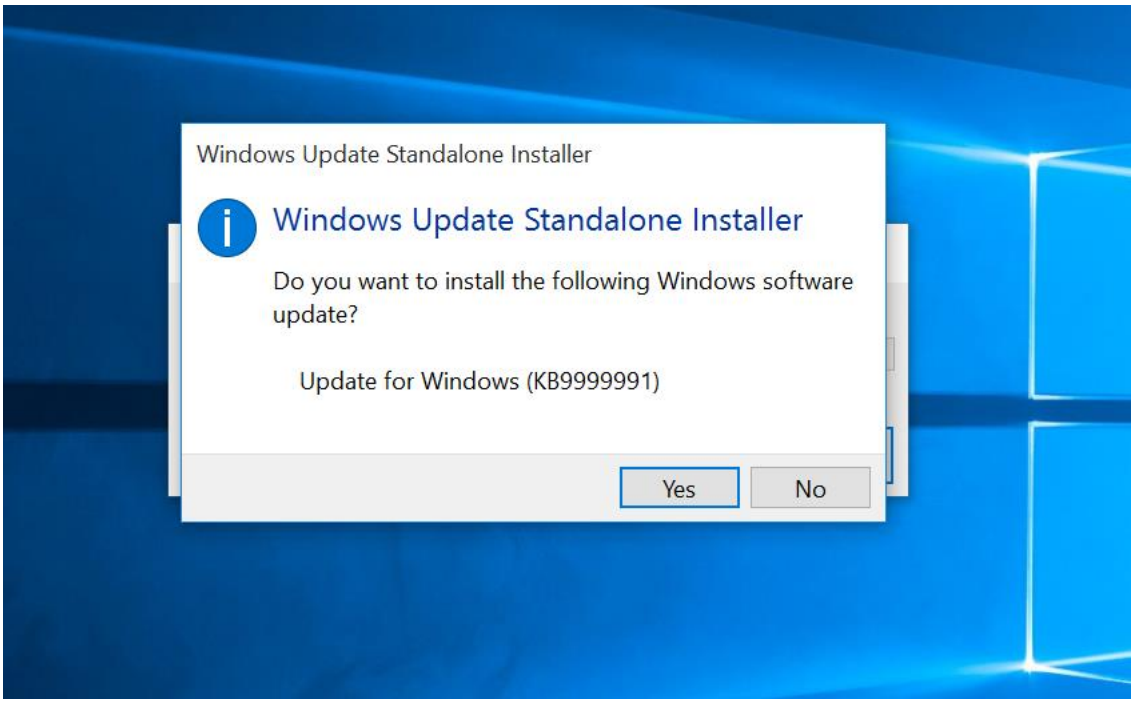
35.3.2.3. Results

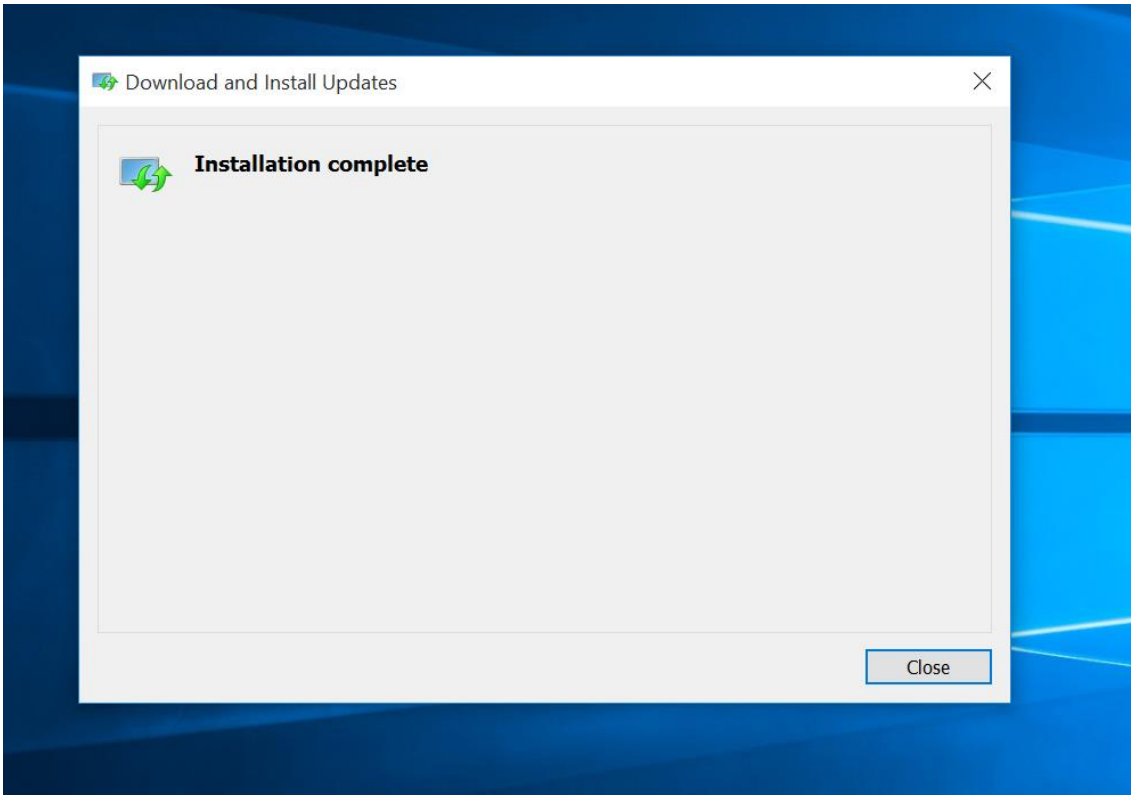
The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- Surface Pro 3 with Windows 10 x64 Enterprise Edition.
- Surface Book with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition.
- Windows Server 2012 R2 Hyper-V with Windows Server 2012 R2 Standard Edition.

The evaluator has attempted to install an update file which has a valid digital signature belonging to the vendor. The obtained results have been the same in all tested platforms; the update has been installed successfully.

The following screenshots, which are taken from one of the tested platform, show this fact:





35.3.2.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 2** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 2**.

35.4. Final Verdict

Due to the documentation review activity and all test activities 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 FPT_TUD_EXT.1.2 requirement.

36. FPT_TUD_EXT.2.1

36.1. Assurance activity

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.)

36.2. Documentation review activity

36.2.1. Findings

The evaluator has reviewed the operational guidance, which includes the following information in its section **12. Managing Updates.**

The following help topics describe how to check for updates to Windows Store installed applications on Windows 10:

- Check for updates for apps and games from Windows Store: http://windows.microsoft.com/en-us/windows-10/check-for-updates-for-apps-and-games-from-windows-store

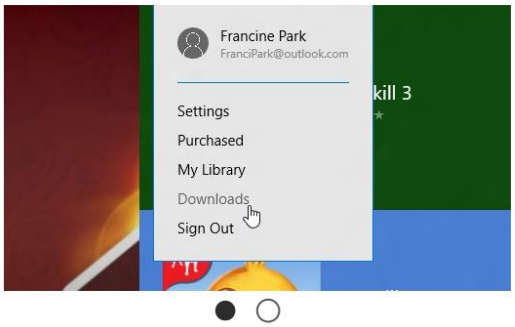
Follow the same procedures on Windows Server 2012 R2 if Desktop Experience is configured:

- Desktop Experience Overview:https://technet.microsoft.com/en-us/library/dn609826.aspx

This section provides two pointers to the vendor support webpage. The first one includes the steps that should be performed in order to check for application updates through Windows Store for Windows 10:

Check for updates for apps and games from Windows Store

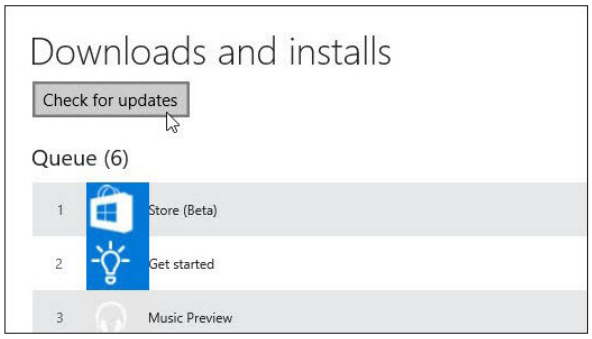
Applies to Windows 10



Step 1: On the menu in the upper right, select **Downloads.**

Check for updates for apps and games from Windows Store

Applies to Windows 10



Step 2: On the download page, select **Check for updates.**

The second one provides information related to the Desktop Experience. This feature allows enable the Windows Store, which is disabled by default. Once the Desktop Experience is installed and the Windows Store is enabled, the steps to check for application updates are the same as the defined one for Windows 10. In order to install the Desktop Experience the following PowerShell command shall be executed:

To install Desktop Experience with Windows PowerShell, use the following commands:

```
Import-Module ServerManager

Install-WindowsFeature Desktop-Experience
```

36.2.2. Verdict

The operational guidance includes in its section **12.Managing Updates** two pointer to the vendor support website. These pointers include enough information to allow the evaluator determines how to check for new application updates. Additionally, this information is provided for both operating systems, Windows 10 and Windows Server 2012 R2.

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.

36.3. Test Activity

36.3.1. Test - Checking for application update over a trusted channel

36.3.1.1. Setup

The following tool must be installed in the platform in order to allow the evaluator perform this test:

- A network protocol analyzer, e.g. Wireshark.

In addition, the Desktop Experience feature must be installed in platforms with Windows Server 2012 R2. To do that, the evaluator shall carry out the following steps:

1. Open a PowerShell terminal as administrator and type the following commands:

   Import-Module ServerManager
   Install-WindowsFeature Desktop-Experience

2. Once the Desktop Experience installation has finished, restart the computer to apply the changes. Note that the administrator built-in is not able to open the Windows Store after its installation. Therefore, in order to perform this test other administrator account shall exist in the computer and shall belong to default *Administrators* group.

Finally, Windows Store must be configured with a Microsoft account before checking for updates.

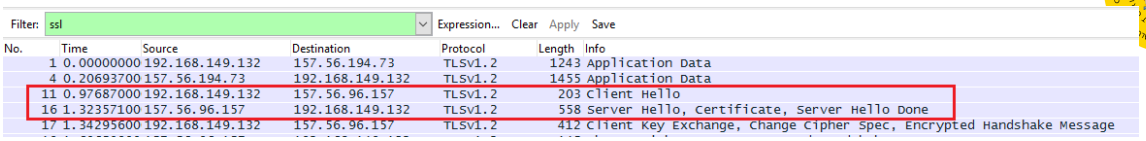
36.3.1.2. Procedure

The following steps must be carried out in order to ensure that the update checking process is performed over a trusted channel as described in FTP_ITC_EXT.1.

1. Open Wireshark and configure it to listen through the active Ethernet interface.

2. Open Windows Store, and follow the shown steps in the Documentation Review activity section in order to check for application updates.

3. 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. In order to make easier the traffic analysis, the evaluator can apply a filter over the capture, 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.



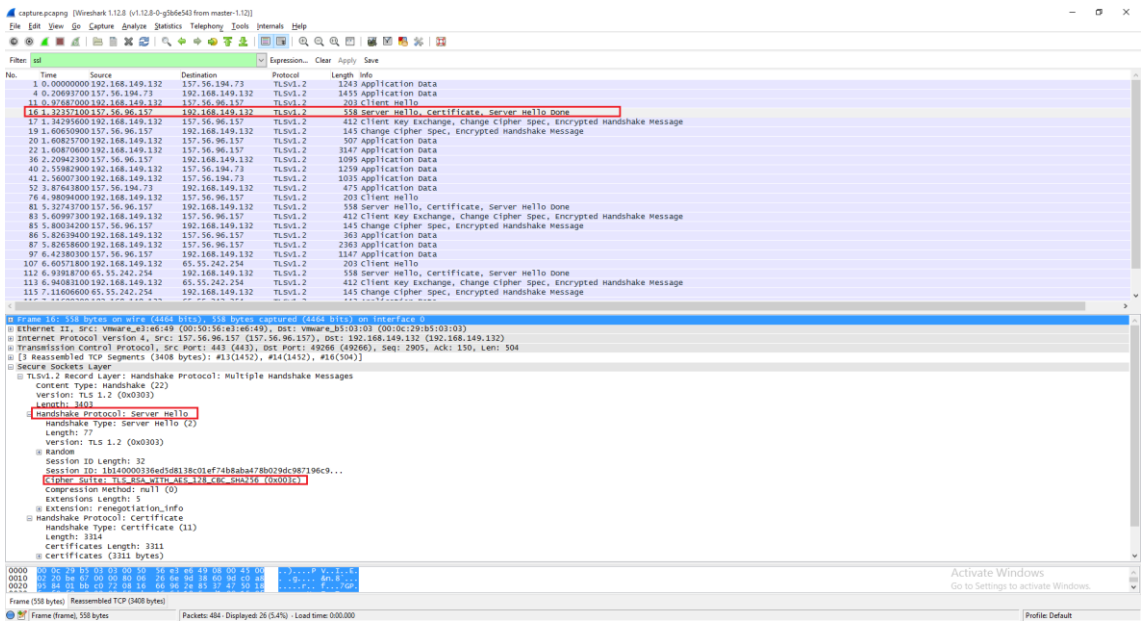
36.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

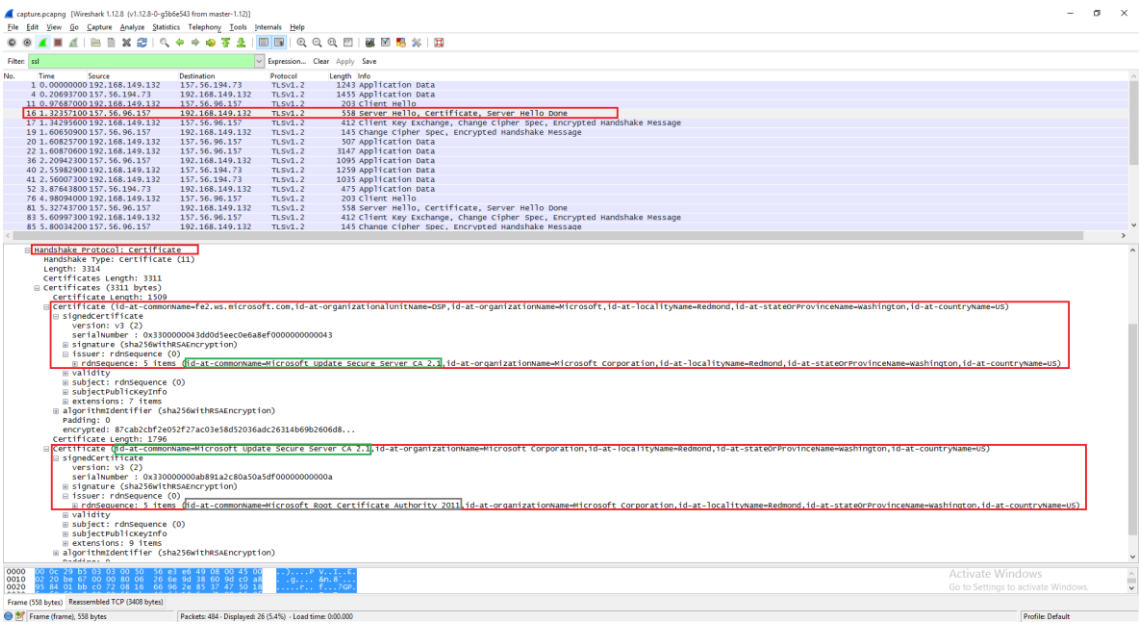
- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition.

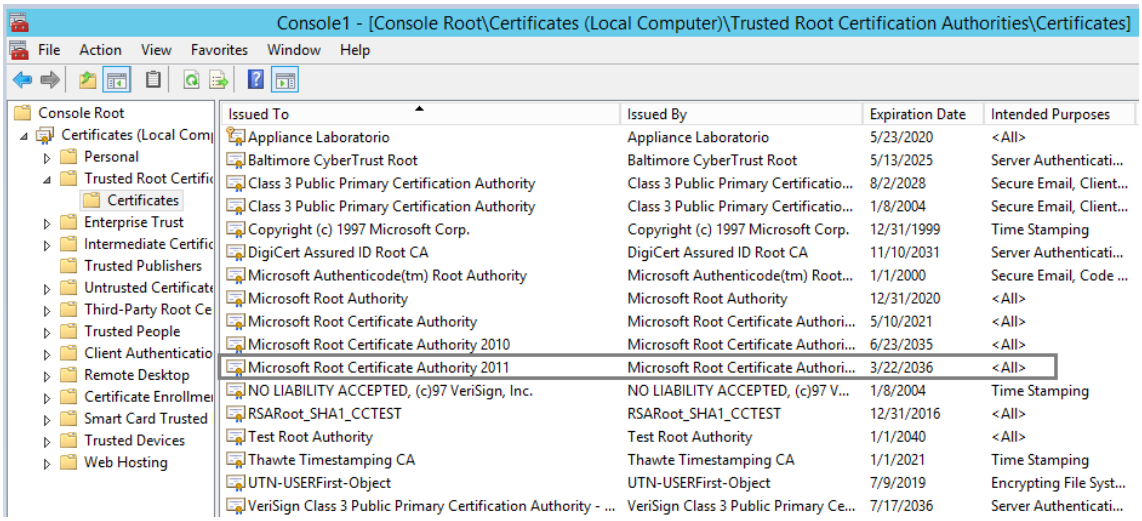
The evaluator has initiated the update checking process and has checked that the connection between the TOE and the update server is established over a secure channel. The evaluator has obtained the same results in all the tested platforms. The following screenshots, which are taken from a Wireshark capture in one of the tested platform, show this fact.

The evaluator has identified the handshake messages between the client and the server in order to configure and establish the secure channel. First of all, the evaluator has identified the cipher suite selected by the server, which matches with one of the cipher suites defined in FCS_TLSC_EXT.1.



Additionally, the server sends the certificate and its certificate chain. The root CA of the certificate chain sent by the server match with one of the root certificate stored in the client machine.





36.3.1.4. Verdict

As the above results state, 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.

The evaluator considers that, the test results obtained during this test activity demonstrate the fulfillment of the test requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to this test.

36.4. Final Verdict

Due to both the documentation review activity and the test activity 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 FPT_TUD_EXT.2.1.

37. FPT_TUD_EXT.2.2

37.1. Assurance activity

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.

37.2. Documentation review activity

37.2.1. Findings

The evaluator has reviewed the information provided in TSS, section **6.6.5.1 Windows Store Applications.** This section include information related the structure of the Windows Store Applications and how their integrity is validated during the installation process.

In addition to this information, a footnote is included. This footnote states that the Windows Store Applications are typically downloaded from the Windows Store, however the appx packages can be copied to the local computer.

Therefore the evaluator can determine that the Windows Store is the main origin of the installation and update supported by the operating system.

37.2.2. Verdict

The evaluator has reviewed the TSS and the information included in its section **6.6.5.1 Windows Store Applications.** The provided information is enough in order to allow the evaluator determine which are all application update origins supported by the operating system.

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

37.3. Test Activity

37.3.1. Test 1

37.3.1.1. Setup

The following tools must be installed in the tested platform:

- 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. WinHex.
- A compression file tools must be installed in the tested platform.
- The application installer file used during the test procedure shall be available in the tested platforms. The needed files to install the application are the following:
    - RemoteDesktopApp.appxbundle, Remote Desktop Application installer file.
    - Microsoft.WinJS.2.0.appx, this file is required due to dependencies.
- Additionally, the vendor has provided an application file with two different versions. These files shall be used in some test cases.

37.3.1.2. Procedure

The evaluator shall carry out the following two approaches in order to invalidate the digital signature.

A. Approach 1 - Digital signature modification:

1. Create a copy of the application installer file, which is going to be modified.
2. Open a cmd terminal and go to the folder where SignTool is installed (typical path is ***%programfiles(x86)%\WindowsKits\10\bin\x86***).
3. Obtain the digital signature typing the following command: "*signtool.exe verify /all /pa /v updatefile.msu*". The signature will be shown in *Hash of file (sha256)* field.

4. On the other hand, open the application installer file (*RemoteDesktopApp.appxbundle)* with a compressor file tool and extract the *AppxSignature.p7x.* This file contains the digital signature of the application installer file.

5. Open the obtained file in WinHex editor and search the value obtained at the step 3. Modify some bytes of the digital signature and save the file.

6. Replace the original *AppxSignature.p7x* with the modified one using the file compressor tool. Note that, the application installer has not been unpacked, therefore its internal structure is still valid.

7. The digital signature can be checked using the Signtool utility and typing the following command:

   signtool /verify /all /pa /v RemoteDesktopApp.appxbundle

   Ensure that digital signature is no longer valid after the modification.

8. 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-AppxPackage RemoteDesktopApp.appxbundle -DependencyPath Microsoft.WinJS.2.0.appx

B. Approach 2 - Testing using vendor applications:

The evaluator shall perform the following test using the application files provided by the vendor:

1. Modifying a file used during the installation.
2. Modifying a file not used during the installation.
3. Modifying a file used during the update.
4. Modifying a file not used during the update.

37.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.
- HP Pro x2 612 with Windows 10 x64 Pro Edition.

The evaluator has attempted to install a modified application installer file and has obtained different results depending on the used approach. These results have been the same in both tested platforms.

The results obtained during the test execution are described as follows:

A. Approach 1 – Digital signature modification:

The evaluator has executed the *signtool* command and has obtained SHA-256 hash from the application installer file:

```
Verifying: AppxSignature.p7x
Signature Index: 0 (Primary Signature)
Hash of file (sha256): 4150505841585043DC11421E7E314F4D45E6E5C53E3E31AD10621D4E1DD74C0696DF6C0B6D437F994158434 49A17
        869FB9659576E88C9C41E17A61B426CECBE1DBBFC5015FE44551054DCEBF415843548AE94B77A2A3A7C0123A95DA7311FFF3C947B0D
        F40E003FF5EA69492E7B8EEAA4158424D88B41F9662C397CAB39B041C52A08890BFB5CD34A59C1AAACF08F030A15687A0

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority 2011
    Issued by: Microsoft Root Certificate Authority 2011
    Expires:   Sat Mar 22 23:13:04 2036
    SHA1 hash: 8F43288AD272F3103B6FB1428485EA3014C0BCFE

        Issued to: Microsoft MarketPlace PCA 2011
        Issued by: Microsoft Root Certificate Authority 2011
        Expires:   Fri Mar 28 22:19:39 2031
        SHA1 hash: EB9204785126C26ADADCBBD44FE2B0C642A71078

            Issued to: Microsoft Marketplace CA 006
            Issued by: Microsoft MarketPlace PCA 2011
            Expires:   Fri Nov 07 23:41:12 2014
            SHA1 hash: 401AF7A24DBE2EDA61C914412301D8987E610A21

                Issued to: Microsoft Corporation
                Issued by: Microsoft Marketplace CA 006
                Expires:   Sat Oct 12 21:43:21 2013
                SHA1 hash: B7961F97BA65B1C6FBA4A79D22FBEB6CFBC9F010

The signature is timestamped: Wed Oct 09 21:43:58 2013
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 Jun 27 21:13:14 2014
            SHA1 hash: 74C2D86CBB54F52C6B69BC0A127F427EF60A8B8A

Successfully verified: AppxSignature.p7x

Number of signatures successfully verified: 1
Number of warnings: 0
Number of errors: 0
```

Then, the evaluator has extracted the *AppxSignature.p7x* file and has opened it in WinHex editor. After that the evaluator has searched the hash obtained above. This hash is at the offset 0x00000090. The original hash is as follows:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	50	4B	43	58	30	82	29	34	06	09	2A	86	48	86	F7	0D	PKCX0▮)4..*▮H▮÷.
00000010	01	07	02	A0	82	29	25	30	82	29	21	02	01	01	31	0F	... ▮)%0▮)!...1.
00000020	30	0D	06	09	60	86	48	01	65	03	04	02	01	05	00	30	0...`▮H.e.....0
00000030	81	F2	06	0A	2B	06	01	04	01	82	37	02	01	04	A0	81	.ò..+....▮7....
00000040	E3	30	81	E0	30	35	06	0A	2B	06	01	04	01	82	37	02	ã0.à05..+....▮7.
00000050	01	1E	30	27	02	04	01	01	00	00	04	10	B3	58	5F	0F	..0'.........³X_.
00000060	DE	AA	9A	4B	A4	34	95	74	2D	92	EC	EB	02	01	00	02	Þª▮K¤4▮t-´ìë....
00000070	01	00	02	01	00	02	01	00	02	01	00	30	81	A6	30	0D	...........0.¦0.
00000080	06	09	60	86	48	01	65	03	04	02	01	05	00	04	81	94	..`▮H.e........▮
00000090	41	50	50	58	41	58	50	43	DC	11	42	1E	7E	31	4F	4D	APPXAXPCÜ.B.~1OM
000000A0	45	E6	E5	C5	3E	3E	31	AD	10	62	1D	4E	1D	D7	4C	06	EæåÅ>>1-.b.N.×L.
000000B0	96	DF	6C	0B	6D	43	7F	99	41	58	43	44	9A	17	86	9F	▮ßl.mC.▮AXCD▮.▮▮
000000C0	B9	65	95	76	E8	8C	9C	41	E1	7A	61	B4	26	CE	CB	E1	¹e▮vè▮▮Aáza´&ÎËá
000000D0	DB	BF	C5	01	5F	E4	45	51	05	4D	CE	BF	41	58	43	54	Û¿Å._äEQ.MÎ¿AXCT
000000E0	8A	E9	4B	77	A2	A3	A7	C0	12	3A	95	DA	73	11	FF	F3	▮éKw¢£§À.:▮Ús.ÿó
000000F0	C9	47	B0	DF	40	E0	03	FF	5E	A6	94	92	E7	B8	EE	AA	ÉG°ß@à.ÿ^¦▮´ç¸îª
00000100	41	58	42	4D	88	B4	1F	96	62	C3	97	CA	B3	9B	04	1C	AXBM▮´.▮bÃ▮Ê³▮..
00000110	52	A0	88	90	BF	B5	CD	34	A5	9C	1A	AA	CF	08	F0	30	R ▮.¿µÍ4¥▮.ªÏ.ð0
00000120	A1	56	87	A0	A0	82	12	2E	30	82	05	75	30	82	04	5D	¡V▮ ▮..0▮.u0▮.]
00000130	A0	03	02	01	02	02	0A	48	A1	D4	62	00	00	00	00	22	.....HiÔb....."
00000140	14	30	0D	06	09	2A	86	48	86	F7	0D	01	01	0B	05	00	.0...*▮H▮÷.....
00000150	30	81	8A	31	0B	30	09	06	03	55	04	06	13	02	55	53	0.▮1.0...U....US
00000160	31	13	30	11	06	03	55	04	08	13	0A	57	61	73	68	69	1.0...U....Washi
00000170	6E	67	74	6F	6E	31	10	30	0E	06	03	55	04	07	13	07	ngton1.0...U....
00000180	52	65	64	6D	6F	6E	64	31	1E	30	1C	06	03	55	04	0A	Redmond1.0...U..
00000190	13	15	4D	69	63	72	6F	73	6F	66	74	20	43	6F	72	70	..Microsoft Corp
000001A0	6F	72	61	74	69	6F	6E	31	0D	30	0B	06	03	55	04	0B	oration1.0...U..
000001B0	13	04	4D	4F	50	52	31	25	30	23	06	03	55	04	03	13	..MOPR1%0#..U...
000001C0	1C	4D	69	63	72	6F	73	6F	66	74	20	4D	61	72	6B	65	.Microsoft Marke
000001D0	74	70	6C	61	63	65	20	43	41	20	30	30	36	30	1E	17	tplace CA 0060..
000001E0	0D	31	33	31	30	30	39	32	30	33	33	32	31	5A	17	0D	.131009203321Z..
000001F0	31	33	31	30	31	32	32	30	34	33	32	31	5A	30	74	31	131012204321Z0t1

Once the hash has been modified:

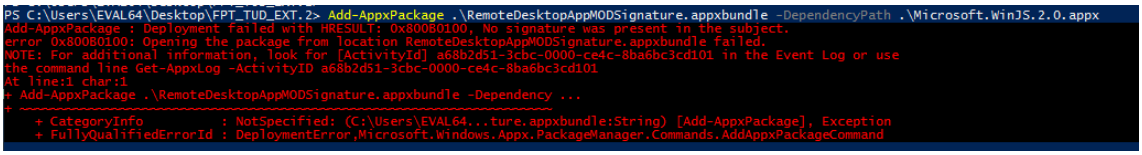
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	50	4B	43	58	30	82	29	34	06	09	2A	86	48	86	F7	0D	PKCX0▮)4..*▮H▮÷.
00000010	01	07	02	A0	82	29	25	30	82	29	21	02	01	01	31	0F	... ▮)%0▮)!...1.
00000020	30	0D	06	09	60	86	48	01	65	03	04	02	01	05	00	30	0...`▮H.e.....0
00000030	81	F2	06	0A	2B	06	01	04	01	82	37	02	01	04	A0	81	.ò..+....▮7... .
00000040	E3	30	81	E0	30	35	06	0A	2B	06	01	04	01	82	37	02	ã0.à05..+....▮7.
00000050	01	1E	30	27	02	04	01	01	00	00	04	10	B3	58	5F	0F	..0'.........³X_.
00000060	DE	AA	9A	4B	A4	34	95	74	2D	92	EC	EB	02	01	00	02	Þª▮K¤4▮t-'ìë....
00000070	01	00	02	01	00	02	01	00	02	01	00	30	81	A6	30	0D	...........0.¦0.
00000080	06	09	60	86	48	01	65	03	04	02	01	05	00	04	81	94	..`▮H.e.........▮
00000090	41	50	50	58	41	58	50	43	DC	11	42	1E	7E	31	4F	4D	APPXAXPCÜ.B.~1OM
000000A0	45	E6	E5	C5	3E	3E	31	AD	10	62	1D	4E	1D	D7	4C	06	EæåÅ>>1-.b.N.×L.
000000B0	96	DF	6C	0B	6D	43	7F	99	41	58	43	44	9A	17	86	9F	▮ßl.mC.▮AXCD▮.▮▮
000000C0	B9	65	95	76	AA	AA	AA	AA	B1	7A	61	B4	26	CE	CB	E1	¹e▮vªªªªªza´&ÎËá
000000D0	DB	BF	C5	01	5F	E4	45	51	05	4D	CE	BF	41	58	43	54	Û¿Å._äEQ.MÎ¿AXCT
000000E0	8A	E9	4B	77	A2	A3	A7	C0	12	3A	95	DA	73	11	FF	F3	▮éKw¢£§À.:▮Ús.ÿó
000000F0	C9	47	B0	DF	40	E0	03	FF	5E	A6	94	92	E7	B8	EE	AA	ÉG°ß@à.ÿ^¦▮'ç¸îª
00000100	41	58	42	4D	88	B4	1F	96	62	C3	97	CA	B3	9B	04	1C	AXBM▮´.▮bÃ▮Ê³▮..
00000110	52	A0	88	90	BF	B5	CD	34	A5	9C	1A	AA	CF	08	F0	30	R ▮.¿µÍ4¥▮.ªÏ.ð0
00000120	A1	56	87	A0	A0	82	12	2E	30	82	05	75	30	82	04	5D	¡V▮ ▮..0▮.u0▮.]
00000130	A0	03	02	01	02	02	0A	48	A1	D4	62	00	00	00	00	22	.....H¡Ôb....."
00000140	14	30	0D	06	09	2A	86	48	86	F7	0D	01	01	0B	05	00	.0...*▮H▮÷.....
00000150	30	81	8A	31	0B	30	09	06	03	55	04	06	13	02	55	53	0.▮1.0...U....US
00000160	31	13	30	11	06	03	55	04	08	13	0A	57	61	73	68	69	1.0...U....Washi
00000170	6E	67	74	6F	6E	31	10	30	0E	06	03	55	04	07	13	07	ngton1.0...U....
00000180	52	65	64	6D	6F	6E	64	31	1E	30	1C	06	03	55	04	0A	Redmond1.0...U..
00000190	13	15	4D	69	63	72	6F	73	6F	66	74	20	43	6F	72	70	..Microsoft Corp
000001A0	6F	72	61	74	69	6F	6E	31	0D	30	0B	06	03	55	04	0B	oration1.0...U..
000001B0	13	04	4D	4F	50	52	31	25	30	23	06	03	55	04	03	13	..MOPR1%0#..U...
000001C0	1C	4D	69	63	72	6F	73	6F	66	74	20	4D	61	72	6B	65	.Microsoft Marke
000001D0	74	70	6C	61	63	65	20	43	41	20	30	30	36	30	1E	17	tplace CA 0060..
000001E0	0D	31	33	31	30	30	39	32	30	33	33	32	31	5A	17	0D	.131009203321Z..
000001F0	31	33	31	30	31	32	32	30	34	33	32	31	5A	30	74	31	131012204321Z0t1

The evaluator has checked the digital signature after the modification. The result obtained states that the digital signature has not been verified.

```
C:\Users\EVAL64\Desktop\FPT_TUD_EXT.2>signtool verify /all /pa /v RemoteDesktopAppMODSignature.appxbundle > signtoolMODSignature.txt
SignTool Error: WinVerifyTrust returned error: 0x80096010
        The digital signature of the object did not verify.
```

```
Verifying: RemoteDesktopAppMODSignature.appxbundle
Signature Index: 0 (Primary Signature)
Hash of file (sha256): 4150505841585043DC11421E7E314F4D45E6E5C53E3E31AD10621D4E1DD74C0696DF6C
        0B6D437F99415843449A17869FB9659576AAAAAAAAE17A61B426CECBE1DBBFC5015FE44551054DCEBF415
        843548AE94B77A2A3A7C0123A95DA7311FFF3C947B0DF40E003FF5EA69492E7B8EEAA4158424D88B41F96
        62C397CAB39B041C52A08890BFB5CD34A59C1AAACF08F030A15687A0

Signing Certificate Chain:
    Issued to: Microsoft Root Certificate Authority 2011
    Issued by: Microsoft Root Certificate Authority 2011
    Expires:   Sat Mar 22 23:13:04 2036
    SHA1 hash: 8F43288AD272F3103B6FB1428485EA3014C0BCFE

        Issued to: Microsoft MarketPlace PCA 2011
        Issued by: Microsoft Root Certificate Authority 2011
        Expires:   Fri Mar 28 22:19:39 2031
        SHA1 hash: EB9204785126C26ADADCBBD44FE2B0C642A71078

            Issued to: Microsoft Marketplace CA 006
            Issued by: Microsoft MarketPlace PCA 2011
            Expires:   Fri Nov 07 23:41:12 2014
            SHA1 hash: 401AF7A24DBE2EDA61C914412301D8987E610A21

                Issued to: Microsoft Corporation
                Issued by: Microsoft Marketplace CA 006
                Expires:   Sat Oct 12 21:43:21 2013
                SHA1 hash: B7961F97BA65B1C6FBA4A79D22FBEB6CFBC9F010

The signature is timestamped: Wed Oct 09 21:43:58 2013
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 Jun 27 21:13:14 2014
            SHA1 hash: 74C2D86CBB54F52C6B69BC0A127F427EF60A8B8A

Number of signatures successfully Verified: 0
Number of warnings: 0
Number of errors: 1
```

After that, the evaluator has attempted to install the modified application installer file. In this case, the modification is detected during the installation and it is not been installed. The following error screen is shown:



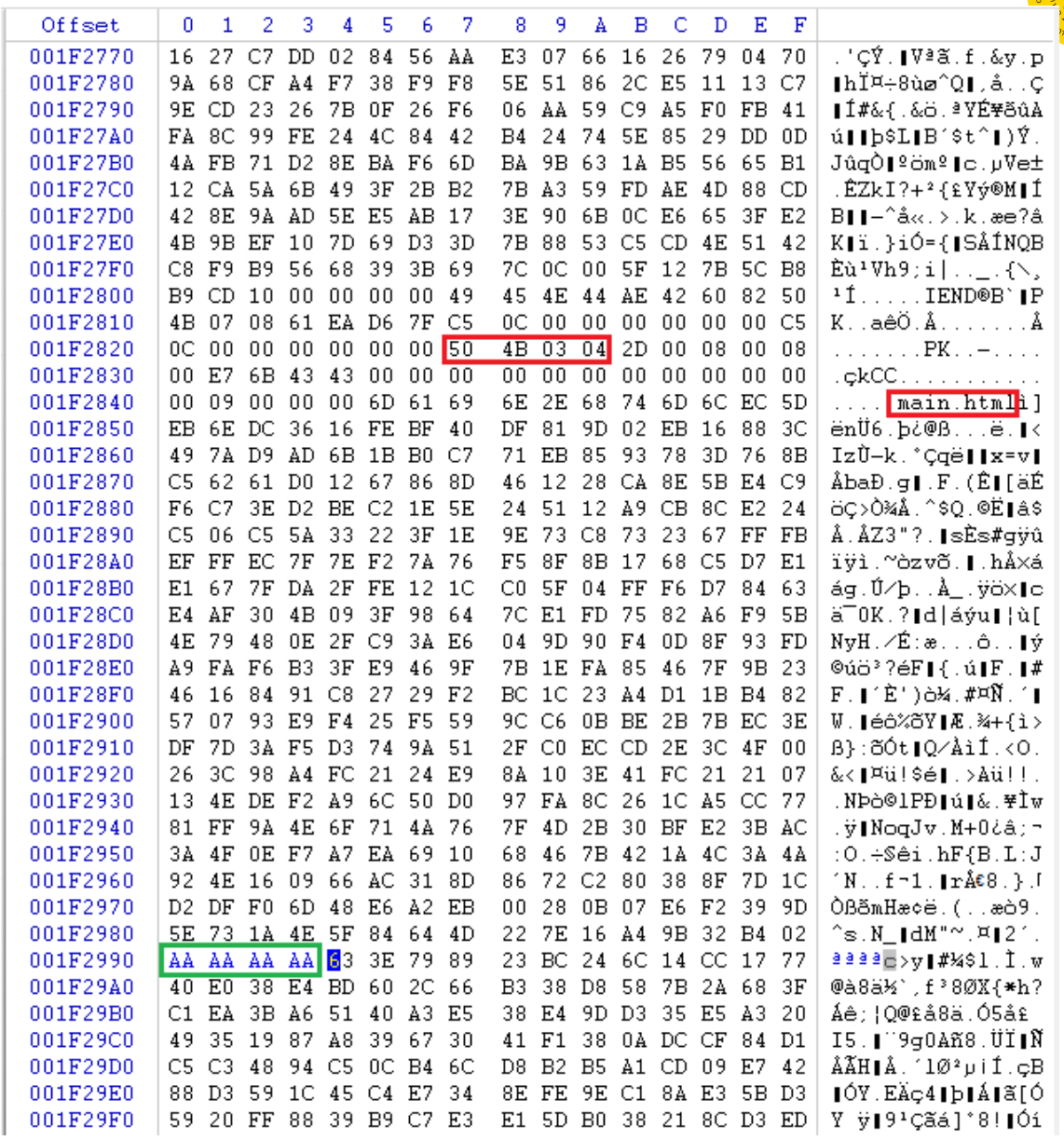
B. Approach 2 – Testing using vendor applications:

The evaluator has obtained the following results:

- **Test 1 - Modifying a file used during installation**: The target package is *remotedesktop_AMD64.appx.* This package contains the main application source files (e.g. *main.html*), so after the modification the application should not be installed. Prior to the modification, the evaluator has identified the file offset using the magic number:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
001D16F0	00	00	00	00	2D	00	2D	00	00	00	00	00	00	00	00	00	....-.-.........
001D1700	28	00	00	00	00	00	00	00	28	00	00	00	00	00	00	00	(.......(.......
001D1710	E4	0C	00	00	00	00	00	00	65	7D	02	00	00	00	00	00	ä.......e}......
001D1720	50	4B	06	07	00	00	00	00	49	8A	02	00	00	00	00	00	PK......I▮......
001D1730	01	00	00	00	50	4B	05	06	00	00	00	00	FF	FF	FF	FF	....PK......ÿÿÿÿ
001D1740	FF	FF	FF	FF	FF	FF	FF	FF	00	00	50	4B	07	08	C2	E3	ÿÿÿÿÿÿÿÿ..PK..Âã
001D1750	88	13	AB	8A	02	00	00	00	00	00	AB	8A	02	00	00	00	▮.«▮......«▮....
001D1760	00	00	50	4B	03	04	2D	00	08	00	00	00	80	85	49	43	..PK..-.....▮▮IC
001D1770	00	00	00	00	00	00	00	00	00	00	00	00	18	00	00	00	....
001D1780	72	65	6D	6F	74	65	64	65	73	6B	74	6F	70	5F	41	4D	remotedesktop_AM
001D1790	44	36	34	2E	61	70	70	78	50	4B	03	04	2D	00	08	00	D64.appxPK..-...
001D17A0	08	00	E7	6B	43	43	00	00	00	00	00	00	00	00	00	00	..çkCC..........
001D17B0	00	00	18	00	00	00	75	6E	64	2F	6C	6F	63	61	6C	69	......und/locali
001D17C0	7A	61	62	6C	65	73	74	79	6C	65	2E	63	73	73	BC	95	zablestyle.css¼▮
001D17D0	DF	6E	D3	30	14	C6	AF	A9	D4	77	B0	76	05	53	9D	B5	ßnÓ0.Æ¯©Ôw°v.S.µ
001D17E0	29	1B	52	7B	07	12	30	89	09	A4	8E	07	B0	93	D3	D4	).R{..0▮.¤▮.°▮ÓÔ
001D17F0	DA	89	1D	6C	67	5D	41	7B	32	2E	78	24	5E	81	93	3F	Ú▮.lg]A{2.x$^.▮?
001D1800	A4	A9	B2	56	59	D9	A8	94	C4	F5	B1	7B	7E	FE	72	7A	¤©²VYÙ¨▮Äõ±{~þrz
001D1810	BE	DF	3F	7F	05	1F	AC	C9	B3	8F	20	62	B0	EF	8D	F6	¾ß?...¬É³. b°ï.ö
001D1820	C3	C1	8F	E1	80	D1	67	49	5F	B8	53	DF	61	C6	C2	49	ÃÁ.á▮ÑgI_¸SßaÆÂI
001D1830	E6	E7	AD	D9	35	A8	64	E5	67	0C	8B	07	58	0A	DD	0F	æç–Ù5¨dåg.▮.X.Ý.

And after that the *main.html* file, which belongs to *remotedesktop_AMD64.appx*, is identified at offset 0x001F2820. The file modification is performed at offset 0x001F2990 as the following image shows:

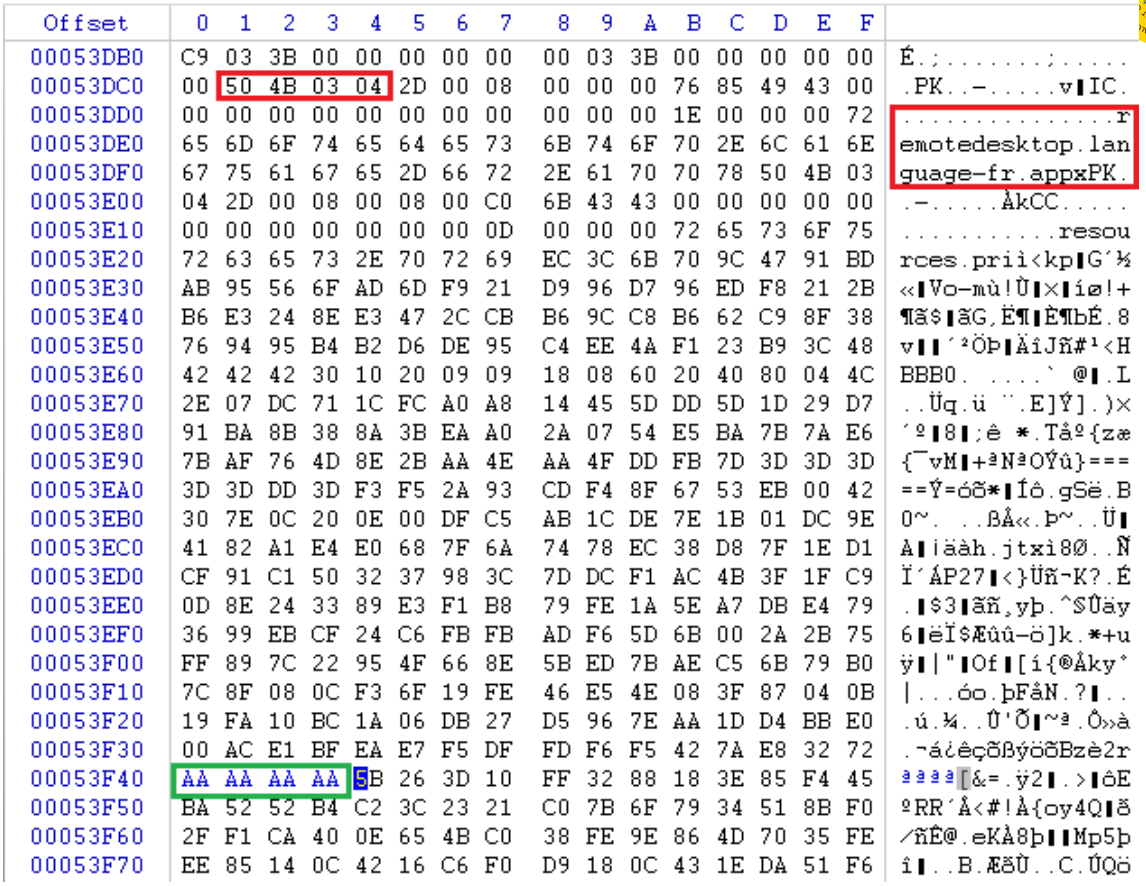


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
001F2770	16	27	C7	DD	02	84	56	AA	E3	07	66	16	26	79	04	70
001F2780	9A	68	CF	A4	F7	38	F9	F8	5E	51	86	2C	E5	11	13	C7
001F2790	9E	CD	23	26	7B	0F	26	F6	06	AA	59	C9	A5	F0	FB	41
001F27A0	FA	8C	99	FE	24	4C	84	42	B4	24	74	5E	85	29	DD	0D
001F27B0	4A	FB	71	D2	8E	BA	F6	6D	BA	9B	63	1A	B5	56	65	B1
001F27C0	12	CA	5A	6B	49	3F	2B	B2	7B	A3	59	FD	AE	4D	88	CD
001F27D0	42	8E	9A	AD	5E	E5	AB	17	3E	90	6B	0C	E6	65	3F	E2
001F27E0	4B	9B	EF	10	7D	69	D3	3D	7B	88	53	C5	CD	4E	51	42
001F27F0	C8	F9	B9	56	68	39	3B	69	7C	0C	00	5F	12	7B	5C	B8
001F2800	B9	CD	10	00	00	00	00	49	45	4E	44	AE	42	60	82	50
001F2810	4B	07	08	61	EA	D6	7F	C5	0C	00	00	00	00	00	00	C5
001F2820	0C	00	00	00	00	00	00	50	4B	03	04	2D	00	08	00	08
001F2830	00	E7	6B	43	43	00	00	00	00	00	00	00	00	00	00	00
001F2840	00	09	00	00	00	6D	61	69	6E	2E	68	74	6D	6C	EC	5D
001F2850	EB	6E	DC	36	16	FE	BF	40	DF	81	9D	02	EB	16	88	3C
001F2860	49	7A	D9	AD	6B	1B	B0	C7	71	EB	85	93	78	3D	76	8B
001F2870	C5	62	61	D0	12	67	86	8D	46	12	28	CA	8E	5B	E4	C9
001F2880	F6	C7	3E	D2	BE	C2	1E	5E	24	51	12	A9	CB	8C	E2	24
001F2890	C5	06	C5	5A	33	22	3F	1E	9E	73	C8	73	23	67	FF	FB
001F28A0	EF	FF	EC	7F	7E	F2	7A	76	F5	8F	8B	17	68	C5	D7	E1
001F28B0	E1	67	7F	DA	2F	FE	12	1C	C0	5F	04	FF	F6	D7	84	63
001F28C0	E4	AF	30	4B	09	3F	98	64	7C	E1	FD	75	82	A6	F9	5B
001F28D0	4E	79	48	0E	2F	C9	3A	E6	04	9D	90	F4	0D	8F	93	FD
001F28E0	A9	FA	F6	B3	3F	E9	46	9F	7B	1E	FA	85	46	7F	9B	23
001F28F0	46	16	84	91	C8	27	29	F2	BC	1C	23	A4	D1	1B	B4	82
001F2900	57	07	93	E9	F4	25	F5	59	9C	C6	0B	BE	2B	7B	EC	3E
001F2910	DF	7D	3A	F5	D3	74	9A	51	2F	C0	EC	CD	2E	3C	4F	00
001F2920	26	3C	98	A4	FC	21	24	E9	8A	10	3E	41	FC	21	21	07
001F2930	13	4E	DE	F2	A9	6C	50	D0	97	FA	8C	26	1C	A5	CC	77
001F2940	81	FF	9A	4E	6F	71	4A	76	7F	4D	2D	30	BF	E2	3B	AC
001F2950	3A	4F	0E	F7	A7	EA	69	10	68	46	7B	42	1A	4C	3A	4A
001F2960	92	4E	16	09	66	AC	31	8D	86	72	C2	80	38	8F	7D	1C
001F2970	D2	DF	F0	6D	48	E6	A2	EB	00	28	0B	07	E6	F2	39	9D
001F2980	5E	73	1A	4E	5F	84	64	4D	22	7E	16	A4	9B	32	B4	02
001F2990	AA	AA	AA	AA	63	3E	79	89	23	BC	24	6C	14	CC	17	77
001F29A0	40	E0	38	E4	BD	60	2C	66	B3	38	D8	58	7B	2A	68	3F
001F29B0	C1	EA	3B	A6	51	40	A3	E5	38	E4	9D	D3	35	E5	A3	20
001F29C0	49	35	19	87	A8	39	67	30	41	F1	38	0A	DC	CF	84	D1
001F29D0	C5	C3	48	94	C5	0C	B4	6C	D8	B2	B5	A1	CD	09	E7	42
001F29E0	88	D3	59	1C	45	C4	E7	34	8E	FE	9E	C1	8A	E3	5B	D3
001F29F0	59	20	FF	88	39	B9	C7	E3	E1	5D	B0	38	21	8C	D3	ED

Finally, the evaluator has attempted to install the modified application and it is not installed properly. Therefore the obtained result is the same as the expected one.

- **Test 2 - Modifying a file NOT used during installation:** The target package is *remotedesktop.language-fr.appx*. This package contains the required files to show the application texts in French. So, after the modification and due to the configured system language is *en-US* (English, United States) the application should be installed successfully.

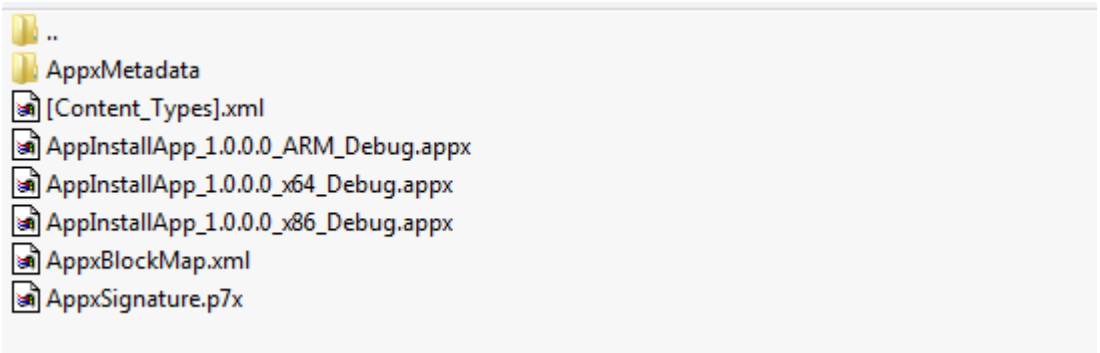
  Prior to the modification, the evaluator has identified the *remotedesktop.language-fr.appx* file at offset 0x00053DC0 using the magic number. The file modification is performed at offset 0x00053F40 (this offset corresponds to *resource.pri* file, which belongs to the *remotedesktop.language-fr.appx*). The following image shows these facts:



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00053DB0	C9	03	3B	00	00	00	00	00	00	03	3B	00	00	00	00	00	É.;.....;.....
00053DC0	00	50	4B	03	04	2D	00	08	00	00	00	76	85	49	43	00	.PK..-.....v▮IC.
00053DD0	00	00	00	00	00	00	00	00	00	00	00	1E	00	00	00	72	...............r
00053DE0	65	6D	6F	74	65	64	65	73	6B	74	6F	70	2E	6C	61	6E	emotedesktop.lan
00053DF0	67	75	61	67	65	2D	66	72	2E	61	70	70	78	50	4B	03	guage-fr.appxPK.
00053E00	04	2D	00	08	00	08	00	C0	6B	43	43	00	00	00	00	00	.-.....ÀkCC.....
00053E10	00	00	00	00	00	00	00	0D	00	00	00	72	65	73	6F	75	...........resou
00053E20	72	63	65	73	2E	70	72	69	EC	3C	6B	70	9C	47	91	BD	rces.prii<kp▮G´½
00053E30	AB	95	56	6F	AD	6D	F9	21	D9	96	D7	96	ED	F8	21	2B	«▮Vo–mù!Ù▮×▮íø!+
00053E40	B6	E3	24	8E	E3	47	2C	CB	B6	9C	C8	B6	62	C9	8F	38	¶ã$▮ãG,Ë¶▮È¶bÉ.8
00053E50	76	94	95	B4	B2	D6	DE	95	C4	EE	4A	F1	23	B9	3C	48	v▮▮´²ÖÞ▮ÄîJñ#¹<H
00053E60	42	42	42	30	10	20	09	09	18	08	60	20	40	80	04	4C	BBB0. ....` @▮.L
00053E70	2E	07	DC	71	1C	FC	A0	A8	14	45	5D	DD	5D	1D	29	D7	..Üq.ü ¨.E]Ý].)×
00053E80	91	BA	8B	38	8A	3B	EA	A0	2A	07	54	E5	BA	7B	7A	E6	´º▮8▮;ê *.Tåº{zæ
00053E90	7B	AF	76	4D	8E	2B	AA	4E	AA	4F	DD	FB	7D	3D	3D	3D	{¯vM▮+ªNªOÝû}===
00053EA0	3D	3D	DD	3D	F3	F5	2A	93	CD	F4	8F	67	53	EB	00	42	==Ý=óõ*▮Íô.gSë.B
00053EB0	30	7E	0C	20	0E	00	DF	C5	AB	1C	DE	7E	1B	01	DC	9E	0~. ..ßÅ«.Þ~..Ü▮
00053EC0	41	82	A1	E4	E0	68	7F	6A	74	78	EC	38	D8	7F	1E	D1	A▮¡äàh.jtxì8Ø..Ñ
00053ED0	CF	91	C1	50	32	37	98	3C	7D	DC	F1	AC	4B	3F	1F	C9	Ï´ÁP27▮<}Üñ¬K?.É
00053EE0	0D	8E	24	33	89	E3	F1	B8	79	FE	1A	5E	A7	DB	E4	79	.▮$3▮ãñ¸yþ.^§Ûäy
00053EF0	36	99	EB	CF	24	C6	FB	FB	AD	F6	5D	6B	00	2A	2B	75	6▮ëÏ$Æûû–ö]k.*+u
00053F00	FF	89	7C	22	95	4F	66	8E	5B	ED	7B	AE	C5	6B	79	B0	ÿ▮|"▮Of▮[í{®Åky°
00053F10	7C	8F	08	0C	F3	6F	19	FE	46	E5	4E	08	3F	87	04	0B	|...óo.þFåN.?▮..
00053F20	19	FA	10	BC	1A	06	DB	27	D5	96	7E	AA	1D	D4	BB	E0	.ú.¼..Û'Õ▮~ª.Ô»à
00053F30	00	AC	E1	BF	EA	E7	F5	DF	FD	F6	F5	42	7A	E8	32	72	.¬á¿êçõßýöõBzè2r
00053F40	AA	AA	AA	AA	5B	26	3D	10	FF	32	88	18	3E	85	F4	45	ªªªª[&=.ÿ2▮.>▮ôE
00053F50	BA	52	52	B4	C2	3C	23	21	C0	7B	6F	79	34	51	8B	F0	ºRR´Â<#!À{oy4Q▮ð
00053F60	2F	F1	CA	40	0E	65	4B	C0	38	FE	9E	86	4D	70	35	FE	/ñÊ@.eKÀ8þ▮▮Mp5þ
00053F70	EE	85	14	0C	42	16	C6	F0	D9	18	0C	43	1E	DA	51	F6	î▮..B.ÆðÙ..C.ÚQö

Finally, the evaluator has attempted to install the modified application and it is installed properly due to the modified file is not used during the installation. Therefore the obtained result is the same as the expected one.

In order to check that the behavior during application update is the same as the one obtained above, the evaluator has performed two additional tests. To do that, the evaluator has used two applications with different version provided by the vendor in *.appxbundle* format. Each *.appxbundle* contains three *.appx* files, one for each architecture (ARM, 32-bits and 64-bits).

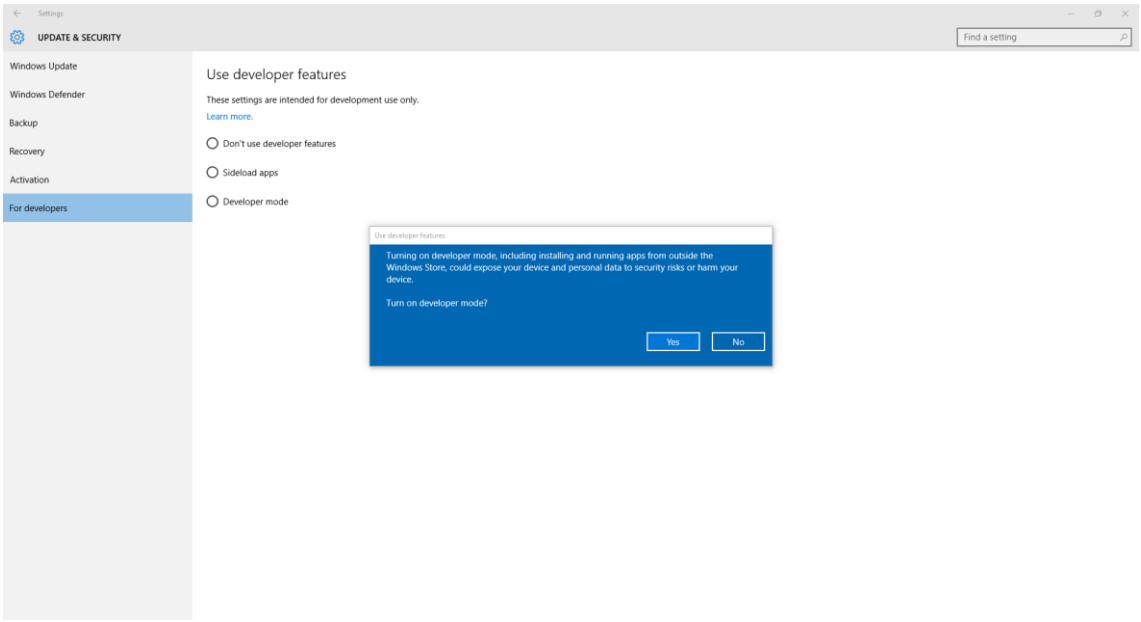


Each *.appx* file contains a file named *resources.pri*, and this file is the one is going to be modified. Using the WinHex editor, the evaluator has identified the offset when each *resources.pri* file starts for each *.appx* file in the *.appxbundle* file.

Offset	Annotation	Time
1D	ARM	01/20/2016 16:39:49
5DE540	resources.pri-1	01/20/2016 16:36:32
627384	x64	01/20/2016 16:39:34
C062EC	resources.pri -3	01/20/2016 16:37:32
C4F12E	x86	01/20/2016 16:39:15
122D7AB	resources.pri -5	01/20/2016 16:38:22

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:

1. Enable the developer mode. To do that, go to *Settings->Update & Security->For developers* and select *Developer mode* option.



2. Install the certificate which the application have been signed in the computer. To do that, open the certificate, click over the *Install Certificate* button and follow the wizard.

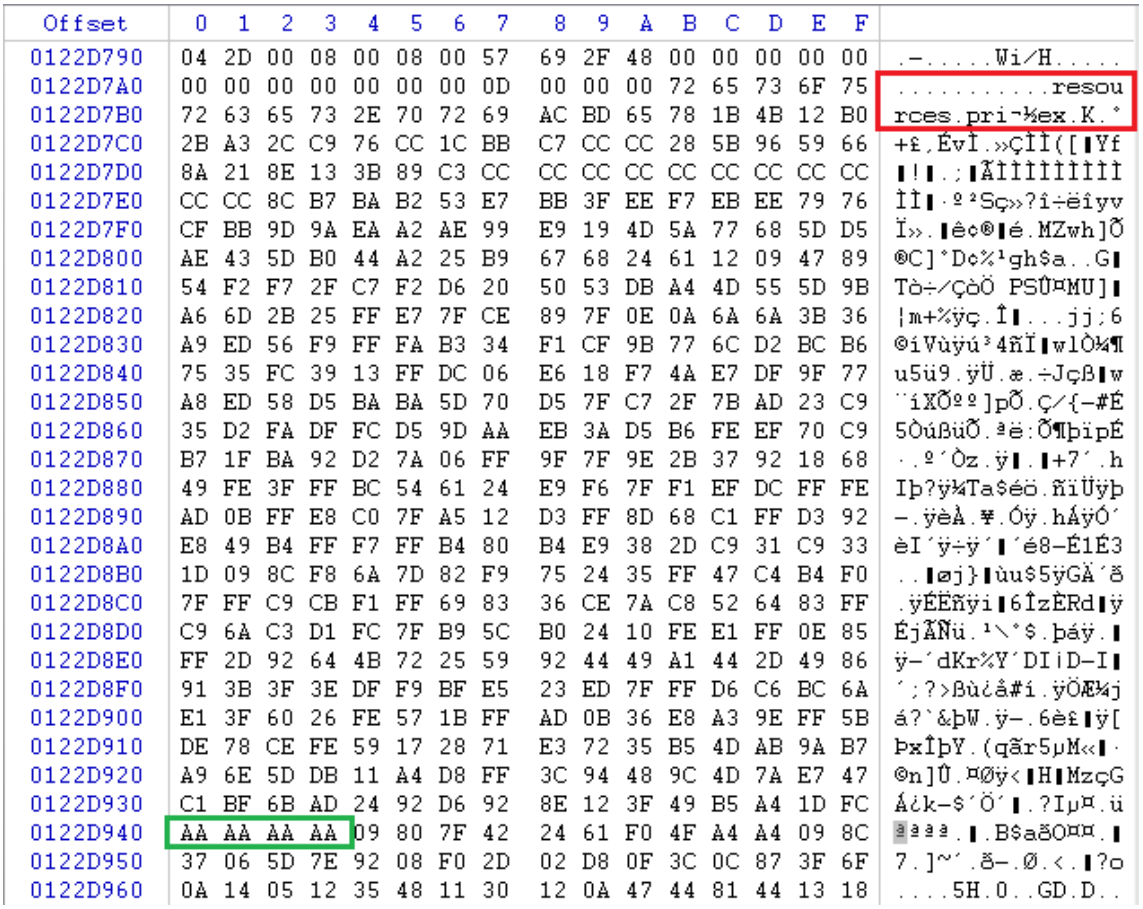
After the previous setup steps have been completed, the test procedure is described as follows:

1. Install the application in its version 1.0.0.0.
2. Create a copy of the application file in its version 1.1.0.0. This file is going to be modified.
3. Perform a modification over the copied file. This modification should be made either over a file which is going to be used during the update process or over a file which is not going to be used during the update process.

4. Attempt to update the application using the modified installer file. If the modification was made over a file which is going to be used, the application will not be updated. Otherwise, the application will be updated successfully.

The obtained results are described in the following bullets:

- **Test 3 - Modifying a file used during an update:** The target package is *AppInstallApp_1.0.0.0_x86Debug.appx*. This package contains the required files to update the application in 32-bits architecture platforms. So, after the modification and due to the tested platform is a Dell Optiplex 755 with Windows 10 x86 Enterprise Edition, the application should not be installed. Prior to the modification, the evaluator has identified the file offset belonging to the *resource.pri* for 32-bits installer. The file modification is performed at offset 0x0122D940 as the following image shows.

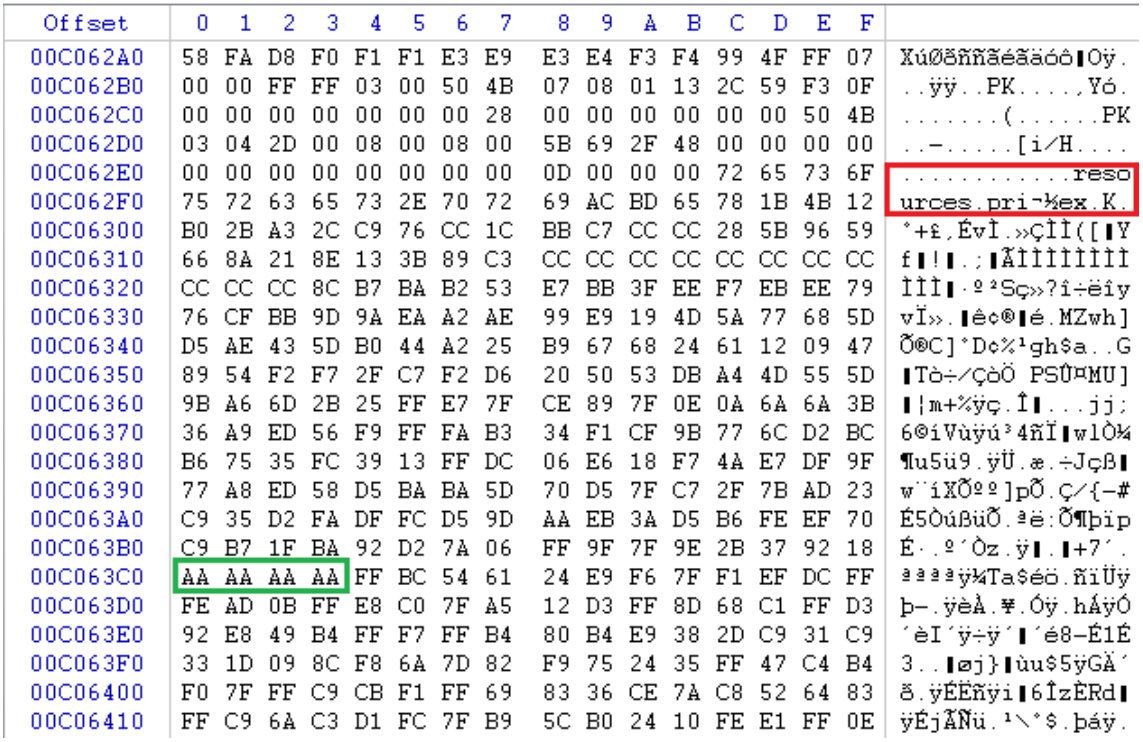


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0122D790	04	2D	00	08	00	08	00	57	69	2F	48	00	00	00	00	00	.-.....Wi/H.....
0122D7A0	00	00	00	00	00	00	00	0D	00	00	00	72	65	73	6F	75	...........resou
0122D7B0	72	63	65	73	2E	70	72	69	AC	BD	65	78	1B	4B	12	B0	rces.pri¬½ex.K.°
0122D7C0	2B	A3	2C	C9	76	CC	1C	BB	C7	CC	CC	28	5B	96	59	66	+£,Évì.»ÇÌÌ([▌Yf
0122D7D0	8A	21	8E	13	3B	89	C3	CC	CC	CC	CC	CC	CC	CC	CC	CC	▌!▌.;▌ÃÌÌÌÌÌÌÌÌÌ
0122D7E0	CC	CC	8C	B7	BA	B2	53	E7	BB	3F	EE	F7	EB	EE	79	76	ÌÌ▌·º²Sç»?î÷ëîyv
0122D7F0	CF	BB	9D	9A	EA	A2	AE	99	E9	19	4D	5A	77	68	5D	D5	Ï»..▌ê¢®▌é.MZwh]Õ
0122D800	AE	43	5D	B0	44	A2	25	B9	67	68	24	61	12	09	47	89	®C]°D¢%¹gh$a..G▌
0122D810	54	F2	F7	2F	C7	F2	D6	20	50	53	DB	A4	4D	55	5D	9B	Tò÷/ÇòÖ PSÛ¤MU]▌
0122D820	A6	6D	2B	25	FF	E7	7F	CE	89	7F	0E	0A	6A	6A	3B	36	¦m+%ÿç.Î▌...jj;6
0122D830	A9	ED	56	F9	FF	FA	B3	34	F1	CF	9B	77	6C	D2	BC	B6	©íVùÿú³4ñÏ▌wlÒ¼¶
0122D840	75	35	FC	39	13	FF	DC	06	E6	18	F7	4A	E7	DF	9F	77	u5ü9.ÿÜ.æ.÷Jçß▌w
0122D850	A8	ED	58	D5	BA	BA	5D	70	D5	7F	C7	2F	7B	AD	23	C9	¨íXÕºº]pÕ.Ç/{-#É
0122D860	35	D2	FA	DF	FC	D5	9D	AA	EB	3A	D5	B6	FE	EF	70	C9	5ÒúßüÕ.ªë:Õ¶þïpÉ
0122D870	B7	1F	BA	92	D2	7A	06	FF	9F	7F	9E	2B	37	92	18	68	·.º´Òz.ÿ▌.▌+7´.h
0122D880	49	FE	3F	FF	BC	54	61	24	E9	F6	7F	F1	EF	DC	FF	FE	Iþ?ÿ¼Ta$éö.ñïÜÿþ
0122D890	AD	0B	FF	E8	C0	7F	A5	12	D3	FF	8D	68	C1	FF	D3	92	-.ÿèÀ.¥.Óÿ.hÁÿÓ´
0122D8A0	E8	49	B4	FF	F7	FF	B4	80	B4	E9	38	2D	C9	31	C9	33	èI´ÿ÷ÿ´▌´é8-É1É3
0122D8B0	1D	09	8C	F8	6A	7D	82	F9	75	24	35	FF	47	C4	B4	F0	..▌øj}▌ùu$5ÿGÄ´ð
0122D8C0	7F	FF	C9	CB	F1	FF	69	83	36	CE	7A	C8	52	64	83	FF	.ÿÉËñÿi▌6ÎzÈRd▌ÿ
0122D8D0	C9	6A	C3	D1	FC	7F	B9	5C	B0	24	10	FE	E1	FF	0E	85	ÉjÃÑü.¹\°$.þáÿ.▌
0122D8E0	FF	2D	92	64	4B	72	25	59	92	44	49	A1	44	2D	49	86	ÿ-´dKr%Y´DI¡D-I▌
0122D8F0	91	3B	3F	3E	DF	F9	BF	E5	23	ED	7F	FF	D6	C6	BC	6A	´;?>ßù¿å#í.ÿÖÆ¼j
0122D900	E1	3F	60	26	FE	57	1B	FF	AD	0B	36	E8	A3	9E	FF	5B	á?`&þW.ÿ-.6è£▌ÿ[
0122D910	DE	78	CE	FE	59	17	28	71	E3	72	35	B5	4D	AB	9A	B7	Þxîþ Y.(qãr5µM«▌·
0122D920	A9	6E	5D	DB	11	A4	D8	FF	3C	94	48	9C	4D	7A	E7	47	©n]Û.¤Øÿ<▌H▌MzçG
0122D930	C1	BF	6B	AD	24	92	D6	92	8E	12	3F	49	B5	A4	1D	FC	Á¿k-$´Ö´▌.?Iµ¤.ü
0122D940	AA	AA	AA	AA	09	80	7F	42	24	61	F0	4F	A4	A4	09	8C	ªªªª.▌.B$aðO¤¤.▌
0122D950	37	06	5D	7E	92	08	F0	2D	02	D8	0F	3C	0C	87	3F	6F	7.]~´.ð-.Ø.<.▌?o
0122D960	0A	14	05	12	35	48	11	30	12	0A	47	44	81	44	13	18	....5H.0..GD.D..

Finally, the evaluator has attempted to update the modified application and it is not updated properly. Therefore the obtained result is the same as the expected one.

- **Test 4 - Modifying a file NOT used during an update:** The target package is *AppInstallApp_1.0.0.0_x64Debug.appx*. This package contains the required files to update the application in 64-bits architecture platforms. So, after the modification and due to the tested platform is a Dell Optiplex 755 with Windows 10 x86 Enterprise Edition, the

application should be installed properly. Prior to the modification, the evaluator has identified the file offset belonging to the *resource.pri* for 64-bits installer. The file modification is performed at offset 0x00C063C0 as the following image shows.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00C062A0	58	FA	D8	F0	F1	F1	E3	E9	E3	E4	F3	F4	99	4F	FF	07	XúØðññãéãäóô▮Oÿ.
00C062B0	00	00	FF	FF	03	00	50	4B	07	08	01	13	2C	59	F3	0F	..ÿÿ..PK.....,Yó.
00C062C0	00	00	00	00	00	00	00	28	00	00	00	00	00	00	50	4B	.......(......PK
00C062D0	03	04	2D	00	08	00	08	00	5B	69	2F	48	00	00	00	00	..-.....[i/H....
00C062E0	00	00	00	00	00	00	00	00	0D	00	00	00	72	65	73	6F	............reso
00C062F0	75	72	63	65	73	2E	70	72	69	AC	BD	65	78	1B	4B	12	urces.pri¬½ex.K.
00C06300	B0	2B	A3	2C	C9	76	CC	1C	BB	C7	CC	CC	28	5B	96	59	°+£,ÉvÌ.»ÇÌÌ([▮Y
00C06310	66	8A	21	8E	13	3B	89	C3	CC	CC	CC	CC	CC	CC	CC	CC	f▮!▮.;▮ÃÌÌÌÌÌÌÌÌ
00C06320	CC	CC	CC	8C	B7	BA	B2	53	E7	BB	3F	EE	F7	EB	EE	79	ÌÌÌ▮·º²Sç»?î÷ëîy
00C06330	76	CF	BB	9D	9A	EA	A2	AE	99	E9	19	4D	5A	77	68	5D	vÏ».▮ê¢®▮é.MZwh]
00C06340	D5	AE	43	5D	B0	44	A2	25	B9	67	68	24	61	12	09	47	Õ®C]°D¢%¹gh$a..G
00C06350	89	54	F2	F7	2F	C7	F2	D6	20	50	53	DB	A4	4D	55	5D	▮Tò÷/ÇòÖ PSÛ¤MU]
00C06360	9B	A6	6D	2B	25	FF	E7	7F	CE	89	7F	0E	0A	6A	6A	3B	▮¦m+%ÿç.Î▮...jj;
00C06370	36	A9	ED	56	F9	FF	FA	B3	34	F1	CF	9B	77	6C	D2	BC	6©íVùÿú³4ñÏ▮wlÒ¼
00C06380	B6	75	35	FC	39	13	FF	DC	06	E6	18	F7	4A	E7	DF	9F	¶u5ü9.ÿÜ.æ.÷Jçß▮
00C06390	77	A8	ED	58	D5	BA	BA	5D	70	D5	7F	C7	2F	7B	AD	23	w¨íXÕºº]pÕ.Ç/{-#
00C063A0	C9	35	D2	FA	DF	FC	D5	9D	AA	EB	3A	D5	B6	FE	EF	70	É5ÒúßüÕ.ªë:Õ¶þïp
00C063B0	C9	B7	1F	BA	92	D2	7A	06	FF	9F	7F	9E	2B	37	92	18	É·.º'Òz.ÿ▮.▮+7'.
00C063C0	AA	AA	AA	AA	FF	BC	54	61	24	E9	F6	7F	F1	EF	DC	FF	ªªªªÿ¼Ta$éö.ñïÜÿ
00C063D0	FE	AD	0B	FF	E8	C0	7F	A5	12	D3	FF	8D	68	C1	FF	D3	þ-.ÿèÀ.¥.Óÿ.hÁÿÓ
00C063E0	92	E8	49	B4	FF	F7	FF	B4	80	B4	E9	38	2D	C9	31	C9	'èI´ÿ÷ÿ´▮´é8-É1É
00C063F0	33	1D	09	8C	F8	6A	7D	82	F9	75	24	35	FF	47	C4	B4	3..▮øj}▮ùu$5ÿGÄ´
00C06400	F0	7F	FF	C9	CB	F1	FF	69	83	36	CE	7A	C8	52	64	83	ð.ÿÉËñÿi▮6ÎzÈRd▮
00C06410	FF	C9	6A	C3	D1	FC	7F	B9	5C	B0	24	10	FE	E1	FF	0E	ÿÉjÃÑü.¹\°$.þáÿ.

Finally, the evaluator has attempted to update the modified application and it is updated properly due to the modified file is not used during the update process. Therefore the obtained result is the same as the expected one.

37.3.1.4. Verdict

Analyzing the obtained results, the evaluator considers that the Windows Application file is not installed or updated properly, when the modified file is used during the installation. In case of the modified file is not used during the installation, their integrity is not validated and therefore the update is applied properly.

Therefore, the test results obtained during this test activity demonstrate that the Test 1 requirements established in the assurance activity section are properly fulfilled. Due to this, the **PASS** verdict is assigned to Test 1.

37.3.2. Test 2

37.3.2.1. Setup

The applicable setup for this test is the same as the one defined for Test 1

37.3.2.2. Procedure

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:

1. Open a PowerShell terminal as administrator and type the following command:

   Add-AppxPackage RemoteDesktopApp.appxbundle -DependencyPath Microsoft.WinJS.2.0.appx

2. Observe that the application shall be installed successfully.

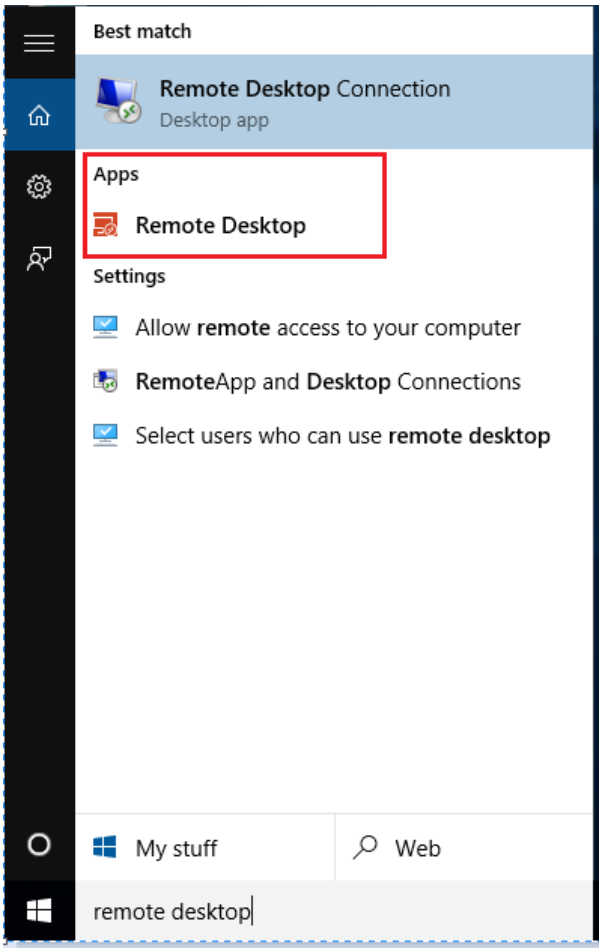
37.3.2.3. Results

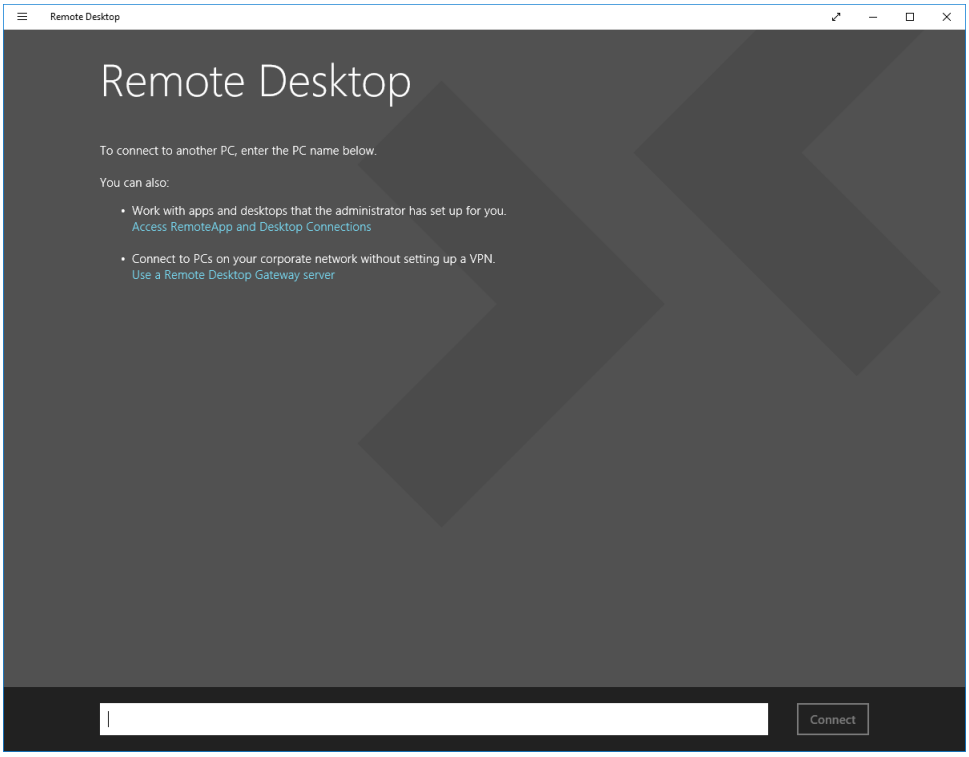
The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition.
- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition.

The evaluator has attempted to install an application file which has a valid digital signature belonging to the vendor. The obtained results have been the same in both tested platforms; the application has been installed successfully.

The following screenshots, which are taken from one of the tested platform, show this fact:





37.3.2.4. Verdict

The evaluator has attempted to install an application file which has a valid digital signature belonging to the vendor, and it has been installed successfully.

Due to this, the evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of **Test 2** requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to **Test 2**.

37.4. Final Verdict

Due to both documentation review activity and test activities 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 FPT_TUD_EXT.2.2.

38. FTA_TAB.1.1

38.1. Assurance activity

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.

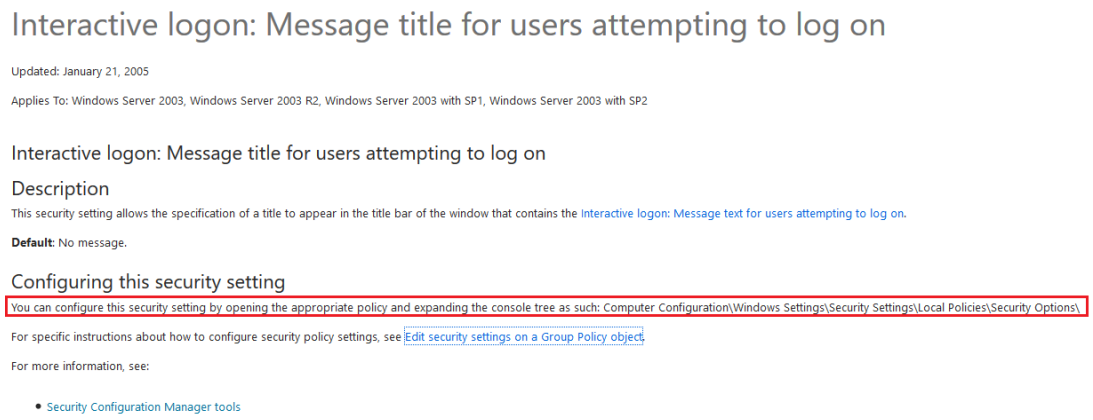
38.2. Documentation review activity

38.2.1. Findings

The evaluator has reviewed the operational guidance in order to configure the OS to display the advisory warning message. This document includes in its section **19.Managing Logon Banner** two pointers to the vendor support webpage.

The evaluator has navigated through these links, and has followed the described steps in order to configure each item (Message title for users attempting to log on and Message text for users attempting to log on).

The following screenshot show the information provided in one of this link:



Interactive logon: Message title for users attempting to log on

Updated: January 21, 2005

Applies To: Windows Server 2003, Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 with SP2

Interactive logon: Message title for users attempting to log on

Description

This security setting allows the specification of a title to appear in the title bar of the window that contains the Interactive logon: Message text for users attempting to log on.

Default: No message.

Configuring this security setting

You can configure this security setting by opening the appropriate policy and expanding the console tree as such: Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options\

For specific instructions about how to configure security policy settings, see Edit security settings on a Group Policy object.

For more information, see:

- Security Configuration Manager tools

38.2.2. 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.

38.3. Test Activity

38.3.1. Test 1

38.3.1.1. Setup

Before the test execution, the following setup conditions must be fulfilled to ensure that there will not be errors during the test execution:

- The PowerShell execution policy shall be configured to allow the execution of PowerShell scripts. To do this, type the following command in a PowerShell terminal: "*Set-ExecutionPolicy Unrestricted*".

38.3.1.2. Procedure

In order to perform this test, the evaluator has followed the following steps:

1. Run as administrator the script *FTA_TAB.1.ps1*, and select the "*Enabled*" option when the dialog is shown. The computer will be restarted. This script allow to modify the Windows Registry keys related to the advisory warning message. These keys are the following:

    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticecaption.
    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticetext.

    The script source code is as follows:

```
$title = "Show warning messages"
$message = "This script enables/disables the warning message on login screen. NOTE: Your computer will be restarted"

$enabled = New-Object System.Management.Automation.Host.ChoiceDescription "&Enabled", "Enable warning message."

$disabled = New-Object System.Management.Automation.Host.ChoiceDescription "&Disabled", "Disable warning message."

$options = [System.Management.Automation.Host.ChoiceDescription[]]($enabled, $disabled)

$result = $host.ui.PromptForChoice($title, $message, $options, 0)

switch ($result)
    {
        0 {
            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"
            Restart-Computer -Force
        }
        1 {
            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 ""
            Restart-Computer -Force
        }
}
```

2. To execute the script, type the following command in a PowerShell terminal: "*.\FTA_TAB.1.ps1*"

38.3.1.3. Results

The evaluator has performed this test on the following evaluated platforms:

- Dell Optiplex 755 with Windows 10 x86 Enterprise Edition

- Dell Optiplex 755 with Windows Server 2012 R2 Datacenter Edition
- Surface 3 with Windows 10 x64 Enterprise Edition
- Surface Book with Windows 10 x64 Enterprise Edition
- Windows Server 2012 R2 Hyper-V with Windows 10 x64 Enterprise Edition

The evaluator has obtained the same results for all tested platforms. After rebooting the machine and before the login screen will be displayed, the following message is shown:

As it can be observed, this screen show the title and the message configured during the script execution

38.3.1.4. Verdict

The evaluator considers that, the tests results obtained during this test activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the test activity.

38.4. Final Verdict

Due to both documentation review activity and test activities 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 FPT_TAB_EXT.1.1.

39. FTP_TRP.1

39.1. Assurance activity

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.*

39.2. Documentation review activity

39.2.1. Findings

In the section *"6.8 of the Windows 10 Security Target"* document describes how the communications are protected using *"TLS"* and "*HTTPS"*. In addition, a pointer to the section *"6.2.2 of the Windows 10 Security Target"* document is included. This section details the TLS cipher suites supported by the TOE.

The section *"6.8 of the Windows 10 Security Target"* describes the remote access methods, this methods are:

- Remote Desktop Services
- Connect to another computer using Remote Desktop Connection.

The operational Guidance describes the steps to establish a remote connection in the section *"17 Managing Remote Administration"* of the *" Windows 10 and Server 2012 R2 GP OS Operational Guidance"* document.

39.2.2. Verdict

The evaluator has found in the TSS the method used in the remote access and the protocol used to protect the connection.

In addition, the evaluator has checked that the information included in the *"Windows 10 and Server 2012 R2 OS Operational Guidance"* describes the remote access methods listed in the Security Target document.

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.

39.3. Test Activity

39.3.1. Test 1 and Test 3

39.3.1.1. Setup

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2 x64)
- Client Machine (Platforms listed in the ST)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.200
- Client Machine, IP = 192.168.1.120
- Client Machine2, IP = 192.168.1.121

The following steps shall be perform in the Server Machine:

- Click "Server manager", click "Manage Add Roles and Features"
- Click "Next".
- Select "Role-based or feature-based installation" and press "Next".
- Select a server from a pool (192.168.1.200) and click "Next".
- In the "Server Roles" expand "Remote Desktop Services and click "Next".

- Click "Next".
- Click "Next".
- Select "Remote Desktop Session Host", press "Add Features" and click "Next".
- Press "Install".

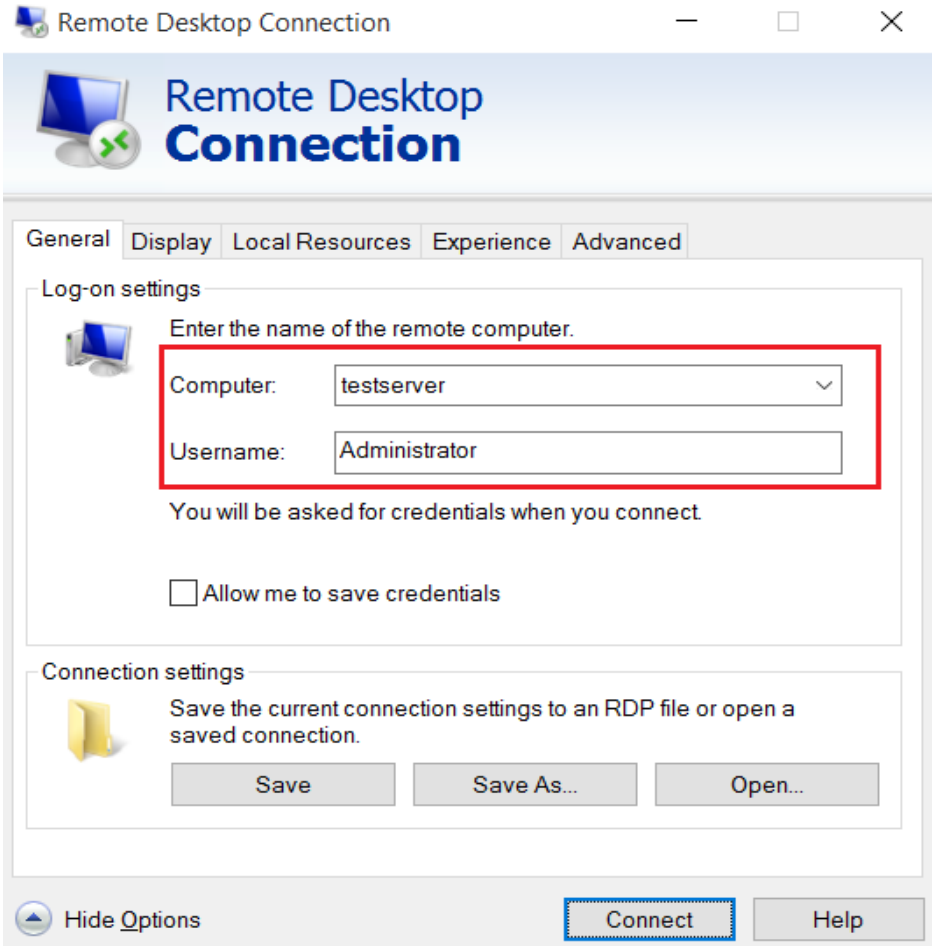
In the Server Machine shall be installed the *"Wireshark"* application.

The Client Machines shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

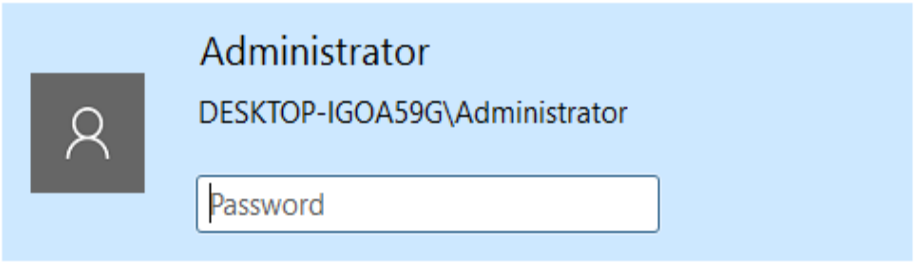
39.3.1.2. Procedure

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

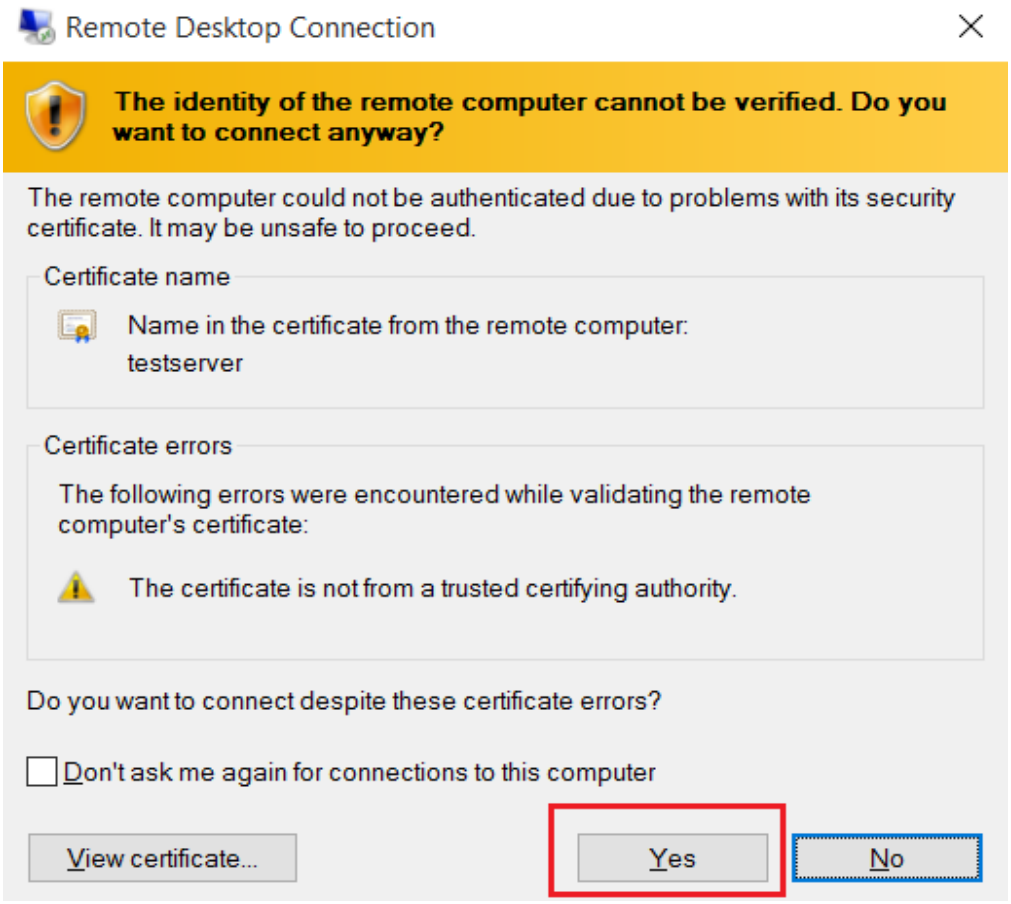
1. In the Server and Client Machines, open a *"Wireshark"* application to verify that remote connection is performed correctly.
2. In the Client Machine type in the *"Search Text Box Remote Desktop"* and open the *"Remote Desktop Connection"* application.
3. In the computer name type the server name *"testserver"*.
4. In the username type *"administrator"*.



5. Click *"Connect"*.
6. Type the password in the new window.

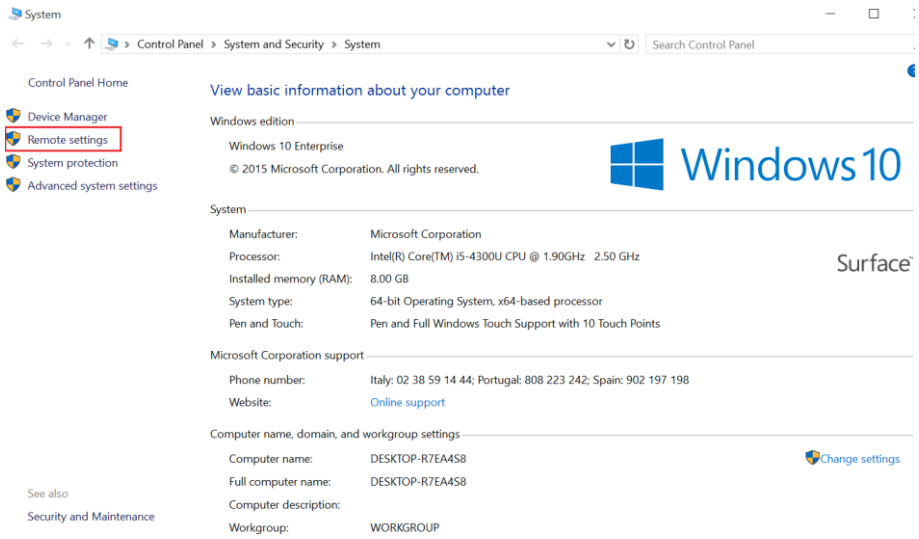


7. When the server sends the certificate, click *"Yes"*.

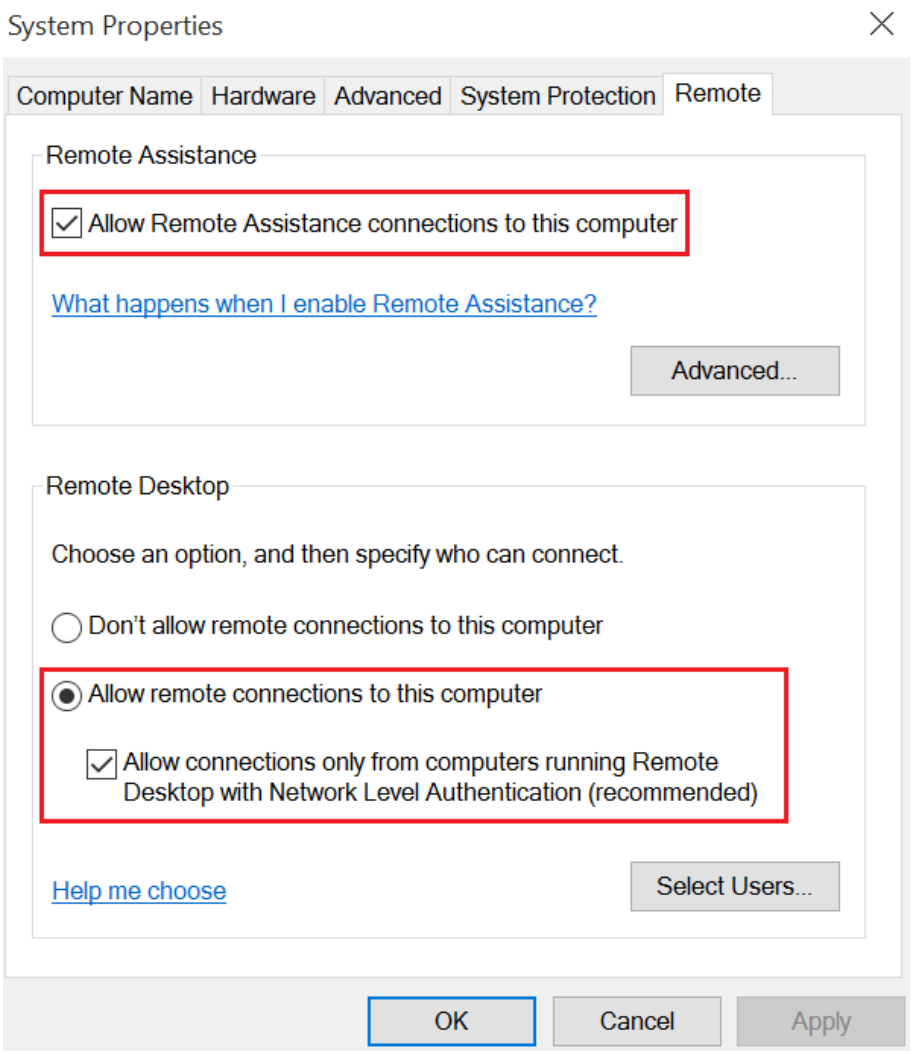


8. In order to connect two Client Machines, the following steps shall be performed:

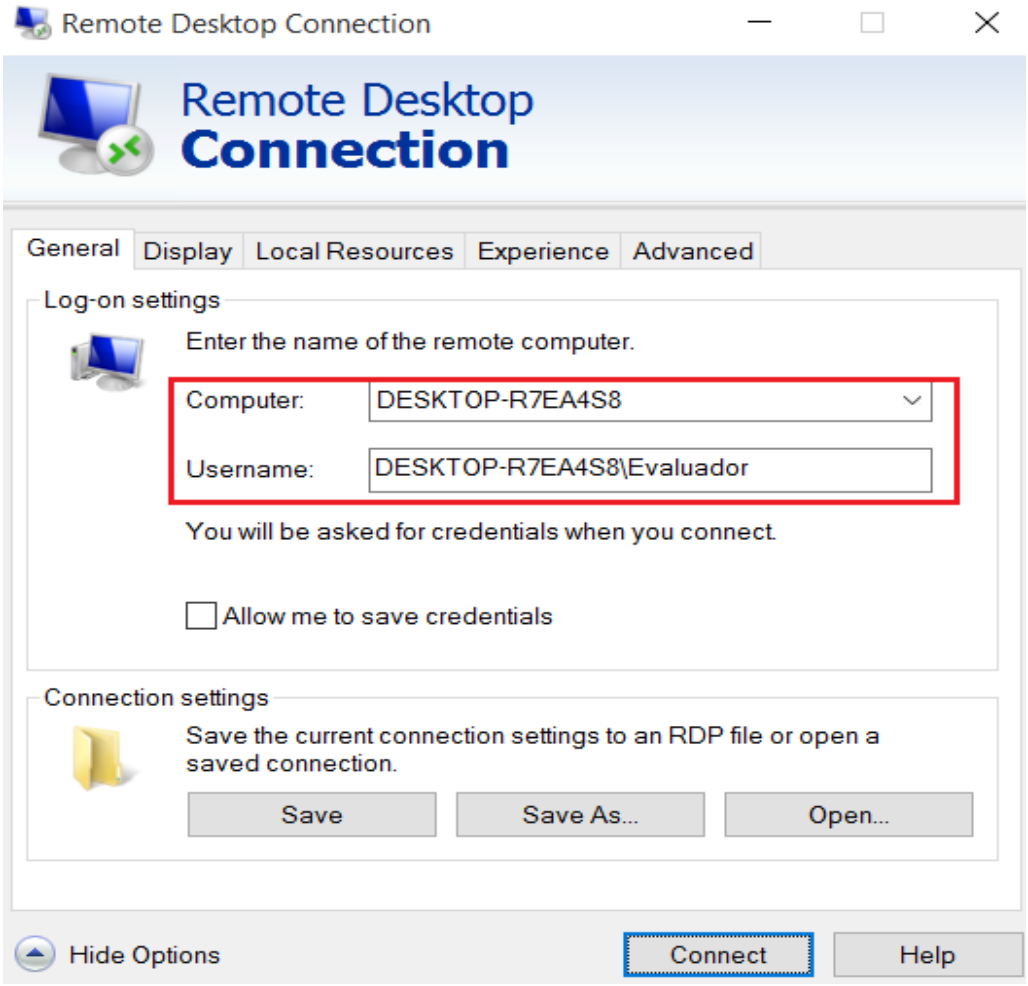
- Right-click on "Start" and select System.
- Click "Remote settings"



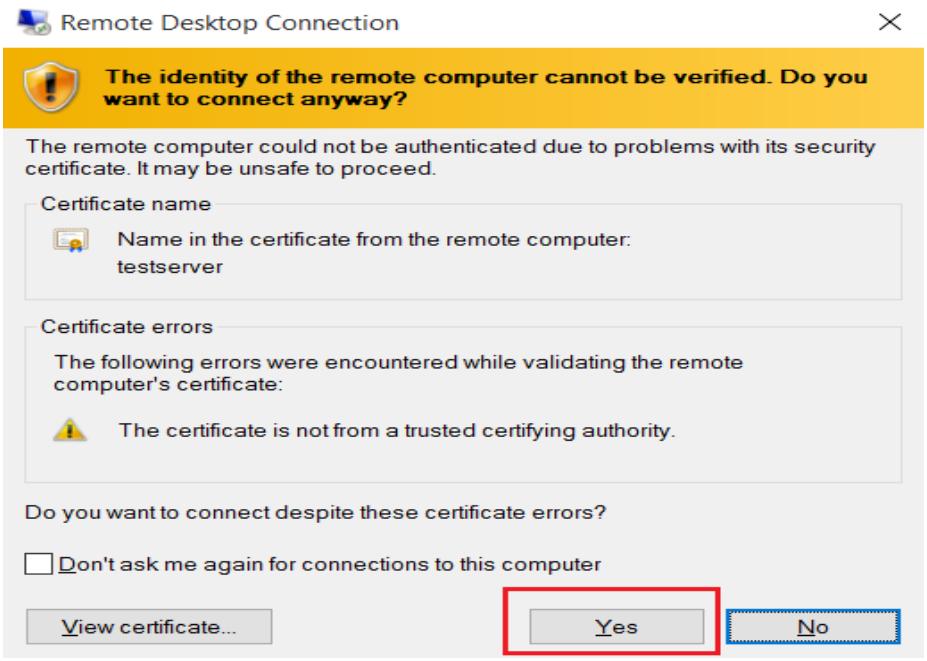
- Select the tag "Remote" and permit the remote connections



9. In the Client Machines, open a *"Wireshark"* application to verify that remote connection is performed correctly.
10. In once Client Machine type in the *"Search Text Box Remote Desktop"* and open the *"Remote Desktop Connection"* application.
11. In the computer name type the server name *"DESKTOP-R7EA4S8"*.
12. In the username type *"evaluador"*.



13. Click *"Connect"*.
14. Type the password in the new window.
15. When the server sends the certificate, click *"Yes"*.

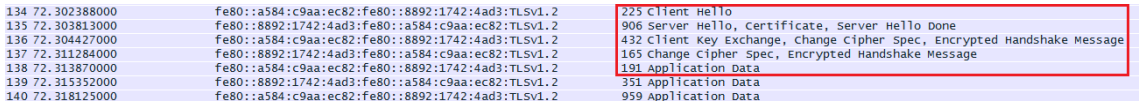


39.3.1.3. Results

The test has been performed in the following platforms:

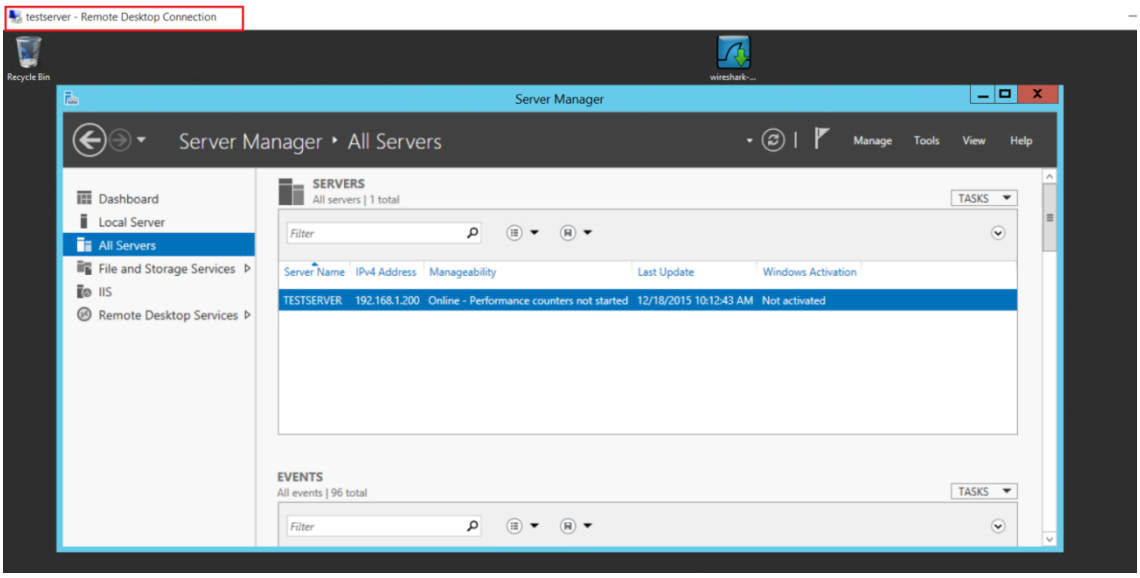
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured in the *"Wireshark",* can be appreciated the handshaking process during the remote connection, as it can be appreciated in the following picture.



Once the connection is established, the user data are sent in the *"Application Data"* messages.

When the secure channel is established, the user can manage the Server Machine, as it can be appreciate in the next picture.



Therefore the remote connection is established correctly using a secure channel.

The results obtained when the connection is established between two Windows 10 machines are the same as exposed above.

39.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests results obtained during the **Test 1** and **Test 3** activity demonstrate the fulfillment of the requirements established in the assurance activity section. Therefore, the **PASS** verdict is assigned to the **Test 1** and **Test 3** activity.

39.3.2. Test 2

39.3.2.1. Setup

The machines shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*.

39.3.2.2. Procedure

The steps listed in the section *"17 Managing Remote Administration"* of the *" Windows 10 and Server 2012 R2 GP OS Operational Guidance"* shall be followed to configure a remote connection.

39.3.2.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.

- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

When the remote connection is configured according to the steps listed in the " *Windows 10 and Server 2012 R2 GP OS Operational Guidance*" document, the connection is established using a trusted channel with *"TLS 1.2"* and *"X509"* certificates.

39.3.2.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

39.3.3. Test 4

39.3.3.1. Setup

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2 x64)
- Client Machine (Platforms listed in the ST)
- MITM Machine (Kali Linux)

These three machines are in the same network with the following configuration:

- Server Machine, IP = 192.168.1.120
- Client Machine, IP = 192.168.1.121
- MITM Machine, IP = 192.168.1.100

The following steps shall be perform in the Server Machine:

- Click "Server manager", click "Manage Add Roles and Features"
- Click "Next".
- Select "Role-based or feature-based installation" and press "Next".
- Select a server from a pool (192.168.1.120) and click "Next".
- In the "Server Roles" expand "Remote Desktop Services and click "Next".

- Click "Next".
- Click "Next".
- Select "Remote Desktop Session Host", press "Add Features" and click "Next".
- Press "Install".

In the Server Machine shall be installed the *"Wireshark"* application.

The Client Machines shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

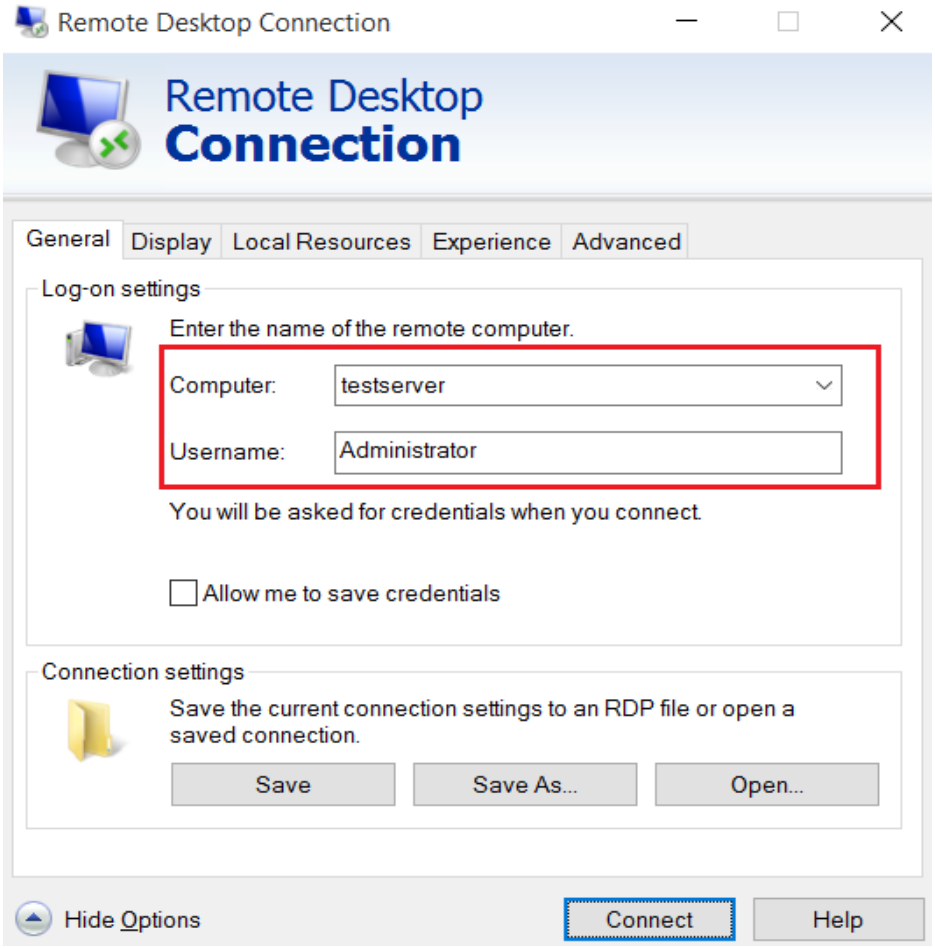
The MITM machine shall be installed *"python-dpkt_1.6+svn54-1_all.deb"* packet.

The *"SSL_Proxy"* tool is used to modify the packets between the Client Machine and Server Machine. This tool shall be copied in the Desktop of the MITM Machine.

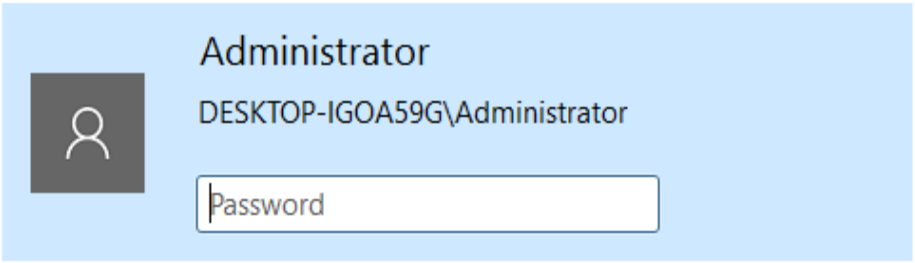
39.3.3.2. Procedure

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

1. In the Server and Client Machines, open a *"Wireshark"* application to verify that remote connection is performed correctly.
2. In the Client Machine type in the *"Search Text Box Remote Desktop"* and open the *"Remote Desktop Connection"* application.
3. In the MITM Machine open a terminal and type the followings commands:
    - *"cd Desktop/SSL_Proxy"*
    - *"chmod 777 run_mitm"*
    - *"./run_mitm"*
4. In the computer name type the server name *"testserver"*.
5. In the username type *"administrator".*



6. Click *"Connect"*.
7. Type the password in the new window.



8. In the Client Machines, open a *"Wireshark"* application to verify that remote connection is not performed correctly.

39.3.3.3. Results

The test has been performed in the following platforms:

- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.

- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can be appreciated the modification performed in the certificate by the MITM Machine. The function used to modify the certificate can be appreciated in the next picture.

```
def send_garbled_packet(self, data, offset):
    packet = data
    newPacket = data
    indexPacket = offset
    global changeCipherSpec

    if changeCipherSpec == 0:
        if packet[0] == '\x14':
            changeCipherSpec = 1

    else:
        if packet[0] == '\x17':
            indexPacket += 1 #application data
            indexPacket += 2 #tls version
            indexPacket += 2 #len
            indexPacket += 1 #second byte encrypted application data
            print "Application Data before modification"
            print packet.encode("hex")
            newPacket = packet[:indexPacket] + '\xAA' + packet[indexPacket + 1:]
            changeCipherSpec = 0

    return newPacket
```

When the server send an "Application Data" message, the MITM Machine modifies the content of this packet. In the next picture can be appreciated the packet before the modification.

```
Application Data before modification
17030301107207dc54c1754b46b5fe60f3730bf030dd9c627b891d7e40d93f0ed218c98564e815164475fabcb776f4af18a2f666063c7598b4581a192de2d1cd9aa3ae8e0639
a5606da3123c10b284df320145db1452fec1073f76e92a416486ed8c43c994a0bf0448264a53d8f90dd7001e96fb39b99845b78d2bc88e164f485d3af8878119cdbc648da5bb
28eeaa66f5f15b775138137d73f5bc599f045249041c414fa26ead14b898ff43825d0133edb779ac6082c2045848db3de5e7ac28831bedd5b9b443335bf0c7f943ad761f343b
b6654844ce06cf3349dfab10ac2fac703576a2a29c3b7e49546b646372031c40479b39428bc9a214febac727bcdd0cd02a327582e14155ec091b56b4a85c26daaefc34
```

After the modification, the packets looks as follow.

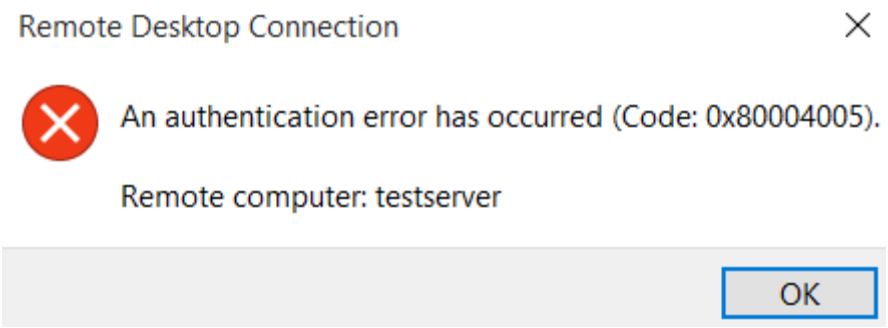
```
TLSv1.2 Record Layer: Application Data Protocol: Application Data
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 272
    Encrypted Application Data: 72aadc54c1754b46b5fe60f3730bf030dd9c627b8
```

```
0000  94 57 a5 b9 49 a9 00 0c  29 07 6c 9b 08 00 45 00   .W..I... ).l...E.
0010  01 3d 63 80 40 00 40 06  51 f9 c0 a8 01 78 c0 a8   .=c.@.@. Q....x..
0020  01 79 0d 3d 08 b5 ff 9c  b1 5e 94 f7 6c 4c 50 18   .y.=.... .^..lLP.
0030  00 1f 30 03 00 00 17 03  03 01 10 72 aa dc 54 c1   ..0..... ...r..T.
0040  75 4b 46 b5 fe 60 f3 73  0b f0 30 dd 9c 62 7b 89   uKF..`.s ..0..b{.
0050  1d 7e 40 d9 3f 0e d2 18  c9 85 64 e8 15 16 44 75   .~@.?... ..d...Du
0060  fa bc b7 76 f4 af 18 a2  f6 66 06 3c 75 98 b4 58   ...v.... .f.<u..X
0070  1a 19 2d e2 d1 cd 9a a3  ae 8e 06 39 a5 60 6d a3   ..-..... ...9.`m.
0080  12 3c 10 b2 84 df 32 01  45 db 14 52 fe c1 07 3f   .<....2. E..R...?
0090  76 e9 2a 41 64 86 ed 8c  43 c9 94 a0 bf 04 48 26   v.*Ad... C.....H&
00a0  4a 53 d8 f9 0d d7 00 1e  96 fb 39 b9 98 45 b7 8d   JS...... ..9..E..
00b0  2b c8 8e 16 4f 48 5d 3a  f8 87 81 19 cd bc 64 8d   +...OH]: ......d.
00c0  a5 bb 28 ee aa 66 f5 f1  5b 77 51 38 13 7d 73 f5   ..(..f.. [wQ8.}s.
00d0  bc 59 9f 04 52 49 04 1c  41 4f a2 6e ad 14 b8 98   .Y..RI.. AO.n....
00e0  ff 43 82 5d 01 33 ed b7  79 ac 60 82 c2 04 58 48   .C.].3.. y.`...XH
00f0  db 3d e5 e7 ac 28 83 1b  ed d5 b9 b4 43 33 5b f0   .=...(.. ....C3[.
0100  c7 f9 43 ad 76 1f 34 3b  b6 65 48 44 ce 06 cf 33   ..C.v.4; .eHD...3
0110  49 df ab 10 ac 2f ac 70  35 76 a2 a2 9c 3b 7e 49   I..../.p 5v...;~I
0120  54 6b 64 63 72 03 1c 40  47 9b 39 42 8b c9 a2 14   Tkdcr..@ G.9B....
0130  fe ba c7 27 bc dd 0c d0  2a 32 75 82 e1 41 55 ec   ...'.... *2u..AU.
0140  09 1b 56 b4 a8 5c 26 da  ae fc 34                 ..V..\&. ..4
```

This modification implies that the Client Machine send a "RST, ACK" packet, closing the connection established. This behavior can be appreciated in the following picture.

23	11.736971000	192.168.1.121	192.168.1.120	TLSv1.2	205 Client Hello
24	11.741943000	192.168.1.120	192.168.1.121	TLSv1.2	886 Server Hello, Certificate, Server Hello Done
25	11.744429000	192.168.1.121	192.168.1.120	TLSv1.2	412 Client Key Exchange, Change Cipher Spec, Encrypted
26	11.758958000	192.168.1.120	192.168.1.121	TLSv1.2	145 Change Cipher Spec, Encrypted Handshake Message
27	11.765746000	192.168.1.121	192.168.1.120	TLSv1.2	171 Application Data
28	11.770904000	192.168.1.120	192.168.1.121	TLSv1.2	331 Application Data
29	11.783738000	192.168.1.121	192.168.1.120	TCP	54 2229-3389 [RST, ACK] Seq=674 Ack=1220 Win=0 Len=0

In addition, the "Remote Desktop" application shows the following error message.



39.3.3.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

39.4. Final Verdict

Due to all test activities 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 this FTP_TRP_EXT.1.

40. FIA_ITC_EXT.1.1

40.1. Assurance activity

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.

40.2. Documentation review activity

40.2.1. Findings

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

40.2.2. Verdict

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

40.3. Test Activity

40.3.1. Test 1

40.3.1.1. Setup

The scenario to perform the assurance activities according to the Protection Profile is composed by the following elements:

- Server Machine (Windows Server 2012 R2)
- Client Machine (Windows 10 x64 Enterprise Edition)
- Server Machine2 (Windows Pro x86)

These three machines are in the same network with the following configuration:.

- Server Machine, IP = 10.10.1.10
- Client Machine, IP = 10.10.1.13, IP = 192.168.1.120
- Server Machine2, IP = 192.168.1.102

In the Client Machine must be installed the applications *"Microsoft Edge"* and *"Wireshark"*. The *"Microsoft Edge"* application allow connection to the server machine.

The Client Machine shall have enabled the secure configuration according to the section *"1.2 Configuration"* of the *"Windows 10 and Server 2012 R2 GP OS Operational Guidance"*, in addition the *"Wireshark"* application shall be installed.

In the Server Machine shall be installed the applications *"Cerberus FTP Server enterprise"* and *"Wireshark"*.

This certificate is auto-created by the Active Directory Certificate Services when him is configured with the purposes:

- Client Authentication.
- Server Authentication.

The Server Machine shall have enabled the following roles: Active Directory Domain Services and Active Directory Certificate Services. In first place, the Active Directory Domain Services shall be installed following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select Add roles and Features.
3. Select *"Role-based or features-based"* installation from the *"Installation Type"* screen, and click next.
4. The current server is selected by default. Click next.
5. From the *"Server Roles"* screen check a mark in the box *"Active Directory Domain Services"*. An additional pop-up screen must appear explain all the features required to install the Domain Services. Click *"Add features"*. A new screen is shown and click next.
6. On *"Select features"*, Click Next.
7. Review the information on the Active Directory Domain Services tab and click Next.
8. Finally, click Install.

Once the Active Directory Domain Services role is installed, can be configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select the Notification Icon from the top of the Server Manager. A subsection appears and then click *"Promote this server to a domain controller"*.
3. From the Deployment Configuration tab select *"Add a new forest"*. Insert your root domain name into the Root Domain name field (For example, winserver.org). And, click Next.
4. Select a Forest and Domain functional level. In this case, the both must be *"Windows Server 2012"*. Domain Name System(DNS) is select by default. Write a password in the DSRM password field and then click Next.
5. A warning pop-up screen appear about DNS options. Click Next.
6. Confirm NetBIOS name (For example, WINSERVER). And click Next.

7. The location of SYSVOL, database folders and log files are selected by default. Click Next.
8. In Review Options Screen click Next.
9. The system will check to ensure all prerequisites are installed on the system for several minutes. If the system passes these check, click Install.

A LDAP user shall be created in the Active Directory Domain Services, following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select *"AD DS"* from the left panel.
3. Right-Click on the server. And select "*Active Directory Administrative Server*".
4. Select the section *"<Domain Name>(local)"* (Domain Name is selected when the Active Directory is installed.)
5. Double-click in Users folder.
6. In the right-panel click New on the Users Section. And then Click User.
7. On the Create User screen, fill the fields: Full name(Alan James), User UPN logon *"ajames"*. User *"SamAccountName"* is completed by default. Introduce a validate password. Optionally, fill the fields: First name and Last name. Click OK.

The Active Directory Certificate Services shall be installed and configured following the next steps:

1. Open the Server Manager from the task bar.
2. From Server Manager Dashboard select Add roles and Features.
3. Select *"Role-based or features-based installation"* from the *"Installation Type"* screen, and click Next.
4. Click in the radio button *"Select a server from the server pool"*. The current server is selected by default. Click next.
5. From the *"Server Roles"* screen check a mark in the box *"Active Directory Certificate Services"*. An additional pop-up screen must appear explain all the features required to install the Domain Services. Click *"Add features"*. A new screen is shown and click next.
6. On *"Select features"*, Click Next.
7. Review the information on the Active Directory Certificate Services tab and click Next.
8. From the *"Server role services"* screen check a mark in the box *"Certification Authority"*. Click Next.
9. Finally, click Install.

10. When the installation is finished, click *"Configure Active Directory Certificate Services on the destination server"* type in blue color.
11. On Credentials, write the name of the administrator in Credentials field. Click Next.
12. From the *"Role services"* screen check a mark in the box *"Certification Authority"*. Click Next.
13. On the Setup Type screen, select Enterprise CA. Click Next.
14. On the CA Type screen, select RootCA. Click Next.
15. On Private Key screen, select create a new private key. Click Next.
16. On Cryptography for CA, select SHA256. Others attributes is selected by default. Click Next.
17. On CA Name, all fields are selected by default. Click next.
18. On validity Period select 25 years. Click Next.
19. On CA Database, the database locations are selected by default. Click Next.
20. Review the information on the Confirmation tab and click Configure.
21. On Results page, click Close.

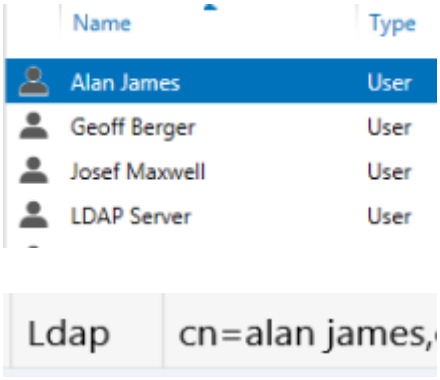
Add the certificates used for the test in the Client Machine following the next steps:

1. Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
2. At the command prompt, type *"mmc"* and press *"ENTER".*
3. On the *"File"* menu, click *"Add/Remove Snap-in".*
4. In the Add standalone Snap-in dialog box, select *"Certificates"*.
5. Press *"Add".*
6. In the Certificates Snap-in dialog box, select *"Computer Account"* and click next.
7. Select Local Computer and Press *"Finish".*
8. *Press "OK".*
9. Expand the Certificates section and select *the "Personal" certification store.*
10. On the Action menu, point to All Tasks, and then click Request New Certificate to start the Certificate Enrollment wizard Click Next.
11. On "Select Certificate Enrollment Policy" screen, Active Directory Enrollment Policy is selected by default. Click Next.
12. On Request Certificates, check a mark in the box "Domain Controller". Click Enroll.
13. Finally, click finish.

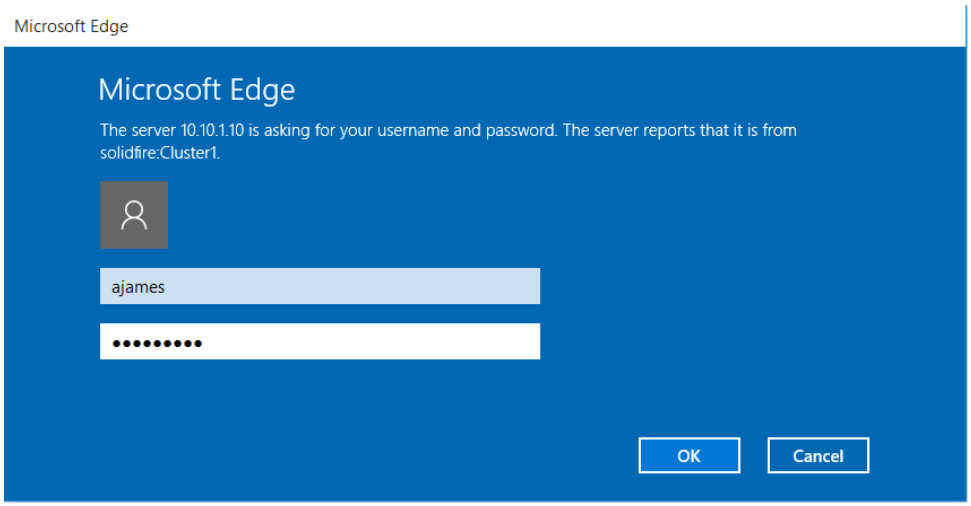
40.3.1.2. Procedure

The following steps shall be performed in order to complete the first part of this test assurance activity:

1. In the server machine, open the Server Manager from the task bar.
2. From Server Manager Dashboard select *"AD DS"* from the left panel.
3. Right-Click on the server. And select *"Active Directory Administrative Server".*
4. Select the section *"<Domain Name>(local)"* (Domain Name is selected when the Active Directory is installed.)
5. Double-click in Users folder.
6. We can see the user "Alan James" is created properly.



7. In the Client Machine, open a *"Wireshark"* application to verify that the handshake operation is performed correctly.
8. Open the "Microsoft Edge" application and type in the explorer field the Server Machine IP -> 10.10.1.10:80.
9. A log in screen appear.

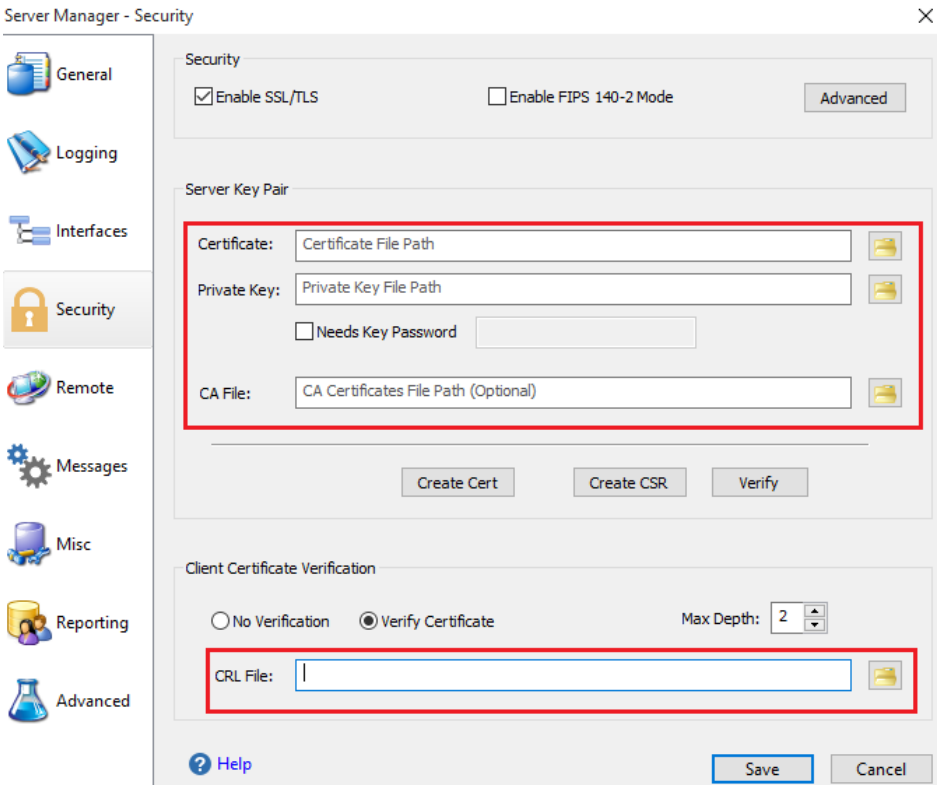


10. Type the user LDAP name and his user LDAP password.
11. Click OK.
12. The LDAP credentials user is verified in the LDAP authentication server.
13. If the credentials are correct, the user can access to the application.

The following steps shall be performed in order to complete the second part of this test assurance activity:

1. Add the certificates used for the test in the client machine following the next steps:
    - Click *"Start"*, click *"Run"*, type *"mmc"* and then click *"OK"*.
    - At the command prompt, type *"mmc"* and press *"ENTER"*.
    - On the *"File"* menu, click *"Add/Remove Snap-in"*.
    - In the Add standalone Snap-in dialog box, select *"Certificates"*.
    - Press *"Add"*.
    - Press *"OK"*.
    - In the Certificates Snap-in dialog box, select *"My user account"* and click next.
    - Press *"OK"*.
    - Expand the Certificates section and select *"Trusted Root Certification Authorities"*.
    - Right-click on *"Trusted Root Certification Authorities"*, select *"All Tasks"*, then select import and browse to folder where the *"RootCA.pfx"* is stored.
    - Select *"Personal"*.
    - Right-click on *"Personal"*, select *"All Tasks"*, then select import and browse to folder where the *"clientRevoked.pfx"* is stored.

2. Update the IP address from *"10.10.1.13"* to *"192.168.1.120"*.
3. Load the server certificate in the application *"Cerberus FTP server enterprise"*, the following steps must be performed.
    - Launch the application *"Cerberus FTP server enterprise"*.
    - Open Configure tag and click in the Security option.
    - Load the *"server.crt"*, the *"server.pem" and "ca.crt.*
    - In Client Certificate Verification select "*Verify Certificate*".
    - In CRL File load the list with the revoked certificates.
    - Press save.
    - Click in the General tag and write *"www.test.com"* in the Public Domain Name text box.
    - Press save.



4. In the client machine add the next line *"192.168.1.102 www.test.com"* in the hosts file located in the folder "C:\Windows\System32\drivers\etc" and reboot the client machine.
5. Open a *"Wireshark"* application to verify that the handshake operation is not performed correctly.

6. In the client machine, open the browser and attempt to navigate to the test web (https://www.test.com).

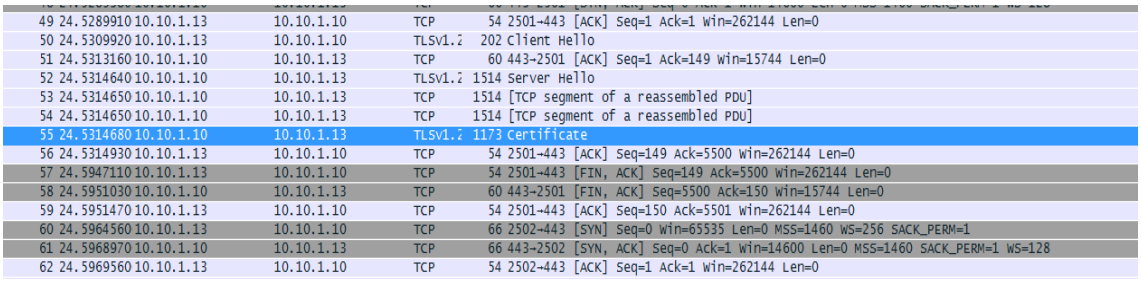
40.3.1.3. Results

The test has been performed in the following platforms:

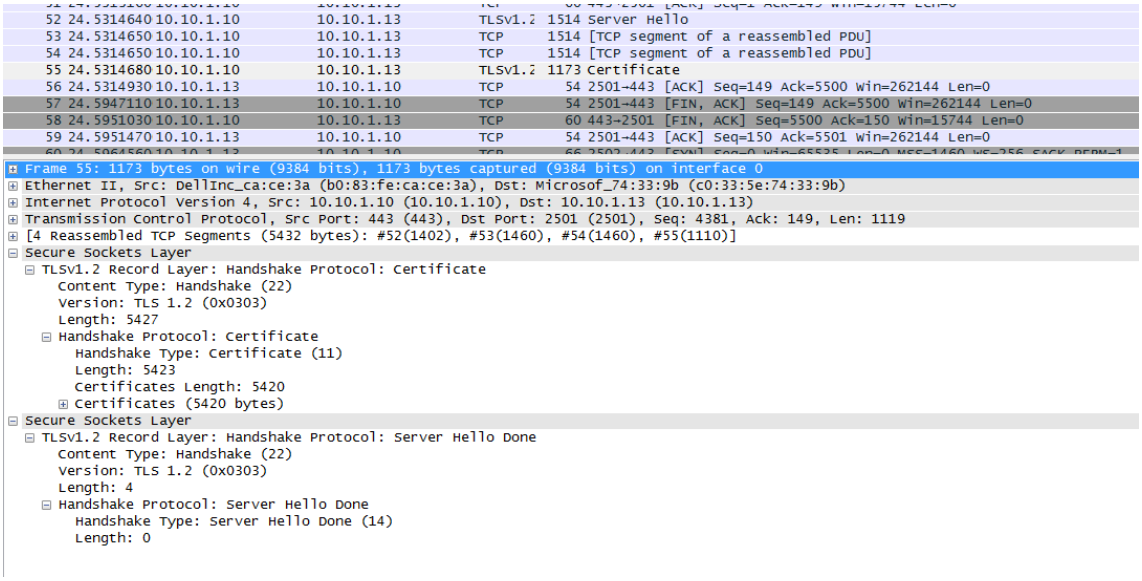
- Microsoft Surface 3 Pro with Windows 10 Enterprise x64.
- Microsoft Surface 3 with Windows 10 Enterprise x64.
- Dell Optiplex 755 with Windows 10 Pro x86.
- Dell Optiplex 755 with Windows Server 2012 R2.
- HP Pro x612 Notebook PC with Windows Pro x64.
- Windows Server 2012 R2 Hyper-V with Windows 10 Home x86.

Analyzing the packets captured by *"Wireshark"*, it can verify that the handshake operation is performed correctly.

The connection between the Client and Server machines *"TLS 1.2"* protocol as it can be appreciated in the following picture.

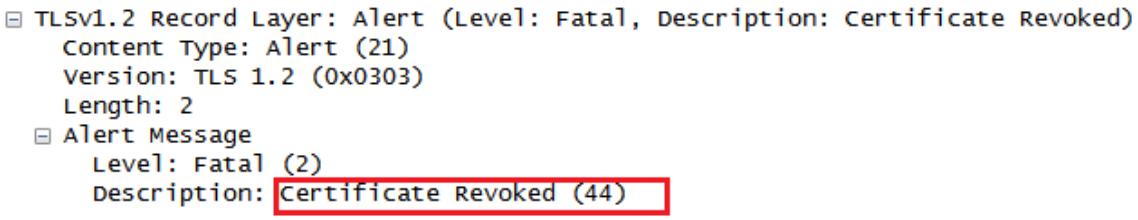


The server sends the certificate (Domain Controller) to the client, to establish the connection.



In the second part of the test, the packets captured with the *"Wireshark"* application shows the handshaking process, whereby can be appreciated that the client sends a *"Certificate Request"*. The content of this packet is displayed in the following capture.

The client sends the certificated *"r1"* that corresponds with the certificated revoked in the CRL. When the server receives the revoked certificate, the server responses as show in the next picture.



In the above pictures can be appreciated that the all steps of the handshaking process use the "*TLS 1.2*" protocol. Therefore, the trusted channel is established with the protocol selected in *"FCS_TLSC_EXT.1"* SFR.

40.3.1.4. Verdict

According to the results presented in the previous section, the evaluator considers that, the tests 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.

40.4. Final Verdict

Due to all test activities 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 this FTP_ITC_EXT.1.1.