Deploying Applications on a Windows Embedded Compact OS with Security Loader

Writer: Randy Ocheltree

Published: December 2011

Applies To: Windows Embedded Compact 7

Abstract

When a Windows Embedded Compact 7 OS image includes the Security Loader, the Security Loader only allows trusted executable files to run. Any application, DLL, or cabinet file must be included in the OS image or be signed with a known certificate before the Security Loader will allow it to load or run. This paper describes how to enable executable files that are not included in the OS image by:

- Signing binary files with one or more certificates.
- Exporting the signing certificates from your development computer.
- Adding the exported certificates to your OS image.

© 2011 Microsoft. All rights reserved.

Contents

© 2011 Microsoft. All rights reserved.

Introduction

You can run applications that are not included in a Security Loader–enabled OS run-time image on a device by following the process described in this article. The process describes how to sign application binary files with certificates and how to add the certificates to the Code Integrity certificate store of a device.

Any device-side applications that you deploy on a device that are not included in the Security Loader–enabled OS run-time image must be accompanied by a signed digital authentication certificate. Otherwise, Security Loader will not permit the application to run on the device. For more information about certificates, see Digital Certificates (http://go.microsoft.com/fwlink/?LinkId=179621).

An application binary must be signed with a certificate, and the certificate used to sign the binary must be included in the Code Integrity certificate store of the device to be trusted by Security Loader. You add certificates to the Code Integrity store by exporting certificates to a Public Key Cryptography Standards (PKCS) #7 file and then adding the PKCS #7 file to your OS image build.

In Windows Embedded Compact 7, the Security Loader, also known as the Loader Verifier Module (LVMOD), replaces the Certification Module (CERTMOD) security feature from previous versions.

Signing Binary Files

Binaries that you include in the Windows Embedded Compact OS run-time image do not need to be signed because they are automatically trusted by Security Loader. However, binaries that you do not include in the Security Loader–enabled run-time image must be signed with a known certificate before they can run.

Choose a signing certificate and a certificate chain that you can use to sign and validate the signatures. For information about certificate trust verification and certificate chains, see Certificate Trust Verification (http://go.microsoft.com/fwlink/?LinkID=205303&clcid=0x409).

We recommend that your certificate chain be at least one level deep so that the root, or any intermediate certificate, can be added to the Code Integrity certificate store of your device, leaving the leaf certificate to be used for signing the binary. (The leaf certificate is the last certificate in the certificate chain and the farthest away from the root.) By using this structure, different leaf certificates can sign for different classes of binaries and then each can chain to the same root or an intermediate certificate. Having different leaf certificates provides a simpler way to revoke or block a certificate and the class of binaries that is signed with that certificate.

If you are using a certificate chain one level deep, sign the binary by using a Personal Information Exchange (PFX) file. The PFX format is also known as the PKCS #12 format.

To sign a binary with a certificate chain one level deep

1. Create a PFX file with the signing certificate and its private key.

© 2011 Microsoft. All rights reserved.

2. Specify the path to this PFX file in the environment variable **BUILDSIGN_CERTPATH**:

```
set BUILDSIGN_CERTPATH=C:\Certificates\CodeSign\Trusted.pfx
```

3. Call **sign <path_to_binary>** to sign the binary.

   Note that you cannot invoke signtool.exe from the Flat Release Directory.

If you are using a certificate chain more than one level deep, sign the binary by using a certificate in the development computer's certificate store.

To sign a binary with a certificate chain more than one level deep

1. Import the signing certificate chain to the development computer's certificate store.
2. Specify the command line to use for **signtool.exe** using the environment variable **BUILDSIGN_CMDLINE**:

```
set BUILDSIGN_CMDLINE=sign /n "<name_of_certificate>"
```

3. Call **sign <path_to_binary>** to sign the binary.

Creating a Public Key Cryptography Standards PKCS #7 File

To add one or more certificates to the Code Integrity store of your device you need to create a Public Key Cryptography Standards (PKCS) #7 file named ciroots.p7b and include the file in your OS image build.

PKCS #7 is the Cryptographic Message Syntax Standard that provides syntax for disseminating certificates or certificate revocation lists and other messages at a root certificate.

Creating a PKCS #7 File Using a Single Certificate

To add a single certificate to the Code Integrity store on the device, you export the certificate to a PKCS #7 file named ciroots.p7b. Then you include the ciroots.p7b file in your OS image build as explained in Building an OS Image That Includes the PKCS #7 ciroots.p7b File.

To create a PKCS #7 file named ciroots.p7b by using one certificate file

1. Right-click the certificate file that was used to sign your binary file, and then click **Open**.
2. In the **Certificate** dialog box, click the **Details** tab, and then click **Copy to the File**.
3. The **Certificate Export Wizard** opens. Click **Next**.

   Note

   If you try to export a certificate with a private key, the **Export Private Key** dialog box appears. If the dialog box appears, select **No, do not export the private key**.

4. In the **Export File Format** dialog box, do the following:
   a. Select **Cryptographic Message Syntax Standard – PKCS #7 Certificates (.P7B)**.

© 2011 Microsoft. All rights reserved.

b. Check **Include all certificates in the certification path if possible**.

c. Click **Next**.

5. In the **File to Export** dialog box, click **Browse**.
6. In the **Save As** dialog box, do the following:
   a. In the **File Name** box, type **ciroots.p7b**.
   b. In the **Save as type** box, select **PKCS #7 Certificates (*.p7b)**.
   c. Click **Save**.
7. In the **File to Export** dialog box, click **Next**.
8. On the **Completing the Certificate Export Wizard** page, click **Finish**.

Creating a PKCS #7 File Using Multiple Certificates

To add multiple certificate files to the Code Integrity store on the device, perform the following procedures to import and then export the certificates to a PKCS #7 file named ciroots.p7b. You then include the ciroots.p7b file in your OS image build as explained in Building an OS Image That Includes the PKCS #7 ciroots.p7b File.

To import multiple certificate files

1. At the command prompt, type **certmgr.msc** and then press **Enter** to open the certificates Management Console snap-in.
2. Click the arrow next to the **Personal** folder.
3. Right-click the **Certificates** folder, point to **All Tasks**, and then click **Import**.
4. The **Certificate Import Wizard** opens. Click **Next**.
5. In the **File to Import** dialog box, click **Browse**.
6. In the **Open** dialog box, select the certificate file to import, and then click **Open**.
7. In the **Certificate Store** dialog box, do the following:
   a. Select **Place all certificates in the following store**.
   b. In **Certificate store**, enter **Personal**.
   c. Click **Next**.
8. On the **Completing the Certificate Import Wizard** page, click **Finish**.
9. Repeat steps 2 thru 8 for each certificate you need to import.

To create a PKCS #7 file named ciroots.p7b by using multiple certificate files

1. At the command prompt, type **certmgr.msc** and then press **Enter** to open the certificates Management Console snap-in, if the console is not already open.
2. Click the arrow next to the **Personal** folder, and then click the **Certificates** folder.
3. In the right pane of the certificates Management Console snap-in window, with **Personal\Certificates** selected in the left pane, select all the imported certificates that you want to export.

© 2011 Microsoft. All rights reserved.

4. After you select the certificates, right-click the highlighted certificates, point to **All Tasks**, and then click **Export**.
5. The **Certificate Export Wizard** opens. Click **Next**.

   Note

   If you try to export a certificate with a private key, the **Export Private Key** dialog box appears. If the dialog box appears, select **No, do not export the private key**.
6. In the **Export File Format** dialog box, do the following:
   a. Select **Cryptographic Message Syntax Standard – PKCS #7 Certificates (.P7B)**.
   b. Check **Include all certificates in the certification path if possible**.
   c. Click **Next**.
7. In the **File to Export** dialog box, click **Browse**.
8. In the **Save As** dialog box, do the following:
   a. In the **File Name** box, type **ciroots.p7b**.
   b. In the **Save as type** box, select **PKCS #7 Certificates (*.p7b)**.
   c. Click **Save**.
9. In the **File to Export** dialog box, click **Next**.
10. On the **Completing the Certificate Export Wizard** page, click **Finish**.

    Note

    After you have exported the certificates, delete them from the **Personal\Certificates** store.

Building an OS Image That Includes the PKCS #7 ciroots.p7b File

The Windows Embedded Compact build system automatically picks up the ciroots.p7b file during the Build Release Directory (Buildrel.bat) and Make Binary Image (Makimg.exe) phases of the build and replaces the empty ciroots.p7b file that comes with Windows Embedded Compact 7.

To add the ciroots.p7b file to your OS image

1. Move the ciroots.p7b file to the **FILES** folder of the project or platform for your OS design.
2. Build your OS image.

   For information about building an OS, see Developing an Operating System Design (http://go.microsoft.com/fwlink/?LinkID=205305&clcid=0x409).

After you build and deploy your OS image that now includes the ciroots.p7b file, your signed applications are ready to run on your device under Security Loader.

© 2011 Microsoft. All rights reserved.

Conclusion

In Windows Embedded Compact 7, the Security Loader, also known as the Loader Verifier Module (LVMOD), replaces the Certification Module (CERTMOD) security feature from previous versions. The Security Loader is an authentication mechanism in Windows Embedded Compact that helps protect the integrity of a device by ensuring that application binaries are trusted before being allowed to run. For binaries to run on the device, they must be included in the Security Loader-enabled run-time image or signed with a certificate. The certificate used to sign the binary must be included in the Code Integrity certificate store of the device to be trusted by Security Loader.

You add certificates to the Code Integrity store by exporting certificates to a Public Key Cryptography Standards (PKCS) #7 file named ciroots.p7b and then adding the ciroots.p7b file to your OS image build. After you build and deploy your OS image that includes the ciroots.p7b file, your signed applications are ready to run on your device.

Additional Resources

- Windows Embedded website (http://go.microsoft.com/fwlink/?LinkID=183524)

© 2011 Microsoft. All rights reserved.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.

© 2011 Microsoft. All rights reserved.