CNT 4603: System Administration
Fall 2013

Scripting – Windows PowerShell – Part 5

Instructor :   Dr. Mark Llewellyn
                    markl@cs.ucf.edu
                    HEC 236, 4078-823-2790
                    http://www.cs.ucf.edu/courses/cnt4603/fall2013

Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida

Code Signing

- In the second set of notes on PowerShell we discussed the execution policy which is part of the security built-in to PowerShell.

- We modified PowerShell's default setting of Restricted, which prevents PowerShell from running any scripts (it is restricted to running in an interactive mode only).

- We changed the setting using the `set-executionpolicy` cmdlet to RemoteSigned, which allowed locally created scripts to be loaded and executed without being digitally signed.

- The other two options are: AllSigned, which is a notch under Restricted, in that all scripts must be digitally signed by a publisher you trust in order to be loaded and executed. The Unrestricted option allows any script to be executed but for non-local scripts the user is prompted for an action.

Code Signing

- In short, code signing is the process of digitally signing scripts, executables, dynamic link libraries (DLLs), and so forth to establish a level of trust for the code.

- The trust granted to digitally signed code is based on two assumptions.

  - One, a signed piece of code ensures that the code hasn't been altered or corrupted since being signed.

  - Two, the digital signature serves to prove the identity of the code's author, which helps you to determine whether the code is safe for execution.

- These two assumptions are a way to ensure the integrity and authenticity of the code. However, these assumptions alone are no guarantee that signed code is safe to run.

Code Signing

- For these two assumptions to be considered valid, you need the digital signature and the infrastructure that establishes a mechanism for identifying the digital signature's originator.

- A digital signature is based on public key cryptography, which utilizes an algorithm for encryption and one for decryption. The algorithms generate a key pair consisting of a private key and a public key. The private key is kept secret so that only the owner has access to it, but the public key can be distributed to other entities. Some form of secure interaction is then required between other entities and the key pair owner. Depending on the type of interaction, one key is used to lock (encrypt) the communication and the other key is used to unlock (decrypt) the communication.

- In digital signatures, the private key is used to generate a signature, and the public key is used to validate the generated signature.

Code Signing

- The process is as follows:

1. A one-way hash (message digest, fingerprint, or compression function), which is a cryptographic algorithm that turns data into a fixed-length binary sequence (one-way means that it difficult to derive the original data from the resulting sequence) of the content being signed is generated by using a cryptographic digest.

2. The hash is then encrypted with the private key, resulting in the digital signature.

3. The hash content is sent to the recipient.

4. The recipient creates another one-way hash of the content and decrypts the hash by using the sender's public key.

5. Finally, the recipient compares the two hashes. If both hashes are the same, the digital signature is valid and the content has not been modified.

Code Signing

- To associate an entity, such as an organization, a person, or a computer with a digital signature, a digital certificate is used.

- A digital certificate consists of the public key and identifying information about the key pair owner.

- To ensure a digital certificate's integrity, it is also digitally signed.

- A digital certificate can be signed by its owner or a trustworthy third party called a certificate authority (CA).

- The act of associating code with the entity that created and published it removed the anonymity of running code. Furthermore, associating a digital signature with a code-signing certificate is much like using a brand name to establish trust and reliability.

Code Signing

- Therefore, armed with this information, users of PowerShell scripts (or any script in general) can make informed decisions about running a script.

- In a nutshell, this is why code signing is an important aspect to system administration activities and in particular to the PowerShell security framework.

Obtaining A Code Signing Certificate

- There are two methods for obtaining a code-signing certificate: generating self-signed certificates and using a CA from a valid public key infrastructure (PKI) like Veri-Sign.

- Generating a self-signed certificate for signing your PowerShell scripts and configuration files is simpler and quicker and has the advantage of not costing anything.

- However, no independent third party verifies the certificate's authenticity, so it doesn't have the same level of trust that's expected from code signing. As a result, no other entity would trust your certificate by default. To distribute your PowerShell script to other machines, your certificate would have to be added as a trusted root CA and a trusted publisher.

Obtaining A Code Signing Certificate

- Although changing what an entity trusts is possible, there are two problems.

- One, entities outside your sphere of control might not choose to trust your certificate because there's no independent method for verifying who you are.

- Two, if the private key associated with your self-signed certificate becomes compromised or invalid, there is no way to manage your certificate's validity on other entities.

- Given these problems, self-signed certificates should be limited to local machines or for testing purposes only.

- If you plan to have your scripts used in an enterprise or the public realm, the second method of using a CA from a PKI should be used.

Generating a Self-Signed Certificate

- Even though the second method is the preferred method for public realm scripting, we'll focus on the first method here so that you can get some practice creating scripts that are digitally signed and then we'll also be able to once again modify the execution policy of PowerShell to require all scripts having a digital signature.

- The method of creating a self-signed certificate is based on using the `makecert` utility, which is part of the .NET Framework Software Development Kit (SDK).

- Follow the steps on the next page to use this utility to create your own digital signature.

Generating a Self-Signed Certificate

1. Download the latest Microsoft .NET Framework SDK available at: www.microsoft.com, and searching for "Microsoft .NET Framework SDK". The current version is 2.0.

2. Install the SDK on the machine where you want to generate the self-signed certificate. Let's use one of our virtual machines for this, like `Mark-Server1`, since that's where you've installed PowerShell.

3. Locate the `makecert` utility. The default location is C:\Program Files\Microsoft .NET\SDK\v2.0\bin.

4. Open a command prompt and change the working directory to this location using the cd command. See page 12.

5. Create a self-signed certificate using the command illustrated on page 12.

```
Administrator: Command Prompt                                          _ ⊡ ☒

Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>cd..

C:\Users>cd..

C:\>cd program files

C:\Program Files>cd microsoft.net

C:\Program Files\Microsoft.NET>cd sdk

C:\Program Files\Microsoft.NET\SDK>cd v2.0

C:\Program Files\Microsoft.NET\SDK\v2.0>cd bin

C:\Program Files\Microsoft.NET\SDK\v2.0\Bin>makecert -r -pe -n "CN= MJL Code Sig
ning" -b 01/01/2011 -e 01/01/2013 -eku 1.3.6.1.5.5.7.3.3 -ss My
Succeeded

C:\Program Files\Microsoft.NET\SDK\v2.0\Bin>_
```

Digital certificate was
successfully created

Makecert –r –pe –n "CN= choose a name" –b beginning date   –e ending date –eku 1.3.6.1.5.5.7.3.3 –ss My

```
PS C:\users\Administrator\MyScripts>
PS C:\users\Administrator\MyScripts> get-childitem cert:\CurrentUser\My -codesign


    Directory: Microsoft.PowerShell.Security\Certificate::CurrentUser\My


Thumbprint                                Subject
----------                                -------
D5F99C4AC4CA7734BC1186AF751E45F384F6E040  CN=MJL Code Signing



PS C:\users\Administrator\MyScripts>
```

Run this cmdlet in PowerShell to see your self-signed digital certificate.

Signing PowerShell Scripts

- When signing a PowerShell script, you use the `set-AuthenticodeSignature` cmdlet, which takes two required parameters.

- The first parameter, `filePath`, is the path and filename for the script to be digitally signed.

- The second parameter, `certificate`, is the X.509 certificate used to sign the script.

- To obtain the X.509 certificate in a format the `set-AuthenticodeSignature` cmdlet understands, you retrieve the certificate as an object with the `get-ChildItem` cmdlet, as shown on the next page.

Signing PowerShell Scripts

```
Administrator: Windows PowerShell                                                              _ □ X

PS C:\users\Administrator\MyScripts> set-authenticodesignature -filePath PS-Part5-p15-signed.ps1 -ce
ditem cert:\CurrentUser\My -codeSigningCert)[0] -includeChain "All"


    Directory: C:\users\Administrator\MyScripts


SignerCertificate                          Status                              Path
-----------------                          ------                              ----
D5F99C4AC4CA7734BC1186AF751E45F384F6E040   Valid                               PS-Part5-p15-signed


PS C:\users\Administrator\MyScripts>
```

Actual command string is:

Set-authenticodesignature –filePath <filename> –certificate @(get-childitem cert:\CurrentUser\My –codeSigningCert)[0] –includeChain "All"

Signing PowerShell Scripts

- To retrieve the certificate you want from the user's certificate store, you use the `get-childitem` cmdlet with the `codeSigningCert` switch parameter.

- This switch parameter can only be used with the PowerShell Certificate provided and acts as a filter to force the `get-childitem` cmdlet to retrieve only code-signing certificates.

- To ensure that the entire certificate chain is included in the digital signature, the `includeChain` parameter is used.

- After the `set-authenticodesignature` cmdlet has been executed successfully, the signed file has a valid digital signature block containing the digital signature appended to it.

Signing PowerShell Scripts

- The digital signature block in a PowerShell script is always the last item in the script and can be found easily because it is enclosed between `SIG # Begin signature block` and `SIG # End signature block`, as shown in the example script we just digitally signed (see next page).

File   Edit   Search   View   Encoding   Language   Settings   Macro   Run   Plugins   Window   ?   X

PS-Part5-p15.ps1

```
39    #-----------------------------------------------------------------------------
40    # SIG # Begin signature block
41    # MIIECQYJKoZIhvcNAQcCoIID+jCCA/YCAQExCzAJBgUrDgMCGgUAMGkGCisGAQQB
42    # gjcCAQSgWzBZMDQGCisGAQQBgjcCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
43    # AgEAAgEAAgEAAgEAAgEAMCEwCQYFKw4DAhoFAAQUVP4ZR8CZTC0cjwbSV6/6ikxc
44    # FBegggIkMIICIDCCAYmgAwIBAgIQIx95PfHJMbtIE8hu9ItkAzANBgkqhkiG9w0B
45    # AQQFADAbMRkwFwYDVQQDExBNSkwgQ29kZSBTaWduaW5nMB4XDTExMDEwMTA1MDAw
46    # MFoXDTEzMDEwMTA1MDAwMFowGzEZMBcGA1UEAxMQTUpMIENvZGUgU2lnbmluZzCB
47    # nzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAtsWx2f821KTgBsP+bOXZcvFo3mm3
48    # v1RQps4S0P/XgJ4QAAf8ruNTKjXbJ3b/fG62yJ0M3+Bwqmpk5ZdpO14gG531Pedv
49    # xEsAeXio/ORmLIVZr/MycBzKr7clurOE6eZFV+H/Jz/s6630x7v11ffT4Lk5OygV
50    # Eq+0c1BSFOe6M8MCAwEAAaN1MGMwEwYDVR01BAwwCgYIKwYBBQUHAwMwTAYDVR0B
51    # BEUwQ4AQBgKfhK2T1IhWVC7go2BS36EdMBsxGTAXBgNVBAMTEE1KTCBDb2RlIFNp
52    # Z25pbmeCECMfeT3xyTG7SBPIbvSLZAMwDQYJKoZIhvcNAQEEBQADgYEAmF3x0qA3
53    # EvhSlehx6Efa/bZqLy4DezfmKjDiRaUqbAj7hnkp8S1eLvDyg0ukfkGAXnoEhAuQ
54    # CxMA47vCnoj0nEG6yxT1ybbw5jv+B8mZfHqPglrrao8ikVrNUFgp5xOk4EI53iQM
55    # HpJmps+/ILOuQkHyB44YSvJmCUJH+1GmyA0xggFPMIIBSwIBATAvMBsxGTAXBgNV
56    # BAMTEE1KTCBDb2RlIFNpZ25pbmcCECMfeT3xyTG7SBPIbvSLZAMwCQYFKw4DAhoF
57    # AKB4MBgGCisGAQQBgjcCAQwxCjAIoAKAAKECgAAwGQYJKoZIhvcNAQkDMQwGCisG
58    # AQQBgjcCAQQwHAYKKwYBBAGCNwIBCzEOMAwGCisGAQQBgjcCARUwIwYJKoZIhvcN
59    # AQkEMRYEFGpRbl64fKVFkpA474eizyhrWi0lMA0GCSqGSIb3DQEBAQUABIGAWkdY
60    # Bp60bjuCHZYnOVA2MZOhusTUqBulrLytnVfiWYDqlYDZ4FTbasgrdpNy7Eobr7Be
61    # XMgam8SFAfCxMtQ2o7ioJ6yDohYDfPtMMqUvnzxepAQ3Y89gt2/lokPCb2saOL9t
62    # zNeU6SjhkWioukd3RTd8f5ip8dChsye49OlnFDk=
63    # SIG # End signature block
64
```

Verifying A Digital Signature In PowerShell

- To verify the digital signature of PowerShell scripts, you use the `get-authenticodesignature` cmdlet.

- This cmdlet returns a valid status or an invalid status, such as `HashMismatch`, indicating a problem with the script.

- Page 20 illustrates verifying a script which contains a valid digital signature.

- Page 21 illustrates a script that was modified after the digital signature was applied.

Verifying A Digital Signature In PowerShell

Verifying A Digital Signature In PowerShell

Verifying A Digital Signature In PowerShell

- Once you begin to digitally sign your scripts you'll need to reset the PowerShell execution policy to AllSigned.

- The next page illustrates doing this using the `set-executionpolicy` cmdlet.

- Once this is done, I attempted to run the modified version of the digitally signed script. Notice that PowerShell will not allow it to execute due to the invalid signature.

```
Administrator: Windows PowerShell                                         _ □ X

PS C:\users\Administrator\MyScripts> get-authenticodesignature PS-Part5-p15-invalidsignature.ps1


    Directory: C:\users\Administrator\MyScripts


SignerCertificate                              Status                      Path
-----------------                              ------                      ----

D5F99C4AC4CA7734BC1186AF751E45F384F6E040       HashMismatch                PS-Part5-p15-invali


PS C:\users\Administrator\MyScripts> set-executionpolicy AllSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution
policy might expose you to the security risks described in the about_Execution_Policies help topic.
 Do you want to change the execution policy?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): y
PS C:\users\Administrator\MyScripts> get-executionpolicy
AllSigned
PS C:\users\Administrator\MyScripts> .\PS-Part5-p15-invalidsignature.ps1
File C:\users\Administrator\MyScripts\PS-Part5-p15-invalidsignature.ps1 cannot be loaded. The conten
\Administrator\MyScripts\PS-Part5-p15-invalidsignature.ps1 may have been tampered because the hash o
 match the hash stored in the digital signature. The script will not execute on the system. Please s
signing" for more details..
At line:1 char:36
+ .\PS-Part5-p15-invalidsignature.ps1 <<<<
    + CategoryInfo          : NotSpecified: (:) [], PSSecurityException
    + FullyQualifiedErrorId : RuntimeException


PS C:\users\Administrator\MyScripts>
```
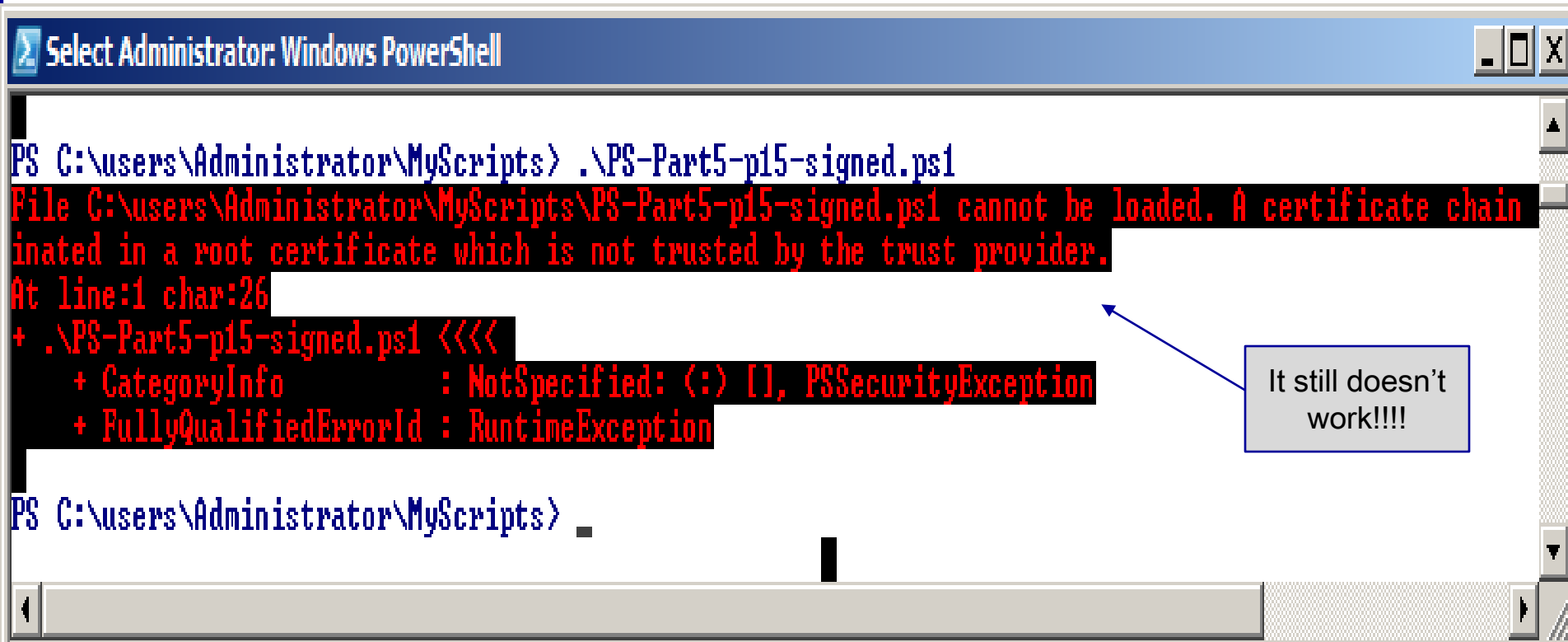
Verifying A Digital Signature In PowerShell

- What happens if we attempt to execute a valid digitally signed script in PowerShell now, assuming that we've changed the execution policy to AllSigned?

```
Select Administrator: Windows PowerShell                                    _ □ X

PS C:\users\Administrator\MyScripts> .\PS-Part5-p15-signed.ps1
File C:\users\Administrator\MyScripts\PS-Part5-p15-signed.ps1 cannot be loaded. A certificate chain
inated in a root certificate which is not trusted by the trust provider.
At line:1 char:26
+ .\PS-Part5-p15-signed.ps1 <<<<
    + CategoryInfo          : NotSpecified: (:) [], PSSecurityException
    + FullyQualifiedErrorId : RuntimeException


PS C:\users\Administrator\MyScripts>
```

It still doesn't work!!!!

Verifying A Digital Signature In PowerShell

- The problem, as you can see from the PowerShell script output, is that we've created the digital signature and signed the script, but the we haven't verified that we trust the creator of the signature yet.

- To do this go to the file that contains the digital signature and right click on it and select Properties.

- Follow the steps on the next few pages to

- You'll see a tab which is titled Digital Signatures. Select this tab.

1. Right click on the file name and select Properties to see this dialog.
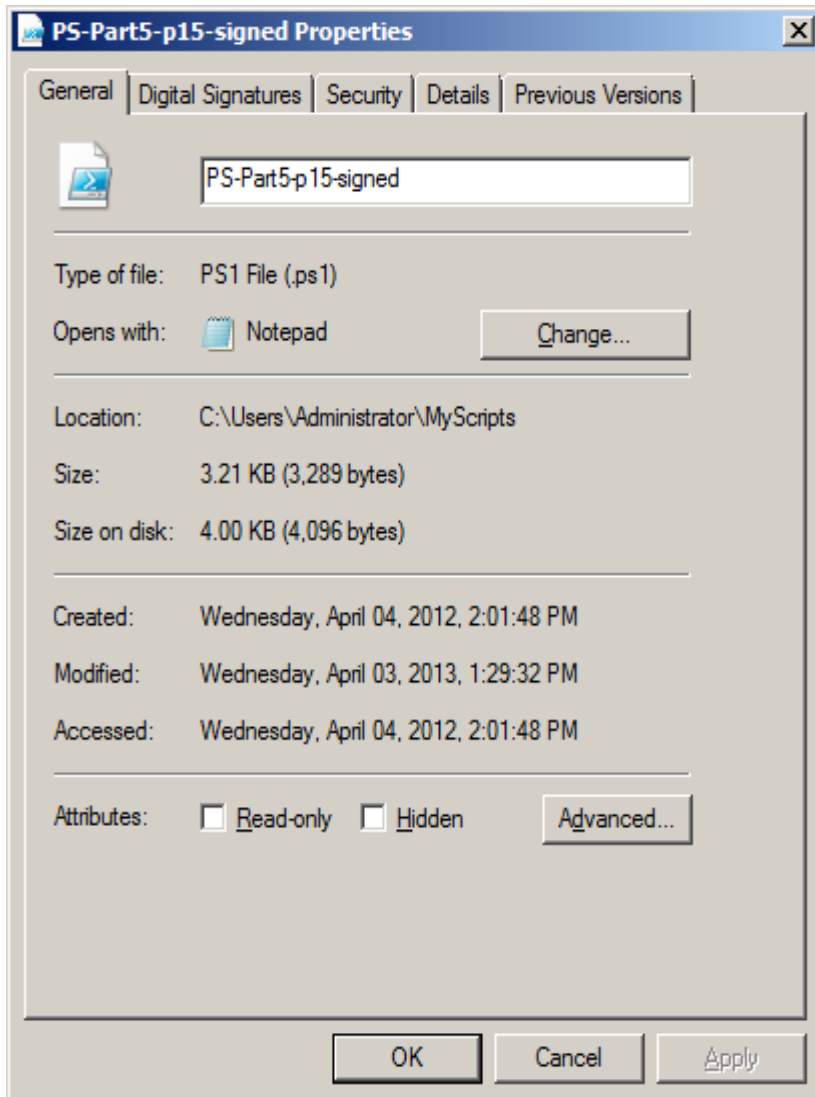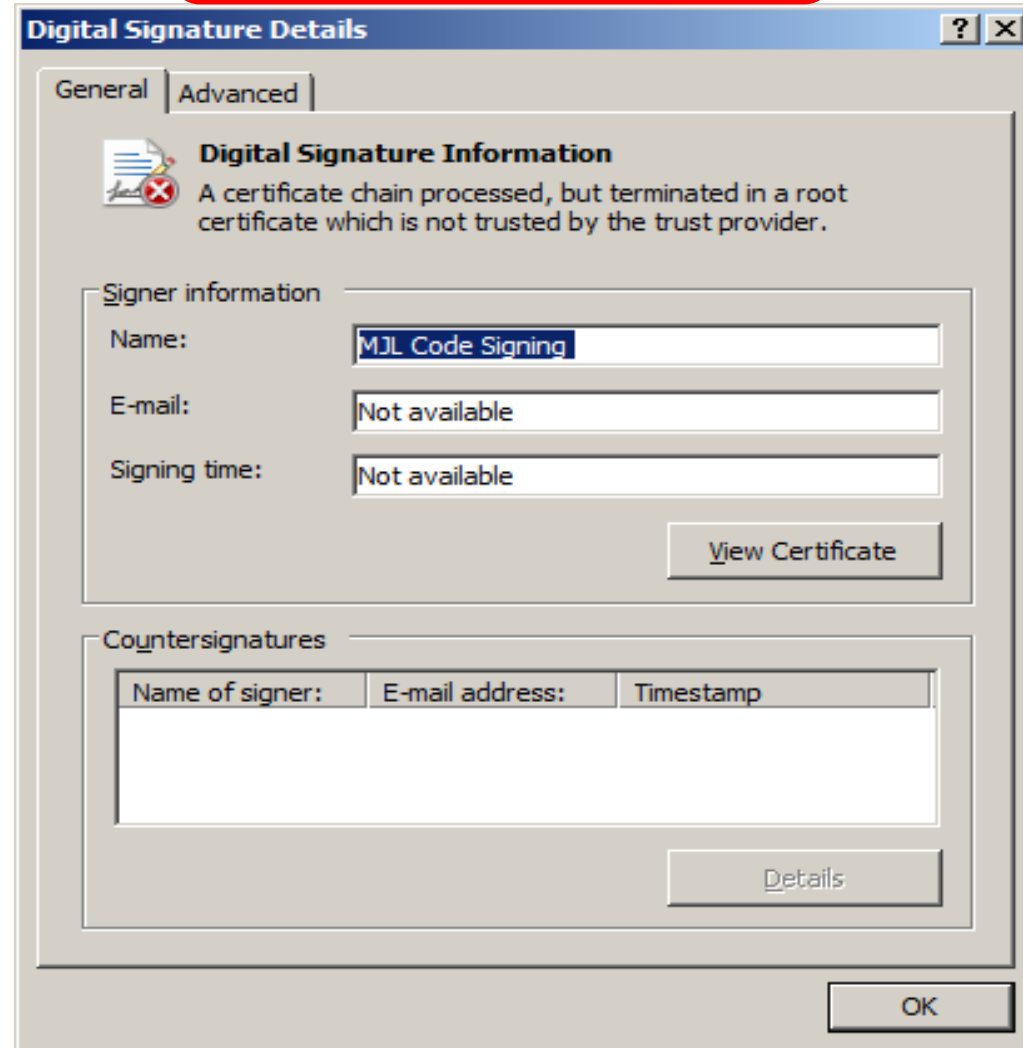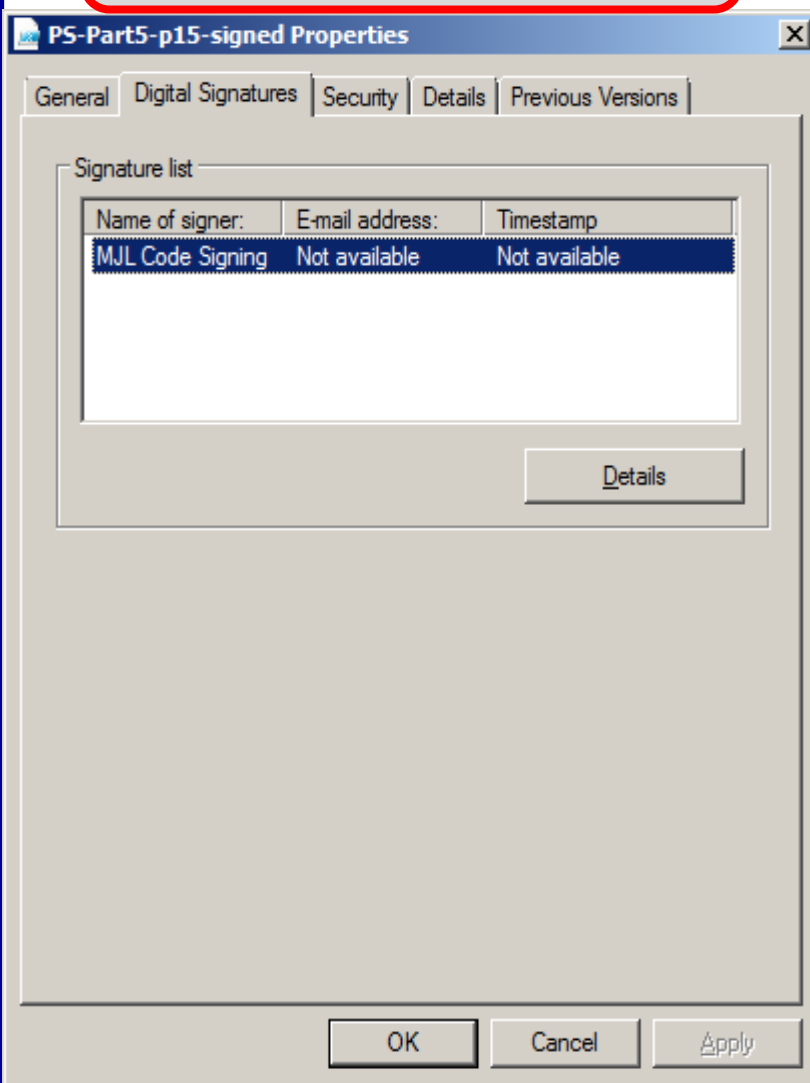
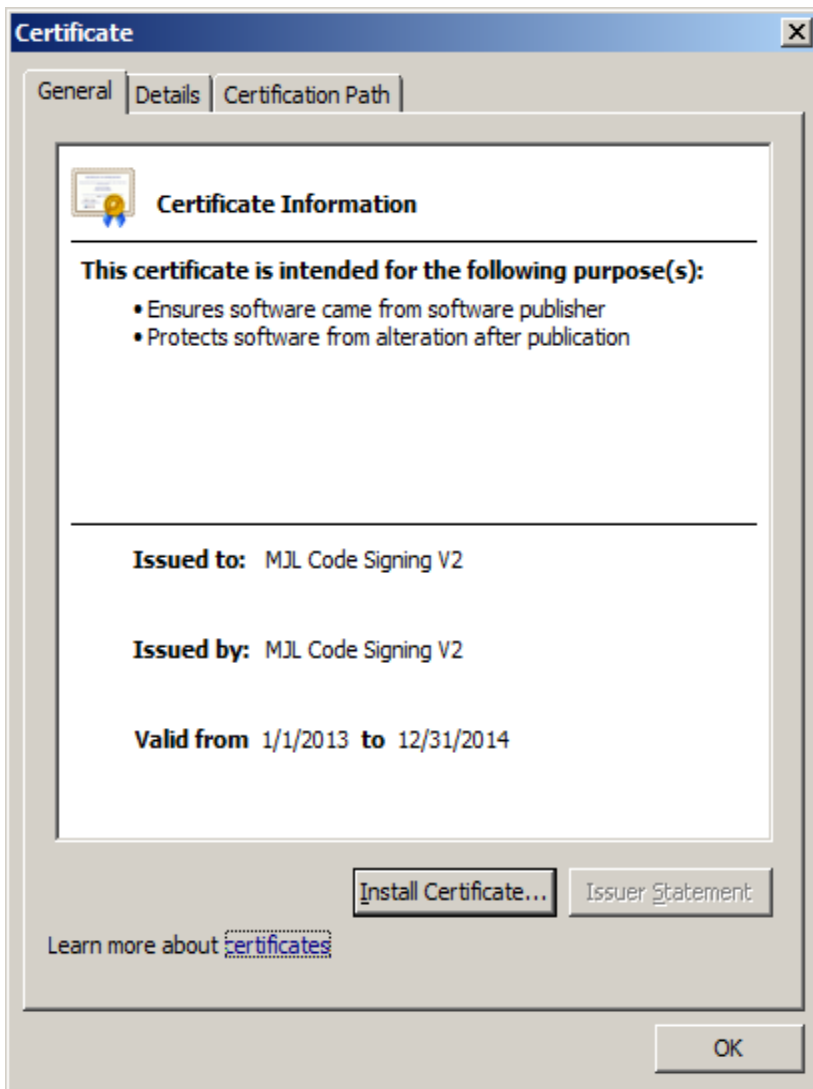2. Select the Digital Signatures tab.
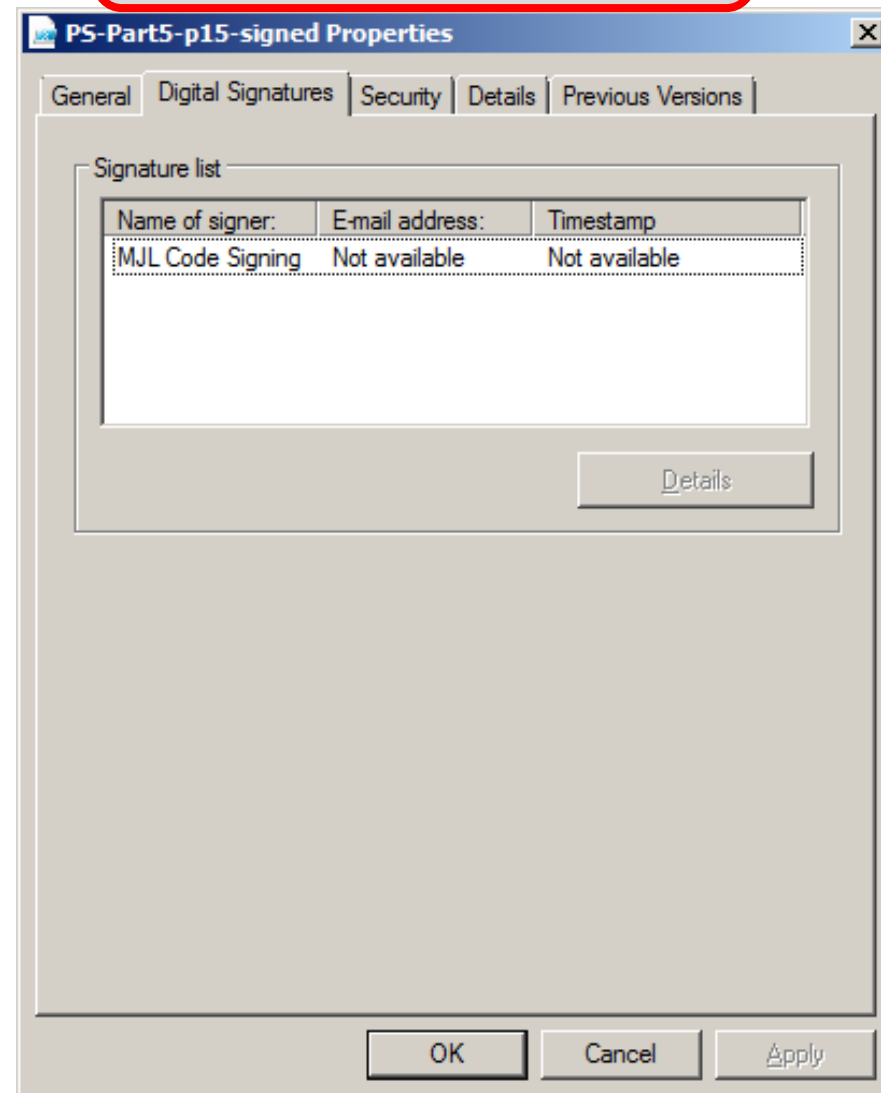
3. Highlight the name of signer you wish to trust.

4. You should see this dialog now. Click View Certificate.

PS-Part5-p15-signed Properties

General | Digital Signatures | Security | Details | Previous Versions

Signature list

Name of signer:	E-mail address:	Timestamp
MJL Code Signing	Not available	Not available

Details

OK | Cancel | Apply

Digital Signature Details

General | Advanced

Digital Signature Information
A certificate chain processed, but terminated in a root certificate which is not trusted by the trust provider.

Signer information

Name: MJL Code Signing

E-mail: Not available

Signing time: Not available

View Certificate

Countersignatures

Name of signer:	E-mail address:	Timestamp

Details

OK

5. Click the Install Certificate button.

6. The Certificate Import Wizard will now run. Click Next.

Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):
- Ensures software came from software publisher
- Protects software from alteration after publication

Issued to: MJL Code Signing V2

Issued by: MJL Code Signing V2

Valid from 1/1/2013 **to** 12/31/2014

Install Certificate... | Issuer Statement

Learn more about certificates

OK

Certificate Import Wizard

Welcome to the Certificate Import Wizard

This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

< Back | Next > | Cancel

7. Choose your option here. I just let mine default to automatically selecting the certificate store based on the type of certificate. Then click Next.

Certificate Import Wizard ✕

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

⦿ Automatically select the certificate store based on the type of certificate

○ Place all certificates in the following store

Certificate store:

[   ] [Browse...]

Learn more about certificate stores

[< Back] [Next >] [Cancel]

8. Click Finish and you're done
   (well almost – see next page)

Certificate Import Wizard

Completing the Certificate Import Wizard

The certificate will be imported after you click Finish.

You have specified the following settings:

Certificate Store Selected	Automatically determined by t
Content	Certificate

< Back   Finish   Cancel

9. A security warning will appear. Read it and then click Yes.

10. Now go back to PowerShell and attempt to rerun the digitally signed script. You will see the following message appear. Click A to always run scripts with this digital certificate, subsequent runs will not issue this prompt. Clicking the A option places the publisher's certificate in the Trusted Publisher's certificate store. Also, the root CA's certificate is placed in the Trusted Root Certification Authorities certificate store, if it isn't already there.

```
PS C:\users\Administrator\MyScripts> .\PS-Part5-p15-signed.ps1

Do you want to run software from this untrusted publisher?
File C:\users\Administrator\MyScripts\PS-Part5-p15-signed.ps1 is published by CN=MJL Code Signing
and is not trusted on your system. Only run scripts from trusted publishers.
[V] Never run   [D] Do not run   [R] Run once   [A] Always run   [?] Help (default is "D"):

PS C:\users\Administrator\MyScripts> .\PS-Part5-p15-signed.ps1 . textfile.csv
Checking Template Path    [OK]
Checking Import File      [OK]
PS C:\users\Administrator\MyScripts>
```