ForeScout

Digital Signing of Windows PE Files in CounterACT®

CounterACT Technical Note

Version 2.0
As of 25th May 2017

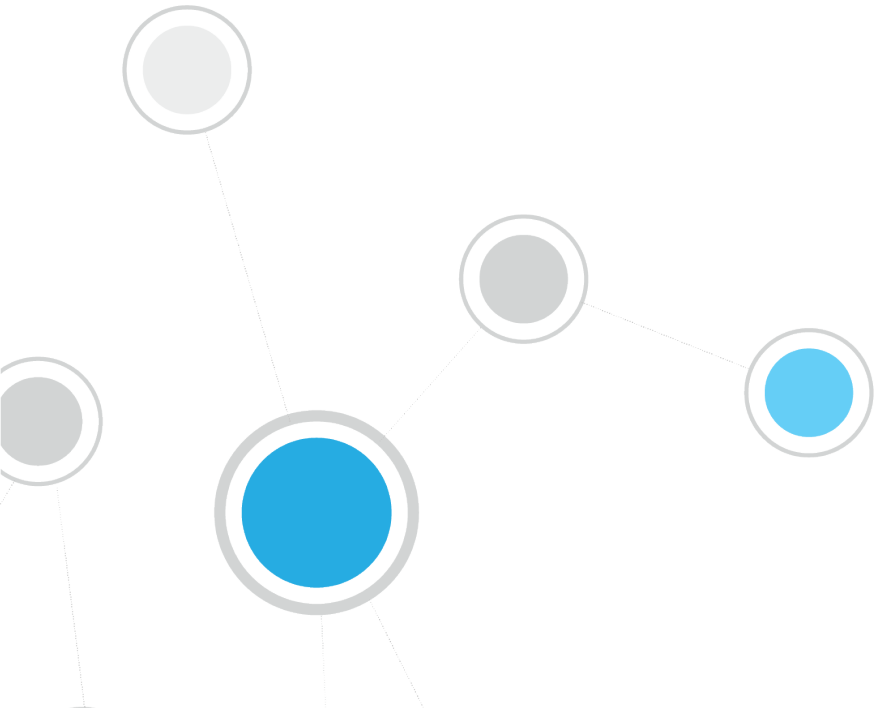


Table of Contents

Scope 3
About Portable Executable (PE) Files 3
Certificate Based Signing of Executable Code 3
Current (as of 25th May 2017) Code Signing Practice 4
Appendix I - History of Changes to ForeScout Code Signing of Windows Executables 7
Code Distributed Prior to 25th October 2016 8
Code Distributed Between 25th October 2016 and 25th May 2017 8
References 11

Scope

This document describes changes which affect all ForeScout CounterACT® Plugins and Modules that contain Windows executables and are released after 1st 2017. This includes, but is not limited to:

- HPS Inspection Engine
- Windows Applications
- Hardware Inventory
- IOC Scanner
- Microsoft SMS/SCCM

About Portable Executable (PE) Files

Microsoft has defined the Portable Executable (PE) file format as a container for executables and object files. CounterACT makes use of various Windows executables, including the SecureConnector executable, the ForeScout Remote Inspection service (fsprocsvc), and various other utility .EXE and .VBS files.

This document describes changes to the algorithms that CounterACT uses for certificate-based signing of files distributed in the PE format.

Certificate Based Signing of Executable Code

Code signing is used to ensure the authenticity and integrity of the code. When executables are signed by a certificate chain of a trusted issuer, entities that run the code can validate that:

- The code being run was provided by the company that signed it – in this case, ForeScout.
- The code has not been tampered with since ForeScout released it.

Code signing supports transparent background CounterACT interaction with endpoints in the following ways:

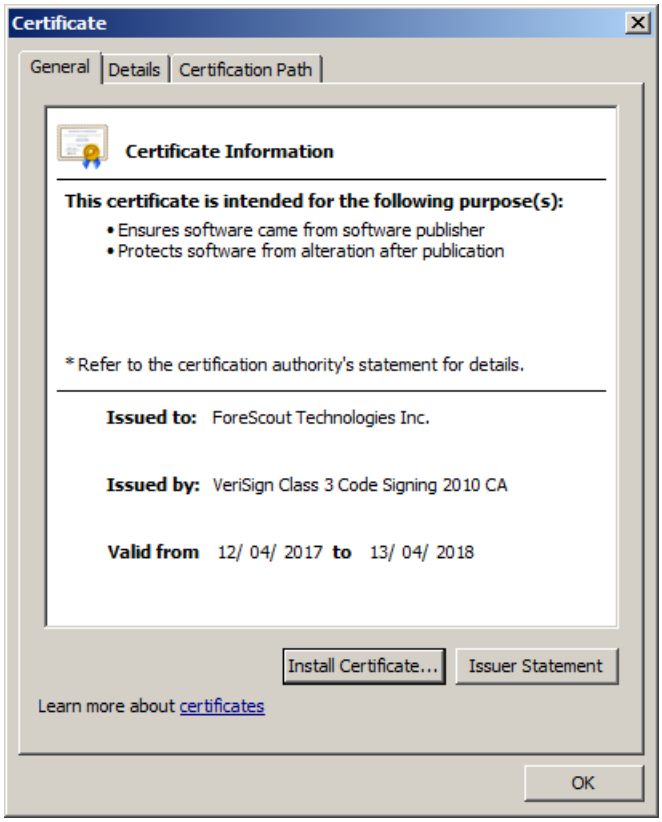
- ***At the browser level***, browsers perform multiple checks on applications that users download, including reputation and potentially anti-malware checks. One of these checks looks at whether the application was signed by a trusted source. Browsers apply various types of these checks when end users download SecureConnector through their browsers. Appropriately signed code satisfies these checks without end user interaction.
- ***At the endpoint OS level***, Windows itself performs checks on executables when running them. Depending on the version of Windows and the security level set, users receive warnings when running an executable which is not signed or not signed with a valid certificate. The warning prompts the user to make a conscious decision as to whether they want to go ahead and run the executable. Correctly signing such files with valid certificates, especially in the case of files that were downloaded from the web, helps avoid this warning.

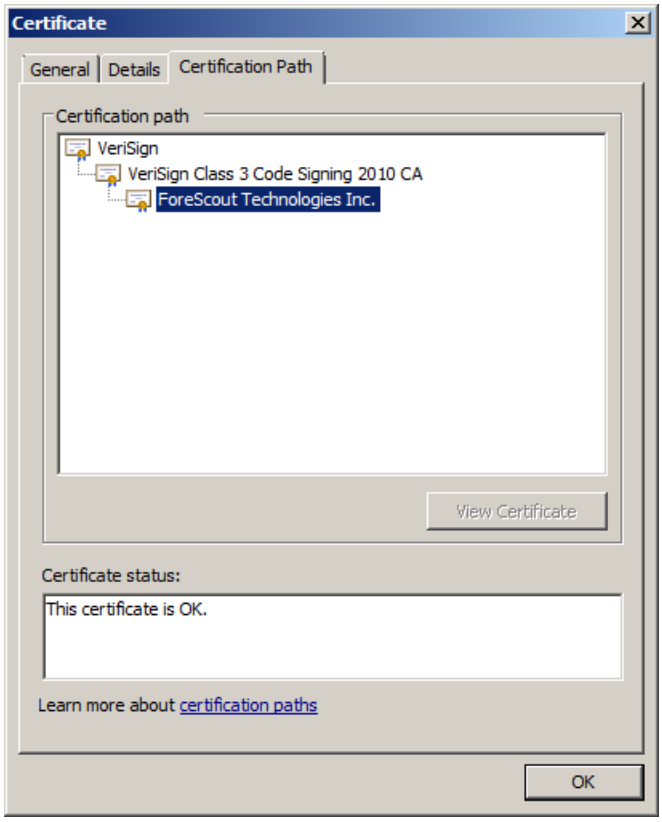
- ***At the endpoint policy/application whitelist level***, for security reasons, some system administrators lock down their desktops and laptops to ensure that only whitelisted executables are run. One strong method of whitelisting requires that executables are signed by one of a number of approved certificates. Depending on your implemented policies and plugin configuration, CounterACT may run scripts and utility executables on your Windows endpoints to retrieve properties and perform actions.

Current (as of 25th May 2017) Code Signing Practice

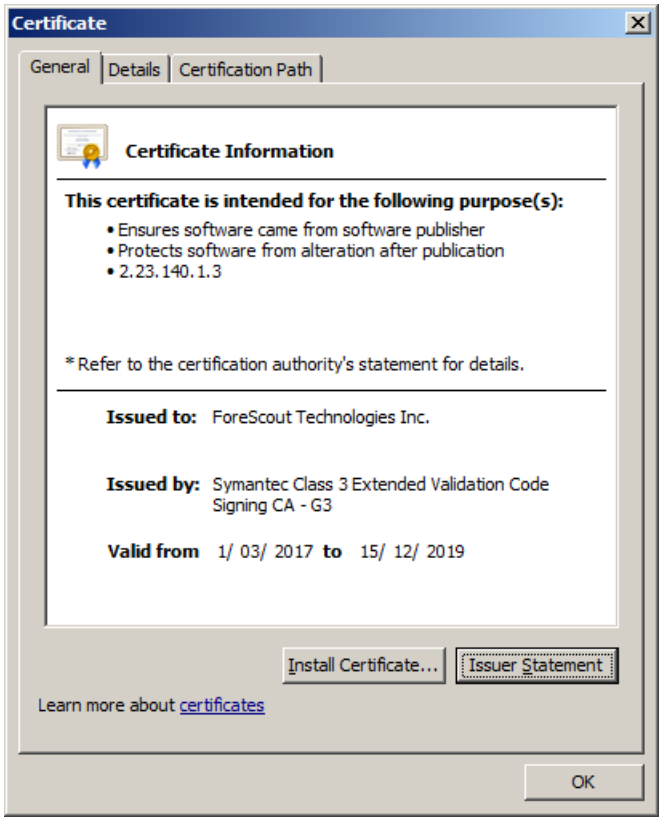
ForeScout utilizes two different types of digital certificates for codesigning of Windows executables:

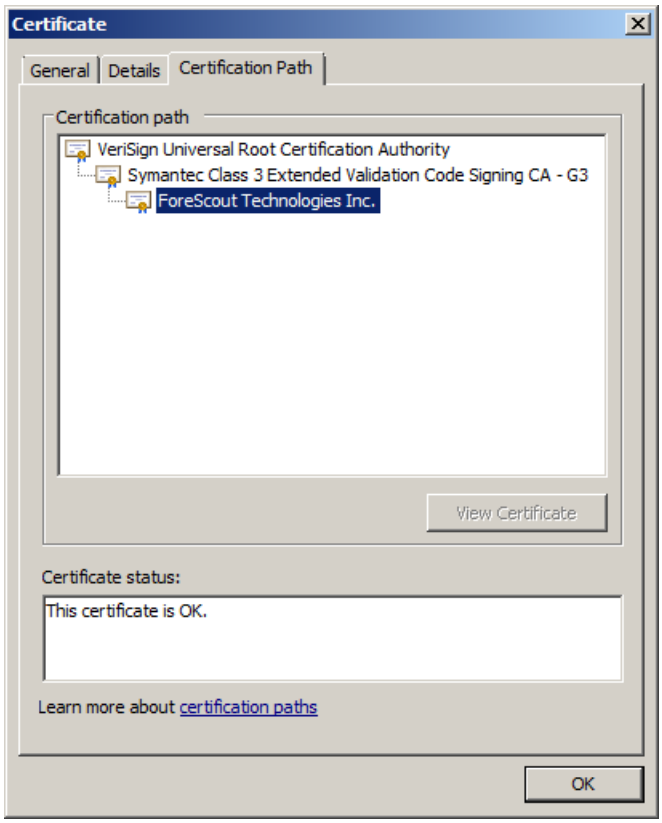
1. A Verisign (Symantec)-issued SHA-1 digital certificate with a SHA-1 root certificate.





2. A Symantec-issued SHA-256 digital certificate with a SHA-256 root certificate.


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
• 2.23.140.1.3
* Refer to the certification authority's statement for details.
Issued to: ForeScout Technologies Inc.
Issued by: Symantec Class 3 Extended Validation Code Signing CA - G3
Valid from 1/ 03/ 2017 to 15/ 12/ 2019
Install Certificate...
Issuer Statement
Learn more about certificates
OK

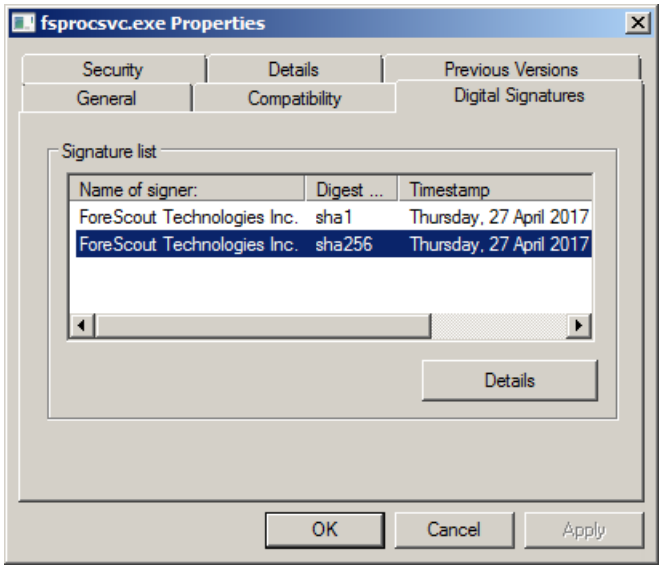

Certificate
General
Details
Certification Path
Certification path
VeriSign Universal Root Certification Authority
Symantec Class 3 Extended Validation Code Signing CA - G3
ForeScout Technologies Inc.
View Certificate
Certificate status:
This certificate is OK.
Learn more about certification paths
OK

ForeScout maintains multiple instances of each certificate type, yet they all have the same common names as shown above. The differences between the certificates are only in the form of the validity dates and the serial numbers.

All Windows PE files included with CounterACT (including CounterACT Plugins and Modules) are digitally signed as described in the following table.

File type	Certificate	Signature Digest Algorithm
EXE	SHA-1	SHA-1
	SHA-256	SHA-256
VBS	SHA-256	SHA-1
MSI	SHA-256	SHA-1

Dual signing of .EXE files ensures that Microsoft Authenticate trusts the executable when it is downloaded by endpoints running any recent or legacy version of Windows. To verify dual signing, right-click on the file, select **Properties** and then view the **Digital Signatures** tab. You should see the following:



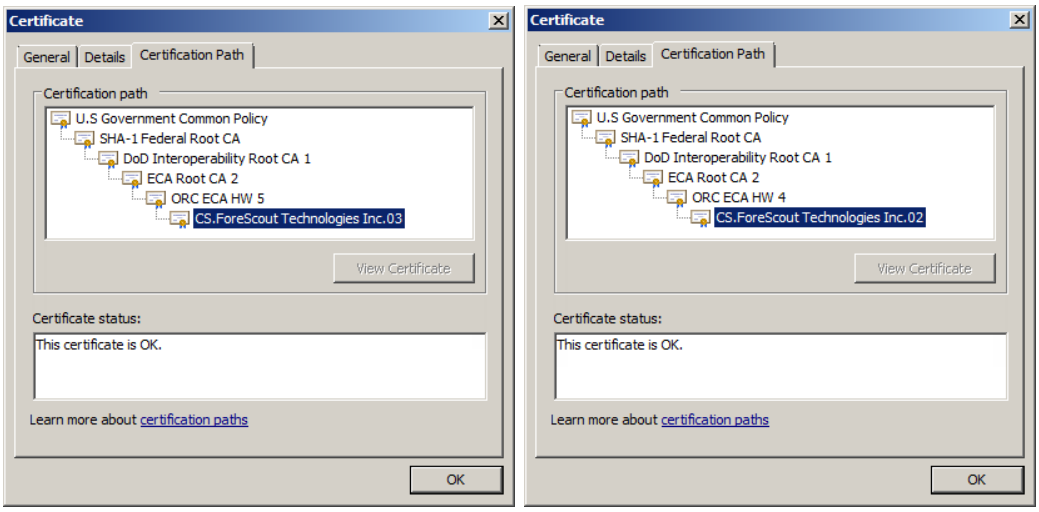
Since VBS files cannot be dual signed, ForeScout signs all VBS files with the SHA-256 based certificate. MSI installer packages use the singing method applied to VBS files. Older versions of Windows which do not support SHA-256 cannot verify this signature, and consider such files unsigned. To run CounterACT VBS scripts and MSI installers, endpoints in your environment that still run these versions of Windows must not have a security policy that prevents running unsigned scripts.

Appendix I - History of Changes to ForeScout Code Signing of Windows Executables

This section describes the background for previous changes to the code signing procedure used by ForeScout, and describes changes for specific files.

Code Distributed Prior to 25th October 2016

All relevant Windows files were signed by a code signing certificate issued to ForeScout. Files are signed by one of two certificates, issued either to “CS.ForeScout Technologies Inc.02” or “CS.ForeScout Technologies Inc.03”. Depending on the certificate used to sign the files, the certificate chain looks like one of the following:



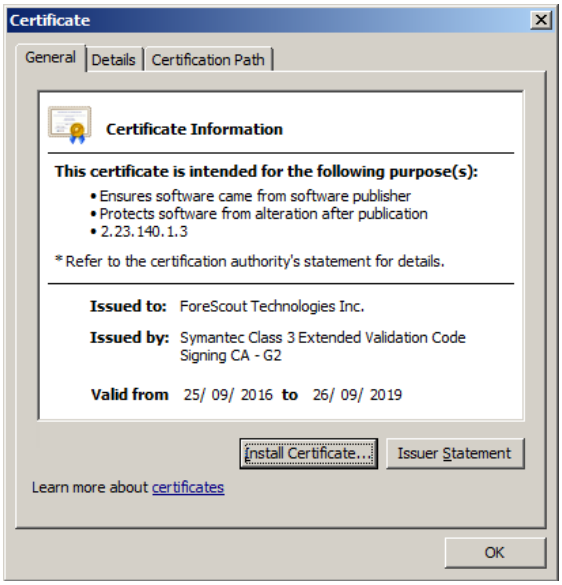
These certificates are SHA-1 based certificates and the digital signatures are SHA-1 based.

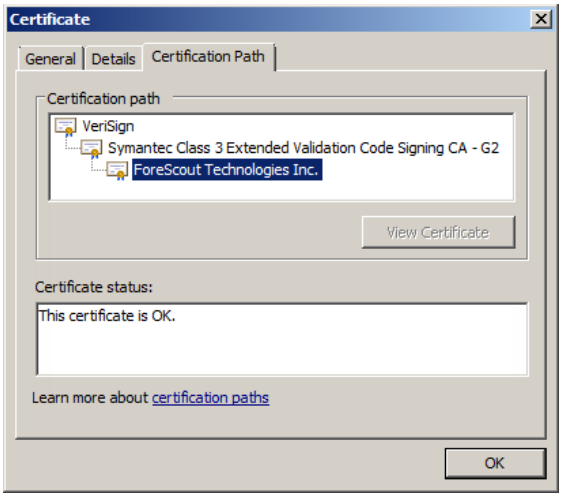
Code Distributed Between 25th October 2016 and 25th May 2017

Due to increasing attacks on the SHA-1 algorithm, SHA-1 certificates were no longer considered secure. Industry best practices recommended the use of SHA-256 based certificates.

During 2016, Microsoft began actively discouraging the use of SHA-1 certificates for code signingm but held off full deprecation of SHA-1 in order to support older Windows operating systems that cannot work with SHA-256 certificates. Similarly, ForeScout wished to continue support for these legacy systems, including Windows XP and Windows Server 2003.

ForeScout introduced the use of a second code signing certificate for signing PE files. The new certificate is SHA-256 based and issued to “ForeScout Technologies Inc.”. This certificate has a SHA-1 root certificate.

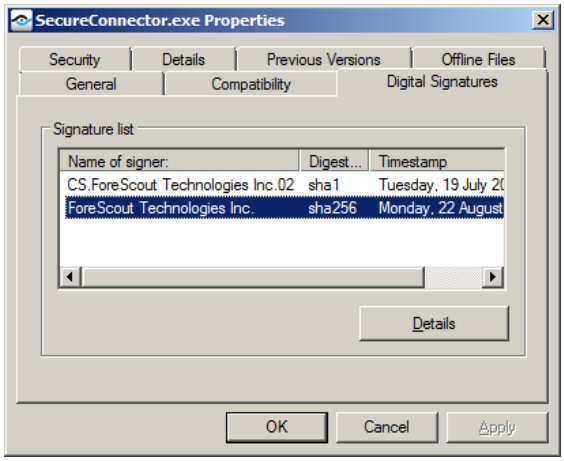




Btween October 25, 2016 and May 25, 2017, ForeScout code containing .EXE and .VBS files were signed as follows:

.EXE Files

.EXE files were dual signed by two separate code signing certificates. The first signature utilized the original SHA-1 code-signing certificate described above. The second signature utilized the new SHA-256 digital certificate. To verify dual signing, right-click on the file, select **Properties** and then view the **Digital Signatures** tab. You should see the following:



.VBS Files

Since VBS files cannot be dual-signed, ForeScout signs all VBS files with the SHA-256 based certificate. Older versions of Windows which do not support SHA-256 cannot verify the signature, and such scripts are considered unsigned. To run CounterACT VBS scripts, endpoints in your environment that still run these versions of Windows must not have a security policy that prevents running unsigned scripts.

References

1. Windows Enforcement of SHA1 Certificates - http://social.technet.microsoft.com/wiki/contents/articles/32288.windows-enforcement-of-authenticode-code signing-and-timestamping.aspx
2. Windows Script Host 5.6 (with references to digitally signing scripts) - https://msdn.microsoft.com/en-us/library/ms974613.aspx and https://technet.microsoft.com/en-us/library/ee176795.aspx
3. Introduction to Code Signing - https://msdn.microsoft.com/en-us/library/ms537361(v=vs.85).aspx

Legal Notice

2018-04-10 09:21