Software Integrity & Authenticity

Implementation Guidance for CIP-010-3 R1 Requirement Part 1.6

Version 1.0

Approved: November 6, 2017

Document Approval			
Ken Keels Director, Practices and Initiatives		Version 1.0	Date Approved: 11/6/2017

Disclaimer

This document was created by the North American Transmission Forum (NATF) to facilitate industry work to improve transmission reliability and resiliency. The NATF reserves the right to make changes to the information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis. "North American Transmission Forum" and its associated logo are trademarks of NATF. Other product and brand names may be trademarks of their respective owners. This legend should not be removed from the document.

Open Distribution

Contents

Revisions

Date	Version	Notes
11/6/2017	1.0	Original issue

Introduction

On July 21, 2016, the Federal Energy Regulatory Commission (FERC) issued Order No. 829 directing the North American Electric Reliability Corporation (NERC) to develop a new or modified Reliability Standard that addresses cyber security supply chain risk management for industrial control system hardware, software, and computing and networking services associated with Bulk Electric System (BES) operations as follows:

[The Commission directs] NERC to develop a forward-looking, objective-based Reliability Standard to require each affected entity to develop and implement a plan that includes security controls for supply chain management for industrial control system hardware, software, and services associated with bulk electric system operations. The new or modified Reliability Standard should address the following security objectives, [discussed in detail in the Order]: (1) software integrity and authenticity; (2) vendor remote access; (3) information system planning; and (4) vendor risk management and procurement controls.

Regarding software integrity and authenticity, the Commission states in Order No. 829 the following:

The new or modified Reliability Standard must address verification of: (1) the identity of the software publisher for all software and patches that are intended for use on BES Cyber Systems; and (2) the integrity of the software and patches before they are installed in the BES Cyber System environment.[1]

The software integrity and authenticity is addressed with additions to CIP-010-3 Requirement 1; specifically the addition of requirement part 1.6. The North American Transmission Forum (NATF) developed this document to outline considerations and potential approaches for implementing the requirements in CIP-010-3 R1 part 1.6, depending on the Responsible Entity's[2] individual facts and circumstances. The examples provided herein are not the only approaches for complying with CIP-010-3, rather, Responsible Entities may choose alternative approaches that better fit their situations.[3]

[1] Order No. 829, paragraph 48

[2] As used in the CIP Standards, a Responsible Entity refers to the registered entities subject to the CIP Standards.

[3] As stated in the November 5, 2015, Compliance Guidance Policy: "Implementation Guidance provides a means for registered entities to develop examples or approaches to illustrate how registered entities could comply with a standard [footnote omitted] that are vetted by industry and endorsed by the ERO Enterprise. The examples provided in the Implementation Guidance are not exclusive, as there are likely other methods for implementing a standard. The ERO Enterprise's endorsement of an example means the ERO Enterprise CMEP staff will give these examples deference when conducting compliance monitoring activities. Registered entities can rely upon the example and be reasonably assured that compliance requirements will be met with the understanding that compliance determinations depend on facts, circumstances, and system configurations." *Available at:* http://www.nerc.com/pa/comp/Resources/ResourcesDL/Compliance_Guidance_Policy_FINAL_Board_Accepted_Nov_5_2015.pdf

CIP-010-3 Requirement R1

R1. *Each Responsible Entity shall implement one or more documented process(es) that collectively include each of the applicable requirement parts in CIP-010-3 Table R1 – Configuration Change Management. [Violation Risk Factor: Medium] [Time Horizon: Operations Planning].*

General Considerations for R1

As part of the Responsible Entity's documented processes for configuration change management, Requirement R1 now includes Part 1.6 which adds required steps for verifying software authenticity and integrity prior to installation in BES Cyber Systems. The objectives at a high level are to:

- Verify software authenticity to ensure that the software being installed in the BES Cyber System is from a legitimate source.
- Verify software integrity to ensure that the software being installed in the BES Cyber System has not been modified from its original obtained source.

Implementation Guidance for Requirement R1

The number of process(es) and their content may depend on a Responsible Entity's management structure and operating conditions. The Responsible Entity may document as many processes as necessary to meet its needs.

Requirement Part 1.6

CIP-010-3 Table R1 – Configuration Change Management			
Part	**Applicable Systems**	**Requirements**	**Measures**
1.6	High Impact BES Cyber Systems Medium Impact BES Cyber Systems Note: Implementation does not require the Responsible Entity to renegotiate or abrogate existing contracts (including amendments to master agreements and purchase orders). Additionally, the following issues are beyond the scope of Part 1.6: (1) the actual terms and conditions of a procurement contract; and (2) vendor performance and adherence to a contract.	Prior to a change that deviates from the existing baseline configuration associated with baseline items in Parts 1.1.1, 1.1.2, and 1.1.5, and when the method to do so is available to the Responsible Entity from the software source: 1.6.1. Verify the identity of the software source; and 1.6.2. Verify the integrity of the software obtained from the software source.	An example of evidence may include, but is not limited to a change request record that demonstrates the verification of identity of the software source and integrity of the software was performed prior to the baseline change or a process which documents the mechanisms in place that would automatically ensure the identity of the software source and integrity of the software.

Implementation Guidance for Requirement Part 1.6

The concept of software verification (verifying the identity of the software source and the integrity of the software obtained from the software source) is a key control in preventing the introduction of malware or counterfeit software. This objective is intended to reduce the likelihood that an attacker could exploit legitimate vendor software download or patch management processes to deliver compromised software updates or patches to a BES Cyber System. The software source refers in many but not all cases to the software source the Responsible Entity documented in CIP-007 R2 Parts 2.1 and 2.2. This requirement handles the situation where security patches can come from an original source (such as an operating system vendor), but must be approved or certified by another source (such as a control system vendor for support or contractual reasons) before they can be assessed and applied in order to not jeopardize the availability or integrity of a control system. The Responsible Entity, at its discretion, may use the same source as it documented in CIP-007 R2, Parts 2.1 and 2.2 or a different source of its choosing, so long as the identity of the software source is verified pursuant to CIP-010-3 R1, Part 1.6 Element 1.6.1.

When a method to do so is available to the Responsible Entity from the software source, it must provide controls for verifying the designated software components of the baseline configuration that have vendor updates before those updates are installed. As outlined and referenced in the requirement part, these components are:

- Operating system(s) (including version) or firmware where no independent operating system exists; (R1 Part 1.1.1)
- Any commercially available or open-source application software (including version) intentionally installed (R1 Part 1.1.2)
- Any security patches applied (R1 Part 1.1.5)

It is important to note that this is not limited to only security patches. Additionally, it is important to note that this requirement only applies when there is a change that deviates from the existing baseline configuration. This requirement is not applicable for the commissioning of new BES Cyber Systems where there is no existing baseline configuration.

Implementation Guidance for Verifying the Identity of the Software Source

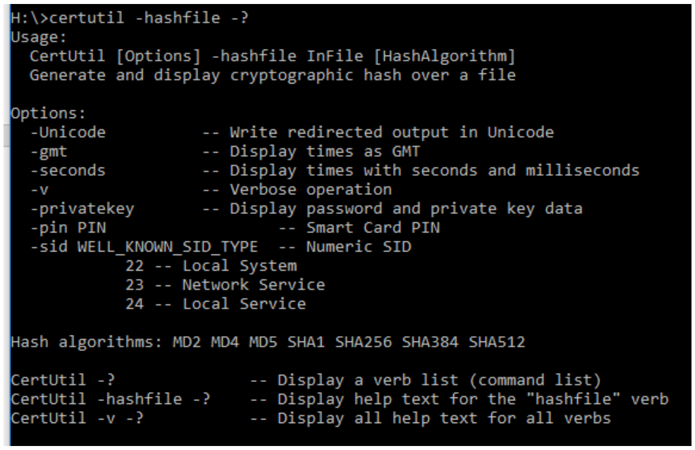
Below are some example approaches to comply with this requirement to verify the identity of the software source:

- Processes or procedural controls that require users to obtain software directly from the developer or vendor’s preferred delivery methods.
- Processes used to deliver software and appropriate control(s) that will verify the identity of the software source and the integrity of the software delivered through these processes. To the extent that the Responsible Entity uses automated systems such as a subscription service to download and distribute software including updates, software verification may likely be an automated byproduct.
- Use of SSL or HTTPS for downloading software updates such that Public Key Infrastructure (PKI) certificates are used for 3rd party verification from a Certificate Authority (CA) to verify the identity of the server hosting the updates.
- Use of a secure central software repository after the identity of the software source and the integrity of the software have been validated, so that verifications do not need to be performed repeatedly before each installation.

Implementation Guidance for Verifying the Integrity of the Software

Below are some example approaches to comply with this requirement to verify the intergrity of software obtained form the software source:

- Verify that the software has been digitally signed and validate the signature to ensure that the software’s integrity has not been compromised.
- Use public key infrastructure (PKI) with encryption to ensure that the software is not modified in transit by enabling only intended recipients to decrypt the software.
- If the vendor software source provides digital fingerprints or hash value for the software, verify the cryptographic hash values prior to installation on a BES Cyber System to ensure the integrity of the software. If provided by the vendor, consider using a method for receiving the verification hash values that is different from the method used to receive the software from the software source. For Windows, the certutil.exe utility is included in several versions and can be used to check the hash of a downloaded file.



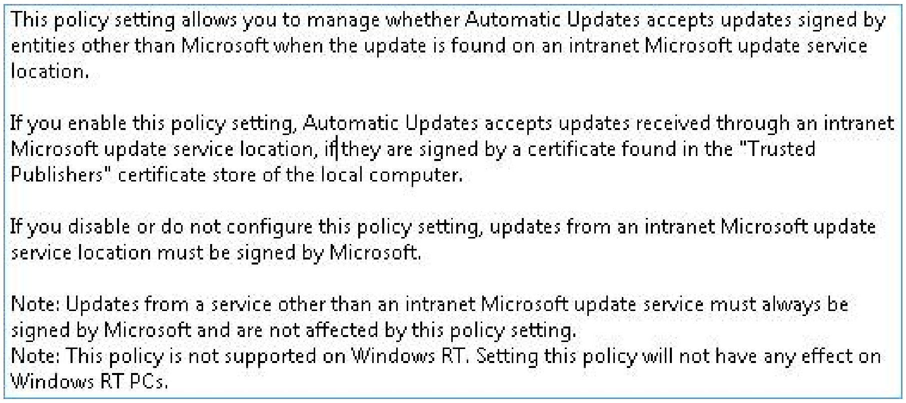
```
H:\>certutil -hashfile -?
Usage:
  CertUtil [Options] -hashfile InFile [HashAlgorithm]
  Generate and display cryptographic hash over a file

Options:
  -Unicode          -- Write redirected output in Unicode
  -gmt              -- Display times as GMT
  -seconds          -- Display times with seconds and milliseconds
  -v                -- Verbose operation
  -privatekey       -- Display password and private key data
  -pin PIN                  -- Smart Card PIN
  -sid WELL_KNOWN_SID_TYPE  -- Numeric SID
            22 -- Local System
            23 -- Network Service
            24 -- Local Service

Hash algorithms: MD2 MD4 MD5 SHA1 SHA256 SHA384 SHA512

CertUtil -?              -- Display a verb list (command list)
CertUtil -hashfile -?    -- Display help text for the "hashfile" verb
CertUtil -v -?           -- Display all help text for all verbs
```

- Use trusted/controlled distribution and delivery options to reduce supply chain risk (e.g., requiring tamper-evident packaging of software during shipping.)
- Use of a secure central software repository after the identity of the software source and the integrity of the software have been validated, so that verifications do not need to be performed repeatedly before each installation.

Implementation Guidance for Verifying the Identity of the Software Source and the Integrity of the Software with a Single Method

Some methods may complete both the verification of the identity of the software source and the verification of the integrity of the software obtained from the software source. Validation of digitally signed software is an example of a method that accomplishes both obligations required in CIP-010-3 Requirement 1, Part 1.6. Further, some processes may handle this in an automated fashion. One example of this is the Microsoft update process using Windows Server Update Services (WSUS) as described in the article found at the following link: https://technet.microsoft.com/en-us/library/cc708550%28v=ws.10%29.aspx?f=255&MSPPError=-2147217396. "Microsoft mitigates the risk of sending update files over an unencrypted channel by signing each update. In addition to signing each update, a hash is computed and sent with the metadata for each update. When an update is downloaded, WSUS checks the digital signature and hash. If the update has been tampered with, it is not installed." Other automated patching applications may also automate the validation of software source and software integrity.

Additionally, a Responsible Entity may demonstrate compliance with CIP-010-3 R1, Part 1.6 by using processes for software updates that technically enforce that only digitally signed software is installed. Microsoft Windows provides such a mechanism for Microsoft signed software through its default configuration and through a Group Policy Object (GPO) that expands this capability to other trusted third parties. This GPO is titled "Allow signed updates from an intranet Microsoft update service location" and a description of its functionality is included below:



This policy setting allows you to manage whether Automatic Updates accepts updates signed by entities other than Microsoft when the update is found on an intranet Microsoft update service location.

If you enable this policy setting, Automatic Updates accepts updates received through an intranet Microsoft update service location, if they are signed by a certificate found in the "Trusted Publishers" certificate store of the local computer.

If you disable or do not configure this policy setting, updates from an intranet Microsoft update service location must be signed by Microsoft.

Note: Updates from a service other than an intranet Microsoft update service must always be signed by Microsoft and are not affected by this policy setting.
Note: This policy is not supported on Windows RT. Setting this policy will not have any effect on Windows RT PCs.

It is possible to validate the software's digital signature using PowerShell ("Get-AuthenticodeSignature $path"). However, when a Responsible Entity is able to demonstrate that the process it uses to install or update software technically enforces this validation in an automated manner, it is not necessary to gather validation evidence for each and every software update and installation to demonstrate compliance with CIP-010-3 R1, Part 1.6.