Electric Reliability and Security for the West

Internal Controls
Failure Points and Guidance Questions
CIP-010-3 R1.6
January 2020

Security Objective

To verify the integrity and authenticity of software before installation in BES Cyber Systems. This will ensure that the software being installed was not modified without the supplier's knowledge and that it is not counterfeit.

NIST Special Publication 800-161 / FIPS-140-2, FIPS 180-4, / NIST Special Publication 800-53 (Rev. 4) CM-5(3), SI-7, SC-13

WECC Intent

The potential failure points and guidance questions give direction to registered entities for assessment of risk, while designing internal controls specific to NERC Reliability Standards and Requirements. The Registered Entity may use this document as a starting point in determining entity risk. It is not WECC's intent to establish a standard or baseline for entity risk assessment or controls design.

Note: *Guidance questions help an entity understand and document its controls. Any responses, including lack of affirmative feedback, will have no consequences on an entity's demonstration of compliance at audit.*

**Please send feedback to ICE@WECC.org with suggestions on potential failure points and guidance questions.*

Potential Failure Points & Guidance Questions

Potential Failure Point (R1.6): Failure to develop guidance detailing how changes will be managed.

1. How do you ensure processes work together to meet the R1.6 objectives?
   a. What existing processes will affect or strengthen R1.6?
   b. How do you coordinate software verification controls with other cybersecurity policies and controls, including change management, patching, and procurement?

Potential Failure Point (R1.6): Failure to define the type or category of changes that are considered deviations from a baseline configuration.

1. How do you ensure software verifications are done before a change that deviates from the existing baseline configuration?
2. Are these controls done at the same time as the 1.4.1 control?

Potential Failure Point (R1.6): Failure to document association of Parts 1.1.1, 1.1.2, and 1.1.5 of baselines to vendors of the CIP-013 plan.

1. How do you classify software for Parts 1.1.1, 1.1.2, and 1.1.5?
2. How do you differentiate between custom and commercially available software?

Potential Failure Point (R1.6): Failure to develop a process to discover methods available from software source providers of the CIP-013 plan.

1. How will you research or discover methods available from software source providers of the CIP-013 plan?

Potential Failure Point (R1.6.1): Failure to develop criteria for verifying the software source.

1. Once sources are identified, what criteria will you use to verify sources?
    a. How will you document evaluations?

Potential Failure Point (R1.6.1): Failure to develop a process describing how to verify software sources.

1. How did you establish methods to verify the identity of the software source?
    a. How will you document the verification?
    b. How will you monitor and handle vendor changes that may affect sources?
    c. How will you manage vendor obligations or contracts for software source service?

Potential Failure Point (R1.6.2): Failure to develop criteria for establishing software integrity.

1. What criteria will you use to establish integrity?
    a. How will you document evaluations?

Potential Failure Point (R1.6.2): Failure to develop a process detailing how to verify software integrity.

1. How did you establish methods to verify the integrity of the software?
2. How do you get software?
    a. Is it delivered through automated digital or physical means or manually retrieved?
    b. Does the software include a digital signature?
3. Is software encrypted?
    a. Does the software include cipher hashes?
4. Does the process provide guidance on when to do the verification?
    a. How do you document the verification?
    b. Does the process outline the verification options, i.e., per instance of use verification required? or an initial verification then periodic reverification?
    c. How will the process monitor vendor changes that might require reverification?
5. Do you use a secure central repository to store the software after it has been verified to avoid loss of integrity due to uncontrolled post-verification handling?
    a. How will you avoid repeated verifications?