Code Signing, Mitigation M1045 - Enterprise

Archived: 2026-04-02 11:25:36 UTC

Code Signing is a security process that ensures the authenticity and integrity of software by digitally signing executables, scripts, and other code artifacts. It prevents untrusted or malicious code from executing by verifying the digital signatures against trusted sources. Code signing protects against tampering, impersonation, and distribution of unauthorized or malicious software, forming a critical defense against supply chain and software exploitation attacks. This mitigation can be implemented through the following measures:

Enforce Signed Code Execution:

- Implementation: Configure operating systems (e.g., Windows with AppLocker or Linux with Secure Boot) to allow only signed code to execute.
- Use Case: Prevent the execution of malicious PowerShell scripts by requiring all scripts to be signed with a trusted certificate.

Vendor-Signed Driver Enforcement:

- Implementation: Enable kernel-mode code signing to ensure that only drivers signed by trusted vendors can be loaded.
- Use Case: A malicious driver attempting to modify system memory fails to load because it lacks a valid signature.

Certificate Revocation Management:

- Implementation: Use Online Certificate Status Protocol (OCSP) or Certificate Revocation Lists (CRLs) to block certificates associated with compromised or deprecated code.
- Use Case: A compromised certificate used to sign a malicious update is revoked, preventing further execution of the software.

Third-Party Software Verification:

- Implementation: Require software from external vendors to be signed with valid certificates before deployment.
- Use Case: An organization only deploys signed and verified third-party software to prevent supply chain attacks.

Script Integrity in CI/CD Pipelines:

- Implementation: Integrate code signing into CI/CD pipelines to sign and verify code artifacts before production release.
- Use Case: A software company ensures that all production builds are signed, preventing tampered builds from reaching customers.

Key Components of Code Signing

- Digital Signature Verification: Verifies the authenticity of code by ensuring it was signed by a trusted entity.
- Certificate Management: Uses Public Key Infrastructure (PKI) to manage signing certificates and revocation lists.
- Enforced Policy for Unsigned Code: Prevents the execution of unsigned or untrusted binaries and scripts.
- Hash Integrity Check: Confirms that code has not been altered since signing by comparing cryptographic hashes.
