Subvert Trust Controls: Code Signing Policy Modification, Sub-technique T1553.006 - Enterprise

Archived: 2026-04-05 12:41:34 UTC

Adversaries may modify code signing policies to enable execution of unsigned or self-signed code. Code signing provides a level of authenticity on a program from a developer and a guarantee that the program has not been tampered with. Security controls can include enforcement mechanisms to ensure that only valid, signed code can be run on an operating system.

Some of these security controls may be enabled by default, such as Driver Signature Enforcement (DSE) on Windows or System Integrity Protection (SIP) on macOS.[1][2] Other such controls may be disabled by default but are configurable through application controls, such as only allowing signed Dynamic-Link Libraries (DLLs) to execute on a system. Since it can be useful for developers to modify default signature enforcement policies during the development and testing of applications, disabling of these features may be possible with elevated permissions.[3][2]

Adversaries may modify code signing policies in a number of ways, including through use of command-line or GUI utilities, Modify Registry, rebooting the computer in a debug/recovery mode, or by altering the value of variables in kernel memory.[4][2][5][6] Examples of commands that can modify the code signing policy of a system include `bcdedit.exe -set TESTSIGNING ON` on Windows and `csrutil disable` on macOS.[4][2] Depending on the implementation, successful modification of a signing policy may require reboot of the compromised system. Additionally, some implementations can introduce visible artifacts for the user (ex: a watermark in the corner of the screen stating the system is in Test Mode). Adversaries may attempt to remove such artifacts.[7]

To gain access to kernel memory to modify variables related to signature checks, such as modifying `g_CiOptions` to disable Driver Signature Enforcement, adversaries may conduct Exploitation for Privilege Escalation using a signed, but vulnerable driver.[8][6]

Source: https://attack.mitre.org/techniques/T1553/006