# TTP Detail – T1553.006

## TTP Information

Name: Code Signing Policy Modification

Description: 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.(Citation: Microsoft DSE June 2017)(Citation: Apple Disable SIP) 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.(Citation: Microsoft Unsigned Driver Apr 2017)(Citation: Apple Disable SIP)  
  
Adversaries may modify code signing policies in a number of ways, including through use of command-line or GUI utilities, [Modify Registry](https://attack.mitre.org/techniques/T1112), rebooting the computer in a debug/recovery mode, or by altering the value of variables in kernel memory.(Citation: Microsoft TESTSIGNING Feb 2021)(Citation: Apple Disable SIP)(Citation: FireEye HIKIT Rootkit Part 2)(Citation: GitHub Turla Driver Loader) Examples of commands that can modify the code signing policy of a system include <code>bcdedit.exe -set TESTSIGNING ON</code> on Windows and <code>csrutil disable</code> on macOS.(Citation: Microsoft TESTSIGNING Feb 2021)(Citation: Apple Disable SIP) 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.(Citation: F-Secure BlackEnergy 2014)  
  
To gain access to kernel memory to modify variables related to signature checks, such as modifying <code>g\_CiOptions</code> to disable Driver Signature Enforcement, adversaries may conduct [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068) using a signed, but vulnerable driver.(Citation: Unit42 AcidBox June 2020)(Citation: GitHub Turla Driver Loader)

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Kill Chain Phases

**•** mitre-attack: defense-evasion

## Malware

* BlackEnergy
* Hikit
* Pandora

## APTs (Intrusion Sets)

* APT39
* Turla