NATIONAL SECURITY AGENCY
CYBERSECURITY INFORMATION

ENFORCE SIGNED SOFTWARE EXECUTION POLICIES

DEMAND AUTHENTICITY

Adversaries rely on specific techniques to install or run their own malicious code during network attacks. For example, following initial compromise of a system, an attacker will install and execute malware in memory or from disk. Sophisticated adversaries intent on long-term presence on a system will also install malware in early boot firmware [1], [2]. This boot firmware can survive operating system reinstallations, providing malicious actors access to the system for years. Other adversaries may simply alter legitimate programs and lure victims to download and install them [3].

Most modern platforms can enforce policies that control what software is allowed to execute on the system, countering these adversary techniques. These policies include validating cryptographic signatures on software at installation time, which limits software installations to those that are authentic and come from a trusted source. Doing the same at execution time provides protection against malware loaded directly into memory. Leveraging secure boot technologies prevents malware from persisting in early stages of the boot process, by limiting execution to signed code.

PLATFORM GUIDANCE

The capability to enforce signed software execution policies varies among Information Technology (IT) products. The following table and sections describe how several platforms currently implement these capabilities, while providing advice on how to activate them.

Platform	Secure Boot	Signatures required for OS* files and drivers	Application signatures required for installation	Application signatures required for execution
Windows®[1]	Available	Required	Available	Available
macOS®[2]	Available	Required	Available	Available
Linux®[3]	Available	Available	Available	Not Available
Android®[4]	Required	Required	Required**	Varies
iOS®[5]	Required	Required	Required	Required
3rd Party Apps	Not Applicable	Not Applicable	Varies	Varies

*Operating System (OS)
** Unless loading from "unknown sources" is enabled
See text below for specific details about versions and limitations.

Microsoft Windows

Microsoft Windows 10 provides the ability to restrict execution to only signed operating system and application software through the Windows Defender Application Control (WDAC) feature [4]. This includes the validation of digital signatures on kernel drivers to ensure their authenticity. Systems without WDAC only perform signature validation of drivers when they are loaded, and only on 64-bit versions of Windows 10.

[1] Windows is a registered trademark of Microsoft Corporation.

[2] MAC is a registered trademark of Apple Corporation.

[3] LINUX is a registered trademark of Linus Torvalds.

[4] Android a registered trademark of Google, Inc..

[5] iOS a registered trademark of Cisco Systems, Inc. in the United States and other countries and is used under license to Apple, Inc.

A related Microsoft AppLocker®[6] feature allows enterprises to create flexible policies to restrict or permit software execution based on digital signatures or other characteristics that may be necessary in some environments [5]. AppLocker allows for control over executable files, scripts, Windows Installer files, dynamic-link libraries (DLLs), packaged apps, and packaged app installers.

Windows 10 systems running on modern x86 platforms provide Universal Extensible Firmware Interface (UEFI) secure boot through the Secure Boot feature [6]. Secure Boot verifies the signatures on each piece of boot software when it is loaded at system startup, including UEFI firmware drivers, UEFI applications, and the operating system [7]. The specific platform technology features required by Secure Boot underscore the need to leverage modern hardware features[7].

Apple macOS

Apple macOS provides the ability to verify the signature of software packages when they are installed for system-wide use through the GateKeeper feature [8]. By default, only software signed by Apple or a certificate registered with Apple's developer program can be installed for system-wide use. Explicit administrator override is needed in order to install software that is not signed accordingly. The System Integrity Protection feature restricts the software that can be installed in certain locations of the system to only those signed by Apple [9]. It was first introduced in macOS 10.11, and apps in the Mac App Store work with System Integrity Protection. Newer Mac systems feature a T2 Security Chip [10]. When configured to do so, this enables the system to provide a secure boot feature that ensures only a signed operating system is loaded at system startup [11].

Linux

Linux distributions such as Red Hat®[8] Enterprise Linux are based on software package management systems that provide a structured format for the deployment of thousands of open source projects. These package management systems enable software makers to digitally sign the software they create. This then enables users to validate authenticity upon installation. Third-party software can also be packaged in this way and signed by its creator or by an enterprise so that it can be distributed in a trusted fashion [12]. Ensuring open source software is properly packaged and signed also aids in software inventory and update processes.

Enterprise-class Linux distributions also provide support for secure boot, to ensure that only an authentic Linux kernel and modules are loaded at boot time [13]. As with other platforms, this often must be activated in UEFI settings.

Apple iOS

Apple iOS requires that all executable code be signed using an Apple-issued certificate prior to its installation [14]. This includes the operating system software as well as third-party apps, which are available from Apple's App Store. When an app runs, iOS checks the code signatures of all executable memory pages as they are loaded to ensure that the software is authentic and has not been modified since it was installed. While there is an exception for developers enrolled within Apple's Developer Enterprise program, this additionally requires deployment of a specific Provisioning Profile to each device, and the app must still receive confirmation from Apple that the app is allowed to run.

All iOS devices enforce a secure boot chain that begins with Boot ROM, which cannot be altered after its manufacture. This Boot ROM verifies and loads the Low-Level Bootloader (on older devices) that then does the same for iBoot, which finally verifies and loads the kernel. All of these protections are enforced automatically as long as the device is not jailbroken.

[6] AppLocker is a registered trademark of Microsoft Corporation.

[7] For more information on these features, please refer to "Leverage Modern Hardware Security Features," part of the *NSA Cybersecurity Top 10 Mitigations.*

[8] Red Hat is a registered trademark of Red Hat, Inc.

Google Android

The Android platform provides mandatory signature verification of apps by default [15]. Verification occurs after the app is downloaded and prior to its installation. Although all applications must be signed, not all signatures are of equal trust. Enterprises should only install applications from trusted sources such as the Google Play Store. Never install applications from unknown sources and never side-load applications on devices intended for operational use. Side-loading refers to the practice of loading apps from computers via universal serial bus (USB) or from memory card storage, or from any source other than a trusted provider such as the Google Play Store. Generally, this can be avoided by disabling the option to “Enable Unknown Sources” for app loading.

Android’s Verified Boot feature ensures boot software comes from a trusted source such as device manufacturers [16]. This includes the bootloader, the boot partition, and partitions which include system software. The dm-verity feature verifies the integrity of other partitions that could potentially contain malicious software. The most recent versions of Android do not permit booting the system if these integrity checks fail. All of these protections are enforced automatically as long as the device is not rooted.

Third-Party Applications

Many types of software will also perform some level of cryptographic signature validation when downloading or installing software updates or other vendor-provided executable or security-related content (e.g. anti-virus DAT files). These implementations vary by vendor, but can be important to protecting the integrity of the software’s functions. These features are not always configurable, but enterprises and users should focus on acquiring applications that perform automatic software updates with these checks. Updates provided via trusted app store platforms automatically perform such checks.

Non-Traditional IT Platforms

Operating systems not traditionally associated with IT endpoints are increasingly adopting trusted booting and integrity protection methods, but like third-party applications, the techniques vary greatly. These platforms include routers, switches, firewalls, network proxies, and industrial control system (ICS) programmable logic controllers (PLCs). While not always easy to monitor, these devices typically change less often than their traditional IT counterparts and usually do not allow custom user applications, providing greater opportunities for monitoring and constraining the code allowed to run. Administrators should learn how to use the integrity monitoring features available to them on these devices.

PROTECT ACROSS THE LIFECYCLE

Major IT products include features to enforce the validation of digital signatures on software. This validation can occur when software is installed or executed, including during the earliest stages of system startup. While weaknesses may sometimes be discovered in the implementation or design of these features, major IT vendors act quickly to address these. These vendors are providing an environment in which it is increasingly difficult for malicious software to arrive and persist on a system. Sensible enterprises must leverage these features to address multiple stages of the adversary attack lifecycle.

REFERENCES

[1] A. Ionescu, Slide presentation, Topic: “Advancing the State of UEFI Boot Kits,” presented at OffensiveCon, Berlin, Germany, Feb. 16—17, 2018. Available: http://www.alex-ionescu.com/publications/OffensiveCon/offensive2018.pdf [Accessed 01 March, 2019]

[2] S. Gallagher, “First UEFI malware discovered in wild is laptop security software hijacked by Russians,” *Ars Technica*, 02 October, 2018. Available: https://arstechnica.com/information-technology/2018/10/first-uefi-malware-discovered-in-wild-is-laptop-security-software-hijacked-by-russians [Accessed 01 March 2019]

[3] A. Greenberg, “Software has a serious supply-chain security problem,” Wired, 18 September 2017. [Online] Available: https://www.wired.com/story/ccleaner-malware-supply-chain-software-security [Accessed 01 March 2019]

[4] J. Sutherland, et al. “Windows Defender Application Control,” *Microsoft,* 07 January, 2019. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/windows-defender-application-control [Accessed 01 March, 2019]

[5] J. Hall, L. Poggemeyer, “AppLocker,” *Microsoft,* 15 October, 2017. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-overview [Accessed 01 March, 2019]

[6] E. Graff, et al. “Secure boot,” *Microsoft*, 04 October, 2017. Available: https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot [Accessed 01 March, 2019]

[7] J. Hall, et al. “Secure the Windows 10 boot process,” *Microsoft,* 15 November, 2018. Available: https://docs.microsoft.com/en-us/windows/security/information-protection/secure-the-windows-10-boot-process [Accessed 01 March, 2019]

[8] Apple, “Safely open apps on your Mac,” 25 September, 2018. Available: https://support.apple.com/en-us/HT202491 [Accessed 01 March, 2019]

[9] Apple, “About System Integrity Protection on your Mac,” 07 November 2017. Available: https://support.apple.com/en-us/HT204899 [Accessed 01 March, 2019]

[10] Apple, “About the Apple T2 Security Chip,” 06 February, 2019. Available: https://support.apple.com/en-us/HT208862 [Accessed 01 March, 2019]

[11] Apple, “About Secure Boot,” 06 November, 2018. Available: https://support.apple.com/en-us/HT208330 [Accessed 01 March, 2019]

[12] Red Hat, “How to sign rpms with GPG,” 22 February 2018. Available: https://access.redhat.com/articles/3359321 [Accessed 01 March, 2019]

[13] M. Doleželová, et al, “Unified Extensible Firmware Interface (UEFI) Secure Boot,” in *System Administrator’s Guide: Deployment, Configuration, and Administration of Red Hat Enterprise Linux 7*, ch. 25.11, *Red Hat.* Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/sec-uefi_secure_boot [Accessed 01 March 2019]

[14] Apple, *iOS Security Guide,* November, 2018. Available: https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf [Accessed 01 March, 2019]

[15] Android Developers, “Sign your app,” *Android Studio User Guide.* Available: https://developer.android.com/studio/publish/app-signing [Accessed 01 March, 2019]

[16] Android Source, “Verified Boot,” *AOSP, Secure, Features.* Available: https://source.android.com/security/verifiedboot [Accessed 01 March 2019]

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

CONTACT INFORMATION

Client Requirements and General Cybersecurity Inquiries
Cybersecurity Requirements Center (CRC), 410-854-4200, email: Cybersecurity_Requests@nsa.gov