Foundation of Mac Security

by Steve Hultquist

In my first article, I offered you background for a perspective on the Mac and its security. In this article, let's look at the first layer and the foundations of securing Mac in the enterprise.

If you were to create a new desktop operating system today, what would be your approach to securing it? Would you design it so third-party software would be necessary to protect it, and require such software to run in the most vulnerable areas of the operating system kernel? Or would you create a system which limits access to operating system components and runtime memory to reduce the possible attack surface as much as possible? The latter is the approach Apple has built into macOS.

ROOT OF TRUST

In my experience, enterprise IT and security teams have difficulty shifting their paradigm from the traditional view of reactive third-party software designed to scan and remove malware and vulnerabilities, to Apple's unified, integrated approach in their design of the Mac. From the most fundamental layers of hardware through firmware, the operating system, applications, and finally data, Apple's integration allows for a "chain of trust" anchored in the Mac itself, securing dramatic limitations on attack surfaces and possible attack vectors.

Think about it this way: if trust is rooted in a cryptographic key integrated with the hardware, where can an attacker surreptitiously violate the chain of trust? The design of Mac hardware and macOS roots trust in the hardware and maintains a chain of trust through each step along the way of bringing the system alive. With macOS Big Sur, available autumn 2020, even the read-only system volume is cryptographically signed. So, on a modern Mac with a T2 security chip, trust for macOS secure boot begins with the T2 chip itself, and both the T2 and Secure Enclave on the T2 execute secure boot processes.

From those early boot processes, the Unified Extensible Firmware Interface (UEFI) firmware trust is rooted in the T2 chip. UEFI firmware updates are digitally signed by Apple and verified by the firmware before updating the storage. To achieve a persistent UEFI firmware infection, an attacker would need to achieve a persistent T2 firmware infection.

FULL SECURITY

As noted in Apple's Platform Security document, Full Security is the default and it behaves like iOS and iPadOS. At the time that software is downloaded and prepared to install, rather than using the global signature which comes with the software, macOS talks to the same Apple signing server used for iOS and iPadOS and requests a fresh, "personalized" signature. A signature is said to be personalized when it includes the ECID—a unique ID specific to the T2 chip in this case—as part of the signing request. The signature which is given back by the signing server is then unique and usable only by that particular T2 chip. When the Full Security policy is in effect, the UEFI firmware ensures that a given signature isn't just signed by Apple, but is signed for this specific Mac, essentially tying that version of macOS to that Mac.

In addition to this protection, the system partition is protected by System Integrity Protection, restricting components to read-only in specific critical file system locations to prevent malicious code from modifying them. Apple says, "macOS Big Sur introduces a cryptographically signed system volume that protects against malicious tampering. It also means that your Mac knows the exact layout of your system volume, allowing it to begin software updates in the background while you work."

This combination of hardware and software cryptography and chain of trust means a dramatically increased challenge to attacks against the system and system software.

In addition to protecting contents and file-system permissions of system files and directories, System Integrity Protection also protects processes against code injection and runtime attachment (like debugging) and DTrace, and protects against unsigned kernel extensions ("kexts").

ROGUE SOFTWARE PROTECTION

Once the system itself is protected in this way, the next consideration is protection against rogue software. Again, consider whether it is more effective to scan software at install time, scan the system for known malware, and attempt to remove malware once installed, or to block any attempt to install software that isn't known to be authenticated and reviewed.

Apple chose the latter. Apple's GateKeeper software ensures that software installation only occurs when the software is signed by a registered developer and notarized by Apple. Notarization by Apple is an automated system which scans software for malicious content and checks for code-signing issues. Apple will even cause apps from a given developer to stop launching if the developer is found to be malicious.

In the unlikely event that malware makes it past these protections, there are more hurdles. Apple's XProtect system runs a signature-based malware scan against apps when they are launched. On top of XProtect, the MRT (Malware Removal Tool) is available in the unlikely event an installed piece of malware needs to be removed.

Does this mean that macOS is 100% impenetrable? Of course not! Nothing is perfectly protected. However, the native protections are so effective at blocking access to the majority of malware attacks that additional protections must be designed and architected to work with the native protections.

Next time, we'll take a more in-depth look at what it means to work with those protections.

ABOUT THE AUTHOR

Steve Hultquist
Solutions Architect for Wipro - Apple Devices Services

Steve works closely with clients and Apple to create effective, productive, and innovative solutions deploying Apple in the enterprise. He has held positions of CIO and CTO for multiple organizations, creating choice programs at each of them. He has also written a CIO column for InfoWorld, been an enterprise management analyst, and a hands-on engineer and executive building networks and connected systems worldwide.

He can be reached at stephen.hultquist@wipro.com.