THE FIRMWARE FACE OF FILELESS THREATS

Attackers are always evolving in order to evade traditional security controls, and in recent years, fileless threats have become one of the most popular attacker strategies. Fileless threats have been around for many years, but have recently made a resurgence in the wild. Unlike traditional malware, fileless threats don't exist as a file that resides on a system's disk. Instead the malicious code may only exist in memory, be run as remote scripts, or run in areas outside of the disk and beyond the view of traditional security tools.

WHY GO FILELESS?

This strategy can make the threat much harder for security and forensics tools to detect and analyze. For example, if the threat doesn't exist on a device's disk, the threat can avoid some of the traditional file-based analysis of an antivirus tool. Likewise, if the threat only manifests itself at runtime, security teams have a very limited window in which to analyze the threat. Attackers can also avoid using traditional files by leveraging valid administration tools such as PowerShell and WMI, which can be used to remotely run scripts or other malicious code on a victim device.

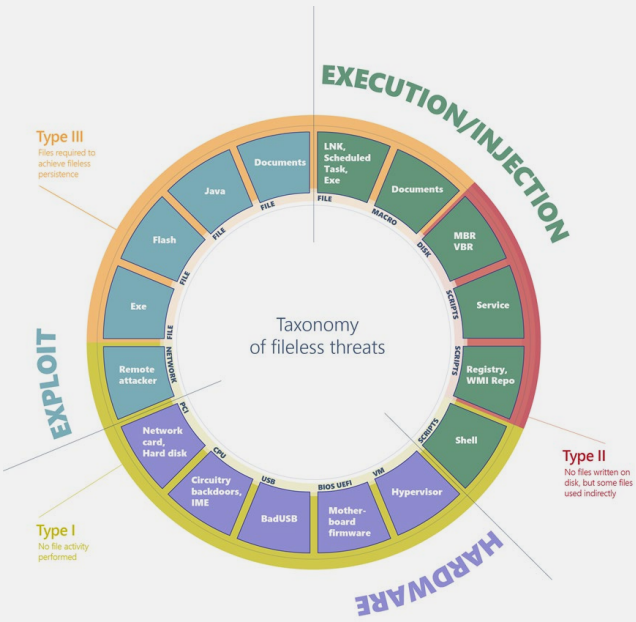
As it turns out, fileless threats also apply to the often overlooked firmware layer of a device. And since the whole purpose of going fileless in the first place is to avoid traditional security controls, the firmware layer is a natural location for attackers to hide their code. This underscores an important and fundamental aspect of security—organizations need to develop the same level of visibility and security at the firmware level that they have traditionally had at the operating system layer and for files that reside on hard drives. With this in mind, let's take a look at the intersection between fileless threats and firmware.

THE FILELESS ATTACK SURFACE

While the definition of a fileless threat is open to some debate, Microsoft provides a very helpful framework for understanding and classifying these threats. The diagram below taken from Microsoft's site breaks down fileless threats in two important ways:

How Malicious Code is Run: This includes code run from hardware, via exploits, and via code injection. All of these options allow attackers to execute code without necessarily relying on traditional file.

How the Threat Relies on Files: While fileless threats only run in memory, they may still depend on files that reside on disk. For example, some threats can be completely fileless, while others may run in memory but only with help from more traditional files that reside on the disk.



Source: Microsoft

This diagram provides two very important take-aways. First, hardware components represent a significant part of the logical attack surface of fileless threats. This includes firmware sources such as system BIOS/UEFI, CPUs, PCI devices, and USB. These are all examples where malicious code can reside outside of the system disk and typically beyond the view of the operating system. In reality this is actually an abbreviated list of the hardware attack surface, and we encourage you to refer to our Know Your Own Device resource to learn more about the many threats and vulnerabilities affecting other components in typical devices.

Secondly, the true fileless category of attacks (Type I in the Microsoft model) is heavily tied to the hardware attack surface. In other words, fileless attacks from firmware sources typically don't require support from other files. This means they leave the fewest traces and can be the hardest to detect by traditional means.

Unfortunately, this style of attacks against firmware is becoming more common. Broad-based malware campaigns targeting the firmware layer of devices have been observed in the wild, and earlier this month outdated firmware was used in a denial of service attack against the US power grid. Likewise, our own recent research demonstrated vulnerabilities that could allow unsophisticated attackers to gain control over the BMC firmware of enterprise servers. This means that the bar for attackers is lowering precisely in the area that is the most ripe for truly fileless threats.

FIRMWARE AND LIVING OFF THE LAND

Fileless threats are also often defined as threats that take advantage of valid tools on the system to perform malicious actions. This is often referred to as 'living off the land" since the threat uses native tools on the system instead of bringing its own malware. This includes tools such as PowerShell, WMI, PsExec and other SysInternals tools. The LOLBAS (living off the land binaries and scripts) project has compiled a list of the many tools used by attackers to live off the land. Most organizations rely on these tools in order to efficiently manage their systems. Unfortunately the power of these tools is equally valuable to attackers, who can abuse the functionality to run malicious scripts or install malicious code. And while WMI can install malicious files that reside on the disk, they are stored in a shared repository making it almost impossible to delete them without damaging valid data.

BMCs and IPMI

The firmware layer contains tools that can play a somewhat similar role for attackers. Within modern servers, the combination of BMCs and IPMI provides administrators with complete remote control over a server. The BMC essentially acts a parallel and independent computer within a server solely for the purpose of remote management. It contains its own firmware, networking capabilities, and even its own power in order to provide management even if the server itself is powered off.

In recent years, BMCs have also been one of the most common sources of

vulnerabilities. Vulnerabilities within the BMC can allow attackers to install their own BMC firmware containing malicious code and gain virtually unlimited power to do damage. As with all firmware threats such code could be used to achieve attacker persistence, to steal data, or to disable devices or components completely.

Intel AMT and ME

These hardware-based management channels are not limited to servers. Intel Active Management Technology (AMT) and the Management Engine (ME) provide similar out-of-band management capabilities for traditional laptops. These components likewise have their own communication channels and have been used by attackers to communicate without the operating system's knowledge.

These tools can be used to deliver code to low-level components and control the behavior of the operating system itself. Much like the BMC of a server, Intel AMT can provide the plumbing for an attacker to deliver malicious code that hides beneath the operating system and without touching the filesystem.

And these are not the only examples. LoJack functionality which resides in a device's firmware is designed to help track and remotely wipe a device in case of theft. This functionality has been compromised by attackers and used as a backdoor and command-and-control channel. Similarly, our research has shown that the very kernel drivers used to manage firmware can be used by attackers as a vector to infect the firmware. Such drivers are often used to update the firmware, set firmware-specific options, or diagnose problems. But in the wrong hands they can provide a natural vector to deliver malicious code that will never touch the disk of an affected system. This once again shows how low-level tools built in a devices firmware can wreak havoc if not properly secured.

ADDRESS YOUR BLINDSPOTS

These types of threats highlight the need to extend security to the many layers where traditional security can't see. It is important to remember that the whole reason attackers go fileless is to avoid the prying eyes of security solutions. As such, it should be no surprise that firmware and fileless threats overlap a great deal. Going forward, organizations need to establish visibility into the hardware and firmware layer to detect vulnerabilities, weaknesses, and threats therein.

And since, fileless attacks are rapidly evolving, it is important to recognize threats that are new and may be unknown to the industry. This means that not only do we need to be monitoring the hardware and firmware of our devices, we also need to be monitoring the behavior of these components to recognize unknown or zero-day threats.

At Eclypsium we specialize in the unique vulnerabilities and threats affecting this layer, and provide an approach to defending against fileless threats that security tools at the operating system level simply can't reach. To learn more about Eclypsium and how we can extend your security strategy to the firmware layer, please contact us at info@eclypsium.com.