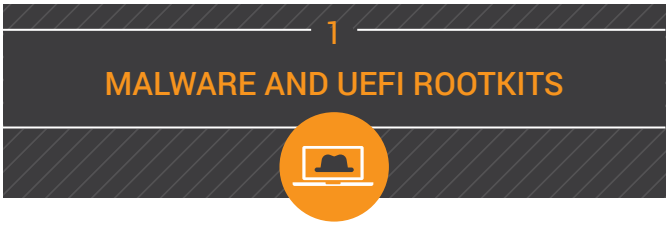
UNDERSTANDING THE TOP 5 COMMON FIRMWARE AND HARDWARE ATTACK VECTORS

As firmware-level threats continue to gain traction in the wild, security teams need to quickly get up to speed on how these threats work and how their devices can be targeted and attacked. In this paper we demystify the most common types of firmware threats today and analyze their path into an organization.

INTRODUCTION

For many years, firmware-level vulnerabilities and threats have largely been out of sight, out of mind for many security teams. However recent changes to the threat landscape are bringing those days to a close. The Meltdown and Spectre vulnerabilities introduced the world to the power of hardware-level weaknesses, LoJax malware brought UEFI rootkits into the wild, and US-CERT alerted the industry to widespread Russian-backed attacks targeting network infrastructure. The ability for attackers to compromise device firmware remotely, while users are traveling with their laptops, and even in the hardware supply chain itself, makes firmware security uniquely challenging.



And while these types of attacks may be new to many security teams, they are very real and are quickly becoming a top priority for regulatory bodies, industry analysts, and virtually any organization that faces advanced attackers. This paper provides an introduction to the most common firmware and hardware attack vectors, and what you need to know in order to start defending yourself.

Attacks against firmware are especially appealing to attackers. Once they have compromised the firmware, they can safely persist on the device and evade your OS, application or software levels of security. Since the malicious code lives within the firmware of physical components, the threat can easily survive a complete re-imaging of the system or even replacement of the hard drive(s).

This sort of persistent attack would typically occur as a second stage of malware infection. Once a system is initially compromised, malware could then look for vulnerabilities in the firmware and missing device protections that could allow malicious code to be implanted in the firmware itself. Two of the most well-known examples from the real-world are the recently discovered LoJax malware and the infamous Hacking Team UEFI Rootkit.

In both of these examples, the malware targeted the system's UEFI firmware. These attacks took advantage of specific vulnerabilities and many other vulnerabilities have been discovered over the past few years in UEFI and related components. However, in some cases attackers don't need to exploit a vulnerability at all in order to install their malicious implants. Older systems and even some recent servers lack basic protections like signed firmware updates. These attacks can apply to virtually any device that can be compromised with malware. As a result, it is imperative that organizations have the tools to find firmware vulnerabilities, missing protections, and both known and unknown implants in order to secure enterprise devices.

2 EXPLOITING FIRMWARE REMOTELY

While malware represents a common attack vector, research has shown that firmware can also be exploited remotely. This attack vector has a lot to do with the growing set of networking options found within UEFI components themselves. In short, the standard UEFI codebase now includes a rich set of network capabilities for Ethernet, WiFi, and even Bluetooth that allow the firmware to communicate remotely and even perform a full HTTP boot from a remote server across the internet. Additionally, vendors have individually been implementing "update over the Internet" features that allow the computer to check for and download firmware updates from a remote server before the host operating system is ever loaded.

Eclypsium researchers found that in some cases the update over the Internet functionality was downloaded unverified and in the clear. The host would try to contact a remote update server using plain HTTP without SSL or any verification. This means that simple man-in-the-middle or otherwise redirection techniques (e.g. DNS/ARP/route poisoning) could be used to modify the response returned to the client and exploit the vulnerability. As a result, our research showed that we could remotely deliver malicious code resulting in buffer overflows and arbitrary code execution just by checking if a newer version of the firmware exists.

The important point to remember is that with network connectivity comes the opportunity for remote exploitability - just as with every other connected device and component in history. Networking has become a standard part of firmware and certainly doesn't show signs of going away. This means that whether from malware or remote exploits, an organization's firmware vulnerabilities are increasingly accessible to attackers.

3 PHYSICAL TAMPERING

Devices are also often open to compromise by an attacker who has physical access to the machine. These threats have been dubbed "Evil Maid" attacks referring to a scenario where a malicious hotel room maid can install a rootkit on a laptop left in the room. This sort of attack is especially relevant to organizations whose employees travel. Any time their laptop is out of the user's control, an attacker can potentially compromise the device at the firmware level without opening the case. This particular attack vector is often tied to the presence of hardware and device debug mechanisms. Debug mechanisms are standard components that assist in tracing the source of faults in virtually all platforms. These mechanisms are primarily used before a platform reaches production, but also are often used for refurbishing and fixing returned platforms. Security researchers have repeatedly published attacks using debug features. Eclypsium research confirmed that debug access over USB enables installation of persistent rootkits in UEFI firmware and runtime SMM firmware on systems that do not securely set debug policy (CVE-2018-3652).

While this may seem like it requires specialized equipment and detailed knowledge, it is actually quite easy in most cases. Most firmware is stored on a Serial Programmable Interface (SPI) flash chip. This creates a physical standard for reads and writes to the storage chip, and SPI flash programmers are relatively easy to buy or create. Other researchers have developed a generic proof-of-concept backdoor that can be easily installed into most firmware modules. One could say that the ease and availability of these tools and techniques make firmware rootkits accessible to non-experts or "script kiddies". Using these techniques we have demonstrated the ability to install a rootkit on an enterprise laptop with no more than 4 minutes of physical access.

4 MANAGEMENT BACKDOORS

While all enterprise devices have a hardware/firmware attack surface, the problem goes even deeper when it comes to remote management. High value servers and even modern laptops are designed to be supported via remote out-of-band management. In the case of servers this is often the role the baseboard management controller (BMC) and standards such as IPMI play. This includes the ability to monitor and manage everything about the server including the ability to update firmware, change the host operating system, and boot or cycle the host. And since no one wants to physically run to a server and cable everytime there is an issue, these functions are available over the network, often on its own dedicated management subnet.

While IPMI and BMCs serve a critical need for data centers, they also present a massive risk. A remotely accessible interface that can control most aspects of the host even when the host is powered off would be an obvious goldmine for attackers. Unfortunately, this critical area is often overlooked when it comes to security. BMCs often use default, widely-known passwords, and administrative access is often not logged. An attacker who gained access to the server via IPMI, could easily gain full control over the power of the data center. Attacks against BMCs have been a particularly busy area of research in the industry, with a variety of new weaknesses being disclosed.

However, the same issues exist for laptops as well. Intel's Active Management Technology (AMT) provides for out of band management for laptops. This functionality provides remote management for laptops in much the same way that IPMI supports servers. Malware has already seized on this functionality for a variety of communication and evasion techniques.

5 SUPPLY CHAIN ATTACKS

Most of our examples thus far have consisted of attackers compromising a deployed, active system. However, devices can be modified in the supply chain before they are ever unboxed by the eventual owner. This type of attack can be incredibly difficult for most organizations to detect given that even the earliest baseline state of the device is already compromised. Recent reports have debated to what degree this style of attack has been successful. However, what is not up for debate is that security leaders, analysts, and governments have made supply chain security a top priority.

NIST recently updated its Framework for Improving Critical Infrastructure Cybersecurity to include a Supply Chain Risk Management (SCRM) category, while greatly improving the guidance related to SCRM throughout the framework. Likewise, the UK's NCSC Cyber Threat to UK Business Report highlighted the recent increase in supply chain attacks as a major area of focus moving forward. Additionally, in Gartner's recent Top 6 Security and Risk Management Trends for 2018, the firm highlighted the importance of "origin over pricing" when evaluating technology purchases and the need to carefully consider the upstream and downstream relationships of all technology suppliers.

Needless to say, security teams will need new tools and safeguards when devices can potentially be compromised "out-of-the-box". This puts greater importance on being able to ensure that all firmware is valid and hasn't been tampered with.

CONCLUSIONS

This paper serves as an introduction to firmware-level threats and high-level approaches that attackers can use to target your infrastructure. Whether servers, networking gear, or personal laptops, firmware is increasingly a target. Building security countermeasures to defend and mitigate these threats is essential, and Eclypsium is purpose-built for this task. However, we also encourage organizations to use the open-source CHIPSEC project to begin looking for firmware issues in your devices today. With the many avenues attackers have to your enterprise firmware, it is critical that firmware security controls become a standard part of your security operations. If you would like to learn more about Eclypsium and our products, please reach out to us at info@eclypsium.com.