WEAK LINKS: FIRMWARE SECURITY FOR ICT SUPPLY CHAINS

INTRODUCTION

Every piece of information and communications technology (ICT) equipment in an organization today is an amalgamation of chips, components, and code from a convoluted chain of vendors and suppliers. Any link in this complex technology supply chain can introduce threats or vulnerabilities that can undermine the security of the asset and the entire acquiring organization. Managing these inherited risks forces IT and Security Teams to look in what is often unfamiliar territory – the firmware ecosystem.

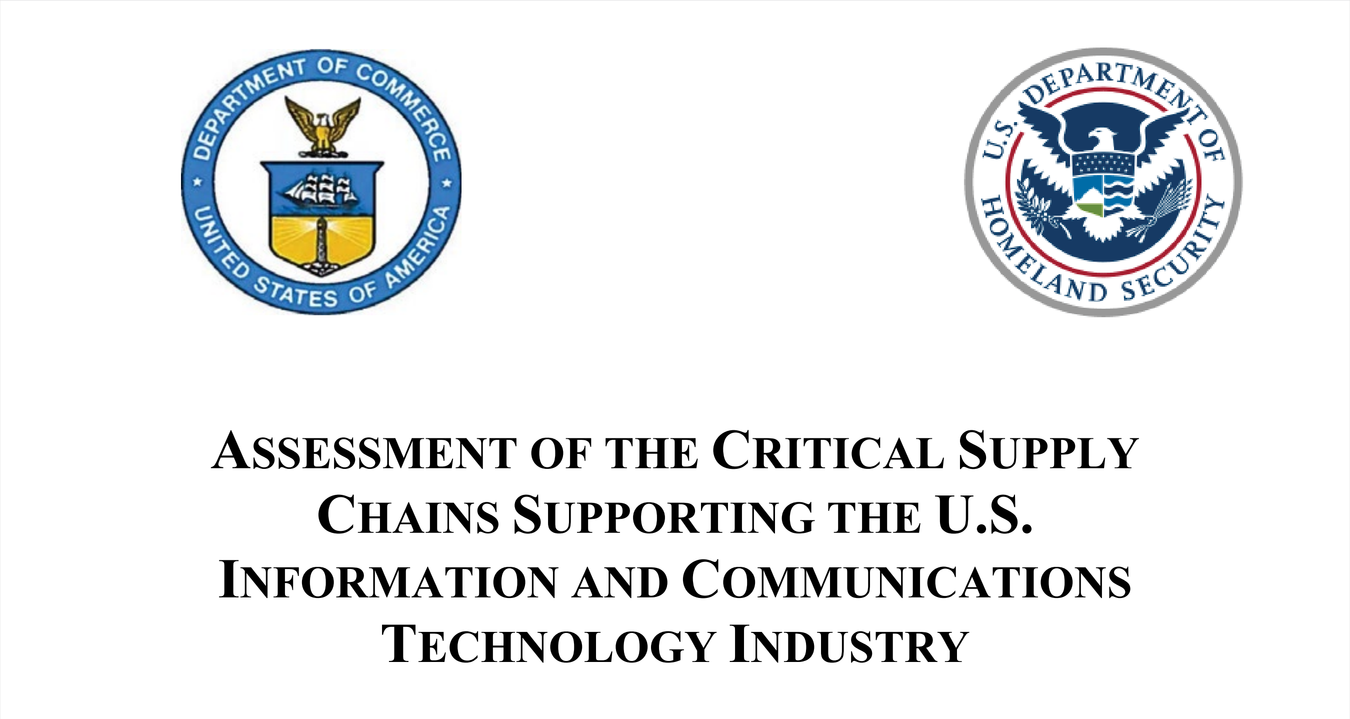
Device firmware is critical software embedded in devices of all kinds. A February 2022 joint report from the United States Departments of Commerce and Homeland Security – "Assessment of the Critical Supply Chains Supporting the U.S. Information and Communications Technology Industry" – explicitly called out the need for the security of firmware in supply chains. A Security Week article summarized the report in a headline that ran "U.S. Government Issues Stark Warning, Calling Firmware Security a 'Single Point of Failure.'"

SECURITYWEEK
CYBERSECURITY NEWS, INSIGHTS & ANALYSIS

U.S. Government Issues Stark Warning, Calling Firmware Security a 'Single Point of Failure.'

The report itself, at the conclusion of a section on supply chain risks, called out a clear alarm:

> **"Firmware presents a large and ever-expanding attack surface as the number of electronic devices grows and the ICT supply chain increases in complexity."**



ASSESSMENT OF THE CRITICAL SUPPLY CHAINS SUPPORTING THE U.S. INFORMATION AND COMMUNICATIONS TECHNOLOGY INDUSTRY

Report from U.S. Dept of Commerce and U.S. Dept of Homeland Security dated February 2022

Why this focus? Virtually all other technologies ultimately depend on physical equipment, whether in the form of an end-user laptop, networking device, or server either locally or in the cloud. And within all of these devices, firmware is pervasive. It's the persistent logic that governs how a device or physical component actually behaves. If devices are compromised at this level, all higher-layer data, applications, and services on the device are put at risk.

When it comes to supply chain risk, the tight integration and dependencies of firmware and hardware create a near-perfect storm. Physical devices have by far the deepest supply chains spread across many countries of origin - creating many opportunities for mistakes or compromises by external adversaries or malicious insiders. Firmware packages are often reused by many suppliers, allowing any vulnerabilities to be passed on to countless products. Like software, firmware often requires regular updates to prevent the supply chain from being compromised even after a device is deployed. And all of these risks are multiplied by the number of vendors and device types which will all have their own unique supply chain and risks.

To mitigate this risk, organizations must be able to independently verify the authenticity, integrity, and posture of their devices and components down to the firmware level. Only by getting to the firmware can an organization truly verify that their technology truly is what it appears to be. While this may require new processes and tools, it is an integral part of supply chain risk management.

This document provides detailed insight into the firmware and hardware side of supply chain security including elements of Supply Chain Risk Management (SCRM) and Supply Chain Assurance. This will cover a variety of interrelated topics including:

- Real-World Supply Chain Attacks and Trends
- Where Firmware Fits in the Big Picture of Supply Chain Risk
- The Role of Firmware in Supply Chain Attacks
- Practices for Firmware Supply Chain Security

REAL-WORLD SUPPLY CHAIN ATTACKS AND TRENDS

Supply chain attacks have grown markedly over the past years, affecting government agencies as well as private sector enterprises. A 2021 Crowdstrike study found that 45% of organizations experienced at least one supply chain attack in the past 12 months, while security firm, Sonatype, reported that supply chain attacks grew by 430%.

This rise in supply chain attacks can be attributed to a variety of factors. First, as organizations have improved their internal cybersecurity practices, a supply chain attack lets attackers shift their focus to softer upstream targets. This not only lets attackers compromise devices and code before it is delivered to the target organization, but it also takes advantage of any inherent trust that an organization has for its technology partners. Supply chain attacks also provide a "one-to-many" infection vector, in which a single compromise could be spread to hundreds or thousands of downstream organizations. Furthermore, the Covid-19 pandemic has introduced even more fluidity into supply chains as OEMs and technology providers have been forced to adapt to global shortages and supply chain disruptions. This has created even more opportunities for attackers to insert themselves into the supply chain.

While supply chain attacks have become more common, they have been well-known for several years and most commonly have been associated with sophisticated attackers. The Breaking Trust project has maintained a centralized record of 139 publicly disclosed supply chain attacks from the past several years. Some of the most well-known examples include:

- **SolarWinds SUNBURST** - Sophisticated attackers were able to infiltrate SolarWinds and implant malicious code within the source code of SolarWind's IT Monitoring and Management platform known as Orion. The compromised code was subsequently published and delivered to more than 18,000 federal and private sector customers including a variety of leading technology companies. By infecting IT management tools, attackers had the opportunity to compromise a wide range of servers and assets including the ability to update device firmware.

- **ASUS ShadowHammer** - Attackers were able to modify the ASUS Live Update Utility to deliver properly signed, yet malicious updates to more than 57,000 users. The LiveUpdate Utility is particularly significant as it provides BIOS and UEFI updates in addition to software updates, allowing attackers to potentially deliver malicious system firmware.

- **CCleaner** - Attackers were able to add a backdoor to the popular CCleaner utility from security firm, Avast. By compromising the build environment used to create the software, attackers were able to insert malicious code that was ultimately delivered to more the 2.2 Million users.

These are only a few of the many supply chain attacks observed in the wild. However, they all highlight how the supply chain can provide a trusted vector into some of the most critical areas and assets within an enterprise.

FIRMWARE IN THE BIG PICTURE OF SUPPLY CHAIN RISK

Technology supply chains come in many forms, but from an acquiring organization's point of view, we can break supply chain risk into three fundamental sources - equipment, applications, and services. All three of these categories have been used as real-world threat vectors. Likewise, all three have the potential to introduce supply chain risk when the technology is initially acquired as well as its ongoing operational lifecycle.

When we look at an organization's equipment and devices, virtually all of the long-term risk is tied to firmware. Additionally, many of the underlying factors that lead to supply chain risk are magnified when we look at the firmware in physical equipment and devices. For example, physical devices have the deepest supply chains spread across the most countries of origin. Code reuse is common at the firmware level, which can lead to extremely widespread issues when new vulnerabilities are found. And naturally, equipment and their firmware are some of the most broadly deployed and high-value assets in an environment, making them also one of the highest impact classes of technology.

When we discuss the varying risk factors and their drivers, it's tempting to see it all in the light of software: "It's all the same issue – it's all code of one kind or another. How is firmware any different?" But for the purposes of this paper we'll take the stance that it is different, especially in the sense that defenders are gaining skill in remediating and hardening applications and OSes, but that attackers are disproportionately better versed in weaponizing firmware than security teams are at defending it.

With that perspective in mind, Let's take a closer look at how the three major categories of the technology supply chain compare.

Risk Factors	Equipment	Applications	Services
Supply Chain Depth	Very High	Medium	Low
Country of Origin	High	Low	Low/Medium
Number of Vendors	Medium/High	High	Low
Code Reuse	High	High	Low
Target Organization Impact	Very High	High	Very High

Eclypsium's assessment of risk levels across different supply chain layers.

Managed Service

Managed service providers regularly need privileged access into a customer environment in order to provide their services. If the service provider is compromised, attackers can abuse this position of trust to then compromise the provider's downstream customers. This strategy has been particularly popular with ransomware gangs as seen in the well-known Kesaya attacks in which attackers used the services firm to infect up to 1,500 businesses with ransomware. These attacks can have a very broad scope and impact to an organization since a service provider may have access to highly sensitive systems and infrastructure with the customer organization.

The only good news is that most organizations will have a very limited number of providers with this level of trusted access to their technology and infrastructure. Service providers are also fairly self-contained and directly provide services themselves instead of farming them out to other vendors. These traits can limit the opportunities for an attacker to gain access to service provider.

Applications and Software

3rd-party applications and software naturally account for a significant portion of an organization's technology and likewise its supply chain risk. One of the most notable issues with application supply chain risks is that most organizations will do business with many application vendors. Any one of these vendors could be compromised by external adversaries, malicious insiders, or can introduce known or unknown vulnerabilities in their code. The other most common source of application supply chain risks stems from the reuse of open source software and projects. Any vulnerabilities in these open source libraries can be passed on to the vendor's applications and lead to widespread exposures.

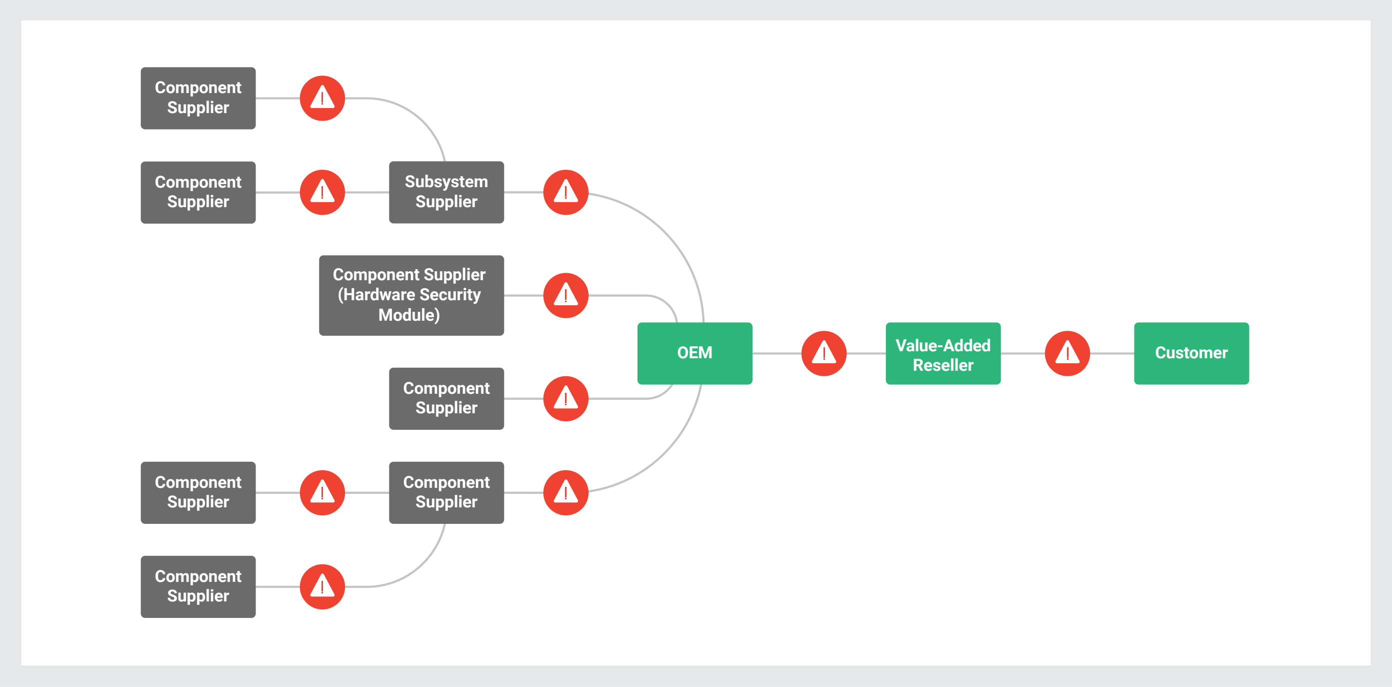
However, outside of open source projects, most application supply chains are not particularly deep. The vendor is responsible for their own code base and even when sold through intermediaries, code is increasingly delivered directly from the vendor's servers. This at least limits the number of entities involved that can lead to a supply chain problem.

Applications require regular updates, and compromising update code or processes has proven to be one of the most common application supply chain vectors. The SolarWinds attack is an example, in which adversaries were able to infiltrate SolarWinds and gain access to source code that was delivered via normal product updates. The scope of an organization's exposure will naturally vary widely based on the nature of the application. In many cases, the exposure may be limited to the theft or loss of the data that the application contains. However, applications that interact with many other systems may enable attackers to spread to other internal assets. Once again, the Solarwinds attack provides a good example - however, it is not coincidental that one of the most significant software supply chain attacks involved software that gave attackers access to an organization's critical infrastructure and equipment.

Device Firmware

The supply chains of physical devices and infrastructure are arguably the most complex, have the most risk, and virtually all of that risk is tied to firmware. Device firmware is the integrated logic that sits below the operating system and controls the actual function of the device and virtually every component within it. Many devices such as IoT devices will only have firmware. Other types of equipment such as networking gear rely heavily on integrated firmware or highly customized operating systems that are tightly integrated with the hardware. And while organizations will typically install their own OS image on new servers and laptops, the firmware is often left unchanged. In short, it is the firmware that governs how the device actually works, and it is the firmware in a device that persists.

Unlike applications and services, physical devices have incredibly deep supply chains that often change. A device can easily have hundreds of specialized components and systems within a device, involving dozens of suppliers and sub-suppliers spanning many countries of origin. Suppliers are often selected based on cost and availability pressures as opposed to security concerns. OEMs may also need to quickly pivot to new suppliers to adapt to changing conditions such as recent supply shortages of important components. A vulnerability, external attacker compromise, or malicious insider in any of these many moving parts can put the integrity of the entire device at risk.



And much like applications, firmware is regularly put at risk due to code reuse. Proprietary and open-source firmware components are regularly used by a variety of vendors, meaning vulnerabilities can be passed on to dozens of vendors. As an example, several leading equipment vendors were affected by UEFI vulnerabilities tied to a common firmware SDK. Likewise, the Ripple20 vulnerabilities in a commonly reused TCP/IP stack were passed on to dozens of vendors. But more importantly, these firmware-level vulnerabilities aren't just theoretical – they're rapidly and actively being exploited:

- HP recently rushed to patch 16 high-impact UEFI firmware vulnerabilities associated with active exploits in its Integrated Lights Out management interface
- CISA warned earlier this month about active exploits targeted at some 60 or so Cisco router vulnerabilities
- Ars Technica wrote about firmware exploits in February of this year under the headline "Russia's most cutthroat hackers infect network devices with new botnet malware", referring to Cyclops Blink attacks on firewalls

Organizations will also naturally buy equipment from a variety of vendors. Laptops, servers, and network infrastructure will all naturally have very unique supply chains. However, even products from the same vendor can vary from model to model or even between generations of the same model. And finally, an equipment supply chain compromise can be particularly damaging in terms of impact and scope. Any built-in vulnerabilities or threats would naturally be shared by similar devices, which could put an organization's entire fleet at risk. And with access to the most privileged code on a device, attackers would be free to pursue almost any malicious activity from stealing data, to establishing stealthy persistence, to permanently disabling devices and infrastructure.

Lastly, it is worth noting that the physical equipment and its firmware is the logical foundation of other aspects of the supply chain. Any applications and services will ultimately run on hardware, and if the underlying infrastructure is compromised, then the security of higher layers is either suspect or, in high risk environments, completely lost.

THE ROLE OF FIRMWARE IN SUPPLY CHAIN ATTACKS

Firmware is the first code to run on a device, and by sitting "below" the kernel it is also some of the most privileged code. These traits naturally make firmware very valuable to an adversary across many phases of a cyberattack. While a full analysis of the many ways attackers use and abuse firmware is beyond the scope of this document, let's review some of the most common ways firmware is used specifically in supply chain attacks.

Initial Access Vector

Firmware can be abused in multiple ways to provide adversaries with initial access into an organization. Most directly, firmware implants or backdoors can be introduced to the product before it is ever delivered to the eventual customer. Such an implant can be very basic and simply provide a trusted toehold within the device that can download additional malicious code and payloads. Such small bits of code can easily be added to existing firmware or stored in unused areas of firmware memory or "code caves".

It is important to note that an implant can be introduced into firmware in several ways. Naturally, an attacker or malicious insider within a supplier could directly insert a backdoor into firmware at the source. However, a variety of widespread vulnerabilities could be exploited by virtually any entity in the supply chain in order to modify firmware from another supplier. These same concepts apply to firmware updates as well, where attackers can directly compromise the source vendor or take advantage of insecure update processes.

However, vulnerabilities within the supply chain can provide attackers with an easy infection vector after a product has been delivered, and this has proven to be one of the most popular routes in real-world attacks. Vulnerabilities within network devices such as VPNs, firewalls, and routers have become some of popular initial infection vectors both for advanced state-sponsored adversaries as well as the most popular ransomware gangs (PDF).

Persistence

Firmware also provides adversaries a way to maintain their position on a device that is off of drives and beyond the realm of the operating system. This enables attackers to easily recover even if a threat is detected and the device is completely wiped, reinstalled, or has the system drives replaced. The recent iLOBleed attacks provide an example from the wild, in which attackers repeatedly reinfected HPE servers with ransomware after the threat was initially detected and the infected systems reimaged.

This level of persistence is particularly important in the context of supply chain attacks. After all, it doesn't make sense to compromise the supply chain if the attacker's code is immediately going to be updated and overwritten. Firmware gives access to code that is both highly privileged and long-lived.

Defense Evasion

Attackers can also take advantage of firmware as a way to avoid the prying eyes of security tools. Most security tools simply lack the ongoing firmware expertise and focus needed to reliably detect problems. This leads to a blindspot both in terms of vulnerability and risk management as well as threat detection.

Additionally, firmware sits in a uniquely powerful position capable of subverting the operating system and any security controls that rely on it. For example, malicious firmware or bootloaders can allow an attacker to control how the system boots and even directly patch the operating system in order to disable features. Malicious firmware can report false information to the operating system, making it difficult for traditional security tools to get reliable visibility into the firmware. Once again, the iLOBleed attacks provide an example in which the firmware would falsely report that the firmware was successfully updated even though a vulnerable version was retained. Once again, all of these traits are particularly relevant in a supply chain attack as it enables attackers to gain control over the protections that most organizations depend on.

Impact

Ultimately firmware can be used to cause extensive damage to an organization by stealing or destroying data or fully disabling the device itself. And unlike software-level attacks where data can easily be restored, disabling devices at the firmware level can permanently damage the equipment or take significant time and resources to recover. Additionally, the potential for supply chain attacks to affect an organization's entire fleet of devices can have a devastating impact on an organization's ability to maintain operations.

BEST PRACTICES FOR FIRMWARE SUPPLY CHAIN SECURITY

Organizations need to be able to verify the authenticity, posture, and integrity of all their critical equipment and devices. A strong supply chain security program will also extend throughout the lifecycle of the product from initial evaluation, acquisition, and ongoing operations of a device. Given that many traditional security tools lack insight into firmware, organizations may need to develop new processes and tools in order to properly assess their devices. A firmware security platform can automate and simplify many of the most important tasks, although they are not expressly required in order to achieve a reliable supply chain security program.

Key steps and best practices should include:

1. IDENTIFY: Establish Independent Firmware Visibility and Verification

Supply chain security has an innate need for independent 3rd party visibility and verification, and this is particularly pronounced when it comes to firmware. While technology vendors naturally play a critical role in supply chain security, customers can't simply rely on the vendor to verify their own integrity when the vendor itself may be compromised.

As a result, organizations must possess the technical tools to verify that the products a vendor delivers are authentic and have not been altered in the supply chain. All of the following technical steps and processes will rely on the ability to measure and analyze a device's firmware to identify vulnerabilities, threats, or other signs of tampering.

2. VERIFY: Automate Verification In Three Distinct Phases

Verification is one of the most critical steps, and can be broken down into three unique phases of activity that span from pre-purchase planning to roll-out and then to continuous monitoring.

2.1 Evaluate Prospective Equipment For Vulnerabilities Before Acquisition

IT and Security Teams should analyze all devices and components for known vulnerabilities and misconfigurations as part of the selection process. A reputable vendor should ensure that their products and all underlying components are free of weaknesses. Also vendors should be assessed to see if they deliver software build of materials (SBOMs) for the firmware included in the product. These SBOMs provide a clear declaration of the firmware in the device that staff can use as a reference later on when device's are ultimately delivered.

Additionally, an OEM will need to ensure that a wide variety of low-level device configurations and security settings are properly enabled and working together. For example, for a Secured-core PC, an OEM will need to properly integrate a variety of technologies from multiple vendors such as System Guard, System Management Mode, and the Trusted Platform module. Small misconfigurations or mistakes could easily put the system at risk in unexpected ways. When possible, the evaluation should include assessing the vendor's firmware update process for weaknesses such as not requiring signed firmware or sending update traffic in the clear.

A firmware security platform can easily scan prospective devices for these and many other vulnerabilities and weaknesses. This can provide critical insight into each vendor and their dedication to delivering secure products, as well as verifying the security of their own supply chain.

2.2 Scan All New Assets for Firmware Vulnerabilities and Threats

It is critical that organizations verify the integrity and security posture of all newly acquired devices down to the firmware level, even when dealing with a trusted vendor. Suppliers and components can regularly change, which could introduce new vulnerabilities or threats. Devices can also potentially be tampered with after leaving the manufacturer such as during warehousing, at a value-added reseller, or during the logistics and delivery process.

In addition to scanning for vulnerabilities, an analysis should verify that all received firmware cryptographically matches the published firmware available from the vendor or component supplier. Likewise, scans should include checks for known firmware threats. Even as threats evolve, attackers will often reuse components of previous firmware threats as seen in the recent MosaicRegressor implant which reused components of the Hacking Team UEFI implant.

This analysis should be applied to all devices and updates regardless of how they are acquired. For example, even if a new laptop is sent directly to a remote user, the device should be scanned or assessed remotely to ensure the integrity of the device. Likewise, teams should scan any equipment acquired as a result of any M&A activity.

2.3 Continuously Evaluate All Updates and Monitor Existing Device Behavior

If a vendor or supplier is compromised by an attacker, it is entirely possible that malicious code will be delivered as properly signed and otherwise seemingly “valid” code from the vendor. Simply comparing firmware to the vendor’s approved whitelist would naturally not reveal the threat since the threat would be baked in to the “approved” version.

As a result, organizations should continue to monitor the behavior of firmware after it is acquired or after an update. Unlike many other forms of software, firmware tends to be remarkably predictable in terms of behavior, which will allow malicious behavior to stand out. Security teams may be able to detect some obvious malicious behaviors such as reaching out to known malicious domains, while more subtle methods may require a firmware security platform. It is important to note however that firmware within components such UEFI, BMCs, or Intel’s AMT can have their own network stacks which will not be seen by the host operating system, making it important to monitor the behavior of the firmware directly.

3. FORTIFY: Patch and Update Firmware Throughout the Supply Chain

One of the critical issues pointed out in the joint Department of Commerce and Department of Homeland Security report cited earlier was the lack of updates and patching for critical firmware.

“Firmware updates present a major logistical challenge for many enterprises. In many instances, device firmware is never updated or may only be updated in an emergency.”

Why? As the government report goes on to say, the update process for firmware is never simple. “For example, it may not be clear what the latest firmware update is for a particular device. As a result, users may not be able to quickly determine whether the device’s firmware is up to date. For devices with firmware that is not cryptographically signed and secure, devices may be updated with unsigned code, meaning firmware could be rewritten without needing any verification from the user. A once-trusted device may no longer be trusted as secure after an unencrypted update.”

Organizations clearly need assistance in updating their firmware. That assistance needs to include locating the components that require updates and finding vetted and trusted update binaries with which to replace them. It also needs to include tight integration with standard IT automation tools like Microsoft SCCM or Intune, asset management solutions like Tanium, and endpoint visibility solutions like OSQuery.

Bearing in mind that “the cloud” is just “someone else’s hardware,” we need to constantly remind ourselves that the modern ICT supply chain also includes cloud platforms like AWS, Azure and Google Cloud. As such, organizations need their firmware security solutions to have rich APIs that integrate with the hardware beneath these leading cloud platforms. Only by assuring an update path for the firmware in those underlying systems can we fully assure our supply chains.

CONCLUSIONS AND NEXT STEPS

Technology supply chains are naturally complex and rely on highly interdependent webs of implied trust. Without the ability to independently verify the integrity and risks of the millions of lines of firmware that serve as the "DNA" for all this equipment, an organization's entire security program will be built on the shakiest of foundations.

To address these risks, organizations need to ensure that they can independently assess the security of their supply chain at all times. Firmware security tools – tools that identify, verify and fortify firmware code – are a key part of ensuring these efforts are done thoroughly, easily, and consistently. This ensures organizations never have to rely on implied trust, and instead can proactively verify their technology and mitigate risk before it ever becomes a problem.

To learn more, visit eclypsium.com.