Enterprise Best Practices for Firmware Updates

INTRODUCTION

A disciplined process of firmware updates is an essential element of good cybersecurity hygiene but can be challenging for many enterprises. This report provides IT and security leaders with insights into firmware update management and guidance on best practices.

Most enterprise IT and security teams understand the critical importance of keeping their operating systems (OS) and applications up to date and free from known vulnerabilities. While organizations typically spend significant resources patching and updating their software, the same process and rigor are often not extended to the firmware that underpins the fundamental behavior of system hardware. In many cases, the firmware on a device is never updated at all — or at best only updated in the case of an emergency.

“By 2022, 70% of organizations that do not have a firmware upgrade plan in place will be breached due to a firmware vulnerability.”

—Gartner, “How To Mitigate Firmware Security Risks in Data Centers, and Public and Private Clouds,” Tony Harvey, 3 July 2019 (Gartner subscription required)

However, the firmware layer has become an active portion of the enterprise attack surface, and organizations are increasingly paying a real price for ignoring firmware security. Recent ransomware has targeted firmware to cause damage, while other malware targets firmware to steal credentials and maintain persistence even after system re-imaging. In other cases, firmware has been attacked as a way to disable critical infrastructure. As attackers continue to target the unguarded firmware layer, it is clear that enterprises can no longer afford to turn a blind eye to firmware vulnerabilities.

According to Dr. Edward Amoroso, CEO of research and advisory firm TAG Cyber and former CISO for AT&T, the need to create world-class firmware update management processes is no longer optional: “Where firmware update management was once considered a nice-to-have component in an organization’s plans for dealing with vulnerabilities,” he explained, “it has now evolved into one of the central elements of any successful security program.”

While most CISOs and security teams would like to improve their firmware security hygiene, there are a variety of real-world challenges. Updating firmware is time-consuming, can be risky, and can require a system reboot and downtime. Organizations may lack the tooling to safely test and roll out updates, or to even know what firmware they have in their environment and if updates are available in the first place.

“Software patching is less daunting and feels safer, but you need to find that right balance of updating your software and hardware. You also need to work closely with your Infosec team to establish and communicate firmware policies.”

—Johnny, Systems Architect at a large high-tech company

This paper examines the current state of firmware update management and how the industry is evolving. We then recommend some specific steps security leaders can take to build a safe and reliable process that integrates firmware into an organization’s risk and vulnerability management strategy.

The Firmware Threat and Vulnerability Landscape

Before diving into the process of managing firmware vulnerabilities, we have to understand why the firmware layer is an attack surface worthy of our attention in the first place. With so many demands on their time, IT and security groups must prioritize work that has the maximum impact on the enterprise. It is therefore essential to recognize why firmware has rapidly evolved into one of the more critical areas in security.

A variety of factors can be boiled down to three main reasons:

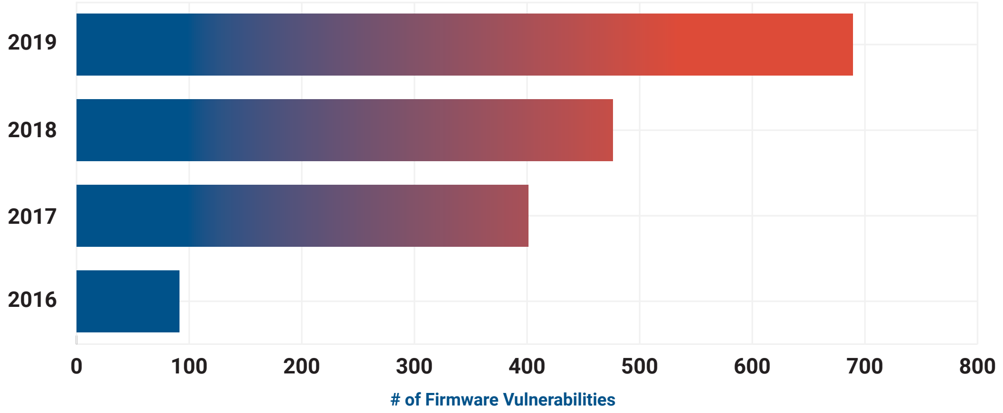
1. The number of identified firmware vulnerabilities is growing.
2. Attacks in the wild are increasingly targeting firmware vulnerabilities.
3. When exploited, these vulnerabilities grant attackers the highest levels of privilege and control over a system. Additional information on attacker motivations and techniques can be found in our online resource, Anatomy of a Firmware Attack.

THE RISE OF FIRMWARE VULNERABILITIES

The overall number of firmware vulnerabilities has skyrocketed in recent years, with each year after 2016 setting new high water marks for total vulnerabilities tracked in the National Vulnerability Database. In addition to having the most vulnerabilities ever recorded, 2019 saw a 43% increase compared with 2018, and 7.5 times growth since 2016.

This spike in vulnerabilities not only represents a rapidly expanding attack surface, it also indicates that both researchers and attackers are increasingly focusing on firmware.

Firmware Vulnerabilities By Year



Source: National Vulnerability Database December 31, 2019

FIRMWARE IS BEING ATTACKED IN THE WILD

The combination of opportunity and impact has translated into real firmware attacks in the wild and the discovery of additional vulnerabilities. A recent Forrester study found that "63% of companies have experienced a data compromise or breach within the past 12 months due to an exploited vulnerability in hardware- or silicon-level security." Likewise, analysis from F-Secure found that compromised firmware was the third-most common infection vector in 1H 2019.

And while firmware attacks are rising in number, they are also evolving in their approach and impact. Attacks include:

LoJax is a UEFI rootkit used by APT28 to exploit a vulnerability in the firmware configuration and install in a system's SPI flash memory. As it resides in the system's firmware, LoJax can survive a Windows reinstall, as well as a hard drive replacement.

JungleSec is ransomware that infects a victim's BMC through unsecured IPMI (Intelligent Platform Management Interface) commands. The FBI has published alerts urging organizations to patch their firmware as a response to threats from ransomware.

ShadowHammer was distributed over a legitimate ASUS Live Update utility and installed backdoors on thousands of Asus computers. As a result, the backdoor allowed hackers to freely access the victim's computer anytime — and without anyone's knowledge.

The waves of espionage and **cyber crime attacks by APT41** exploiting vulnerabilities and installing backdoors inside firmware on Cisco and Citrix network devices, and using ROCKBOOT MBR-based bootkit for persistence on Windows devices among other vectors.

FIRMWARE ATTACKS ARE HIGH-IMPACT

From an attacker's perspective, firmware presents an unusually high-value and strategic target. Firmware gives attackers low-level access to data that can be stolen or held for ransom. Moreover, components or entire devices can be disabled completely. Firmware also provides an avenue for attackers seeking to undermine traditional security in order to carry out long-term attacks against the enterprise. Motivations can include:

- **The Highest Levels of Privilege:** Firmware sits beneath the kernel of the operating system. With control over firmware, attackers can subvert the kernel and thus escalate to the highest levels of privilege on the device.

- **Bypass of Traditional Security:** Attackers can avoid security measures running at the operating system and virtual machine layers by controlling how a system boots, or by patching the operating system itself.

- **Persistence:** Malicious code in firmware is naturally tied to the hardware of the device, and can allow an attacker's code to persist even across a full re-imaging of the system.

- **Stealth:** Compromised firmware also enables attackers to perform critical attack functions without detection. For example, attackers have used out-of-band management features in BMCs and laptop chipsets as a command-and-control channel to evade host-based firewalls.

- **Damage:** Lastly, access to the firmware layer enables attackers to cause irreversible damage to a device. By damaging the firmware itself, attackers can "brick" the device permanently.

A FIRMWARE VENDOR PERSPECTIVE

Malware running at the firmware-level is persistent and cannot be easily removed. Sophisticated malware can intercept and prevent attempts to update the firmware to a known legitimate version of the firmware. Firmware-level malware can have full access to the PC and any other devices on the same network and can inject malware into the OS kernel. As demonstrated by the Hacking Team's UEFI BIOS rootkit, once malware is installed — even if the user reformats the hard disk (or buys a new one) and reinstalls the OS in the system — the malware remains. Compromised PCs on a private network can be used by hackers to exfiltrate sensitive data and infect other PCs or devices on that network. A vulnerability anywhere on the network makes the entire network vulnerable.

In response to vulnerability disclosures, PC product manufacturers make available to their customers firmware updates with patches to mitigate vulnerabilities. It is critical to install security patches as soon as they are made available to most effectively minimize firmware vulnerabilities. It is equally important to update software on hardware components of the PC and firmware on other devices that are attached to the same network, such as printers

.

phoenix® technologies

The combination of high-volume, high-impact vulnerabilities and real-world threats creates risk for enterprises that will need to be actively managed.

Developing a Firmware Update Program

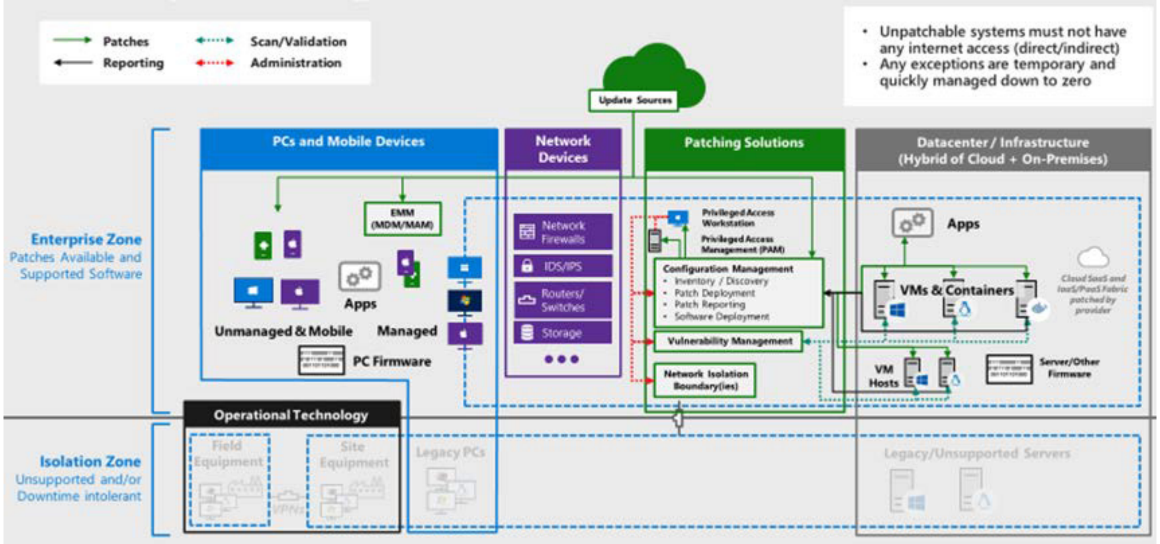
Vulnerability and asset management practices are bedrock components of any security program. Given the importance of firmware, it makes sense for organizations to update firmware with the same rigor and cadence that they use for operating systems. Firmware also should be included as a key component in an organization's overall vulnerability and risk management program.

FIRMWARE PATCHING OR FIRMWARE UPDATING?

> "It is important to acknowledge there is a difference between patching and updating. The terms 'patch' and 'patch management' are used when code is added to or modifies portions of an existing program, while 'updates' refer to completely replacing the target code or program. As such, applications and operating systems are often patched, while firmware is almost always updated. In both cases, new code is provided for the purpose of fixing bugs and removing vulnerabilities."
>
> — Tim Lewis, CTO Insyde Software

The ongoing NIST project **Critical Cybersecurity Hygiene: Patching the Enterprise** highlights the importance of maintaining up-to-date firmware as well as software. While this project is still in the build phase, the current document lays out key steps that an organization can follow to keep its various technologies up to date. Significantly, the document calls out both PC and server firmware and network devices as essential components that need to remain updated.

In the draft document, NIST is proposing that firmware updates also be considered a necessary and important component of cyber hygiene. Firmware is the foundation for all the other technologies and security properties, so it makes sense for organizations to update firmware with the same rigor and cadence that they use for operating systems.



- Unpatchable systems must not have any internet access (direct/indirect)
- Any exceptions are temporary and quickly managed down to zero

Security Patching Reference Architecture
Source: NIST, Critical Cybersecurity Hygiene: Patching the Enterprise.

FISMA, which lays out a comprehensive framework for securing government computing systems, similarly calls out firmware in many of the underlying security controls. NIST's Platform Firmware Resiliency Guidelines (SP 800-193) go even further by providing a detailed analysis of the firmware components and security mechanisms that can keep devices resilient in the face of attacks. For further information on how Eclypsium® can address firmware-related issues mandated by FISMA, please refer to our FISMA compliance best practices and quick reference guide.

Another useful reference for developing a firmware update program comes from Intel IT, which has published its internal strategy for maintaining firmware and drivers within the company. The document, **Developing a Gold Standard for Driver and Firmware Maintenance**, covers Intel's complete approach to firmware updates. This resource is particularly relevant for enterprises, as it gives a real-world perspective on how a very large organization that uniquely understands the importance of firmware security has tackled the challenges of firmware maintenance.

In particular, the Intel document highlights the fact that Intel does not necessarily install every firmware update available. The team evaluates the importance and value of each update against the potential for disruption to users and systems. Intel specifically prioritizes updates with critical security fixes, then evaluates the update to determine if it 1) addresses bugs in the environment, 2) introduces new features that are needed, or 3) is a prerequisite for operating system upgrades.

Gartner also provides guidance for infrastructure and operations leaders on firmware updates in its report "How to Mitigate Firmware Security Risks in Data Centers, and Public and Private Clouds." According to the report: "I&O leaders responsible for data center infrastructure should:

- Partner with the chief information security officer to develop the skills in the team to understand firmware and hardware threats by investing in emerging firmware threat detection and scanning tools and engaging with industry consortia and specialists in this area.

- Integrate a firmware upgrade policy into standard data center procedures, so that updates are made on a regular basis and emergency planning is in place.

- Secure access to firmware updates by implementing network isolation and user access controls, logging, and using the vendor's secure firmware features.

- Ensure that both cloud and on-premises vendors have secure firmware update programs by working with the vendor management team."*

*Gartner, "How To Mitigate Firmware Security Risks in Data Centers, and Public and Private Clouds," Tony Harvey, 3 July 2019

DEVELOPING YOUR FIRMWARE UPDATE STRATEGY

Organizations should incorporate these and other concepts into their existing processes for vulnerability management. While the details will vary from organization to organization, teams should define a strategy and policy around the criteria that are used to prioritize firmware updates. They should also determine conditions that might create legitimate business reasons to not push the update. Key topics should include:

Establishing Cross-Team Communication: Firmware updates will naturally affect a variety of teams, and it is essential that update decisions and procedures are coordinated across the organization. For example, security teams may need an update due to a critical vulnerability, but the implementation will fall to IT teams. Likewise, the business impact of an update will need to be identified — and the relevant teams and management informed — in order to arrive at an appropriate decision for the overall organization.

Understanding the Impact of Updates: Organizations need to understand the impacts an update will have on users as well as applications and services that depend on affected hardware. Teams should establish a phased rollout plan to ensure that representative systems are adequately tested before a full rollout. This also will ensure the organization can maintain an acceptable level of overall availability for systems and services.

Testing, Rollback, and Phased Rollout: Teams will need the appropriate systems and tools in order to test firmware updates before widespread deployment in order to identify potential negative impacts on target systems. Organizations will need tools to support the rollback and recovery of firmware in cases where problems are encountered. Phased rollout plans will allow organizations to 1) identify problems on lower-value systems before impacting more critical systems and 2) ensure that elastic applications can compensate for resources that are temporarily unavailable due to an update.

IT and Security Tools: To deliver on the goals of a disciplined firmware update policy, organizations will need a variety of supporting technical capabilities, which they may or may not have today. This would include establishing reliable visibility into the firmware of their critical devices both at the system UEFI level and at the component level.

Vendor Selection: The quality and manageability of firmware updates should be a key requirement when evaluating new hardware and vendors. This can include understanding how the vendor supports firmware updates. What tools are provided? Is encapsulation supported? Are firmware updates properly signed and the update process secure?

A PERSPECTIVE FROM LINUX VENDOR FIRMWARE SERVICE (LVFS)

Richard Hughes, who founded the Linux Vendor Firmware Service, is passionate about firmware update practices. He sees too many vendors who fail to follow basic principles, such as signing firmware with strong public/private key cryptography. He warns OEMs and ODMs that it should never be possible to "brick" a device with a failed firmware update, whether the problem is an unexpected hard machine power-down, the user physically unplugging halfway through an update, or flashing a "valid" firmware image to the wrong device. Simple steps such as reporting the progress of firmware update operations can help prevent users from thinking the update has stalled and that they need to restart the computer.

FIRMWARE UPDATE CHALLENGES & SOLUTIONS

In an ideal scenario, firmware updates would follow the same processes that are applied to operating systems and other critical software. However, in the real world, this is often not the case due to a variety of challenges.

The state of firmware updates today is similar to the software and OS landscape from the past 10 to 20 years. While automated OS updates have gradually gained acceptance in the enterprise, it took decades for enterprises to gain the necessary confidence in the quality of software updates and the ability to seamlessly roll back in case of a problem.

Operating system vendors, Original Design Manufacturers (ODMs) who design components, Original Equipment Manufacturers (OEMs) who sell finished devices, and open source projects are working to help modernize the way that firmware is managed and maintained. However, these efforts do not happen overnight, and ultimately they must deliver strong, proven track records that can be trusted by risk-averse IT organizations.

Some of these challenges are rooted in industry-wide issues that stem from how firmware is developed and delivered as part of an overall technology supply chain. Other challenges are more closely tied to the operational challenges within an enterprise, such as the potential for downtime, testing efforts, and rollback scenarios. However, while firmware poses some unique challenges, there are a variety of industry efforts underway to help organizations address them.

Industry Challenges

Fractured Firmware Ecosystem

It has taken the better part of 20 years for operating system vendors to develop processes for reliable OS updates and establish trust with enterprise IT teams. In comparison, the firmware landscape is even more daunting. The OS market is largely dominated by a few core vendors and projects with Windows, Linux, and Apple being the main players. By contrast, firmware and firmware updates come from a multitude of sources. Firmware may be developed by the component vendor or outsourced to a third party. This firmware may be passed on to OEM vendors who may use it unchanged or modify it based on their needs. This can lead to a situation where it is not always clear who is taking responsibility for the firmware or where customers should look for updates.

Solutions and Recommendations: Organizations need to be aware of all the firmware in their devices, including the firmware in components and peripheral devices. Ideally, teams should pay attention to upstream component vendors for firmware updates. However, this may only be practical for more sophisticated, well-staffed organizations. Additionally, organizations should strongly consider the quality of firmware management a vendor provides when making purchasing decisions. Organizations might furthermore consider investing in tools to automatically monitor all firmware and detect when updates are available.

Firmware Problems with Long Time Horizons

Unlike software, which typically undergoes a continuous process of development and updating, firmware development is intermittent — years can go by between updates. As a result, firmware updates are often much slower to be delivered and often not supported by the original firmware developers. Consider that many vendors will not update firmware at all unless driven by a specific incident or problem. Meanwhile, attackers may not put effort into looking for vulnerabilities until a particular device has a significant footprint in the wild. And then it can take defenders a long time to detect the presence of a problem due to a lack of visibility into or an inability to monitor the firmware layer. Once a problem is found years later, it falls to the manufacturer to develop a solution. Given that OEMs will often change suppliers, it is quite common that the original firmware developers are no longer available to offer a solution by the time a problem is found, causing even more delays.

For example, the vulnerable firmware code behind the ThinkPwn UEFI vulnerability was already 2 years old when it was discovered. Likewise, illicitly modified versions of the Lojack software from Absolute were in the wild for years before being found to contain malicious command and control domains.

Solutions and Recommendations: Organizations should select technology vendors with strong firmware management capabilities and integration with downstream suppliers. HP Support Assistant, for one, provides strong visibility into the firmware of the components within HP devices and can provide insight into when updates are available. Organizations may want to consider tools to track the state of all enterprise firmware across all devices instead of individually managing firmware visibility on a vendor-by-vendor basis.

Compromises in the Technology Supply Chain

Technology OEMs must maintain an often complex supply chain to support their firmware. This includes firmware licensed from third-party suppliers or included with components from ODM partners. Attackers can potentially compromise firmware within this supply chain, allowing them to infect a device even before it has been delivered to the end customer. To make matters worse, the update infrastructure itself can be compromised. For example, in the case of ShadowHammer, attackers were able to infiltrate ASUS and deliver malware to users that was properly signed with ASUS certificates and delivered through the official ASUS Live Update utility.

Solutions and Recommendations: Defense against supply chain attacks can vary depending on where and how the supply chain has been compromised. Organizations should scan newly acquired devices to ensure that the firmware matches the known, valid firmware available from the vendor — and to detect the presence of any known firmware implants. In the case of new, unknown implants or the compromise of a valid update from the vendor, organizations will need to have capabilities to monitor for abnormal firmware behavior. This will often require security tools specialized for this task.

Lack of Signed Firmware

Based on recent industry and Eclypsium research, it is not uncommon to find components within devices that do not verify that firmware is properly signed before updating or running firmware code. This means that these components have no way to validate that the firmware loaded by the device is authentic and should be trusted. This allows an attacker to insert a malicious or vulnerable firmware image, which the component would blindly trust and run.

Solutions and Recommendations: Firmware and updates must be signed by the relevant technology provider using strong public/private key cryptography. The private key must never be stored on the device itself and must be verified on the silicon or by an MCU that is responsible for writing to the shared flash. While the requirements above are the responsibility of the OEM/ODM, organizations should scan the firmware of devices and their components for vulnerable or unsigned firmware when evaluating new technology for purchase. They should choose vendors that only use signed firmware in all components.

Lack of a Common Vulnerability Taxonomy

Organizations that integrate firmware into their overall vulnerability management programs will often notice that many of the tools and standards available for software are not available for firmware. Vulnerability scans rarely extend to firmware vulnerabilities. The same issues plague the industry itself. For example, while firmware vulnerabilities are included as CVEs, they often are not assigned Common Weakness Enumeration (CWE) codes, which developers and security teams use to understand the underlying weakness of a vulnerability. Intel and other industry forces are increasingly pushing to extend the CWE concept to include hardware issues. Increased standardization would assist researchers as they communicate findings and also help organizations understand the impact of a vulnerability in their environment so they can make informed decisions about prioritization.

Solutions and Recommendations: Such large industry-level challenges will naturally require the coordination of many organizations and vendors. As a result, such changes are likely beyond the scope of any individual enterprise. However, organizations should voice their support for industry efforts such as Intel's and urge other vendors to include hardware and firmware vulnerabilities in the existing taxonomies that are used for software.

Enterprise Challenges

Lack of Firmware and Hardware Inventory

Visibility remains one of the most fundamental challenges for organizations when it comes to firmware and hardware. Teams often lack the basic insight into what firmware is in their devices, if the firmware contains vulnerabilities, and if there are updates available. This problem is greatly magnified in organizations with staged hardware refresh cycles where multiple different hardware platforms are in use.

Solutions and Recommendations: The ability to inspect and inventory an organization's firmware is a prerequisite for good firmware management. While some manufacturers provide better visibility into firmware than others, the variety of device types, vendors, and underlying components can make it almost impossible to keep track of all firmware in an enterprise. Centralized, regular scanning of devices in the environment provides a consistent way to know which devices need to be updated.

Difficult Update Processes

Traditionally, firmware updates have required a more manual effort from IT and security teams as compared to software updates. This naturally creates a higher barrier to updating and ultimately results in an unprotected attack surface. Additionally, firmware updating can often require coordination between different functional teams within an organization. As mentioned before, security teams may champion an update based on a critical vulnerability, but the actual process of applying the updates may fall to an IT team. Meanwhile, teams can encounter other complications during firmware updates, such as prompts for BitLocker recovery keys due to the PCR0 changing. While some operating systems and OEMs provide the option to suspend BitLocker during an update, not all do.

Solutions and Recommendations: Firmware encapsulation provides teams with a much more automated approach to firmware updates that follows much the same model that organizations use for maintaining their operating systems and software. Organizations will need to have visibility into their devices to understand which components can be updated via encapsulation and which will require a more manual approach. In either case, teams must still prepare for testing firmware updates to find potential problems before updates are rolled out broadly. Organizations should be aware of tools and capabilities encapsulating multiple firmware updates and start integrating them into their firmware management strategy. They also may want to consider the availability of firmware encapsulation as a factor during the vendor selection process. Finally, they might want to consider the option to suspend BitLocker during an update when evaluating vendors. Regardless of the update process(es) available, organizations will need to establish procedures that establish clear communication and coordination between the many functional teams related to firmware.

Potential Negative Effects

The potential for a bad firmware update to cause damage remains one of the most persistent fears for enterprises when it comes to firmware updates. The fear of "bricking" (i.e., completely disabling a device) has been enough to keep some organizations from updating firmware in anything but the most extreme circumstances. Fortunately, increased testing and support for both automated and manual firmware rollback options have made firmware updates far safer than they were 10 or 15 years ago. In particular, leading enterprise technology vendors such as Intel, HP Enterprise, Dell, Lenovo and others have invested in tools to automatically roll back to known good states of firmware in the case of a failed update. Specifically, Dell provides a BIOS Recovery Tool, and HP Enterprise includes automated rollback as part of its Smart Update Technology.

However, firmware can also have more subtle impacts on devices that may not be easily recognized, such as an update that makes the device run hotter or affects performance in other ways. Updates can also potentially affect device configurations that may inadvertently disable security features or the secure boot mechanisms of the device.

Solutions and Recommendations: While many vendors have vastly improved their capabilities for both automated and manual firmware rollback, many IT teams are not aware of these improvements. Teams should ask their vendors about these capabilities and factor them into procurement decisions. Organizations need to create processes and develop tools to properly test firmware updates, stage rollouts of updates, and support rollback and recovery options when problems are detected. At the same time, organizations must develop their own processes for testing and controlling the rollout of firmware. Teams should have tools to scan updated devices to ensure that secure boot and other security settings were not disabled during the update. Testing and manual rollback options are also particularly important. Teams need to test any automated rollback functionality and have backup plans to roll back to known good states if automated methods fail

Device Downtime

One of the persistent challenges of firmware management is that updates frequently require a reboot of the device. This means that even in the best of circumstances firmware updates will require a certain amount of downtime. This concern is exacerbated when it comes to updating servers that support critical applications or a variety of assets hosted within virtualized and containerized environments. Additionally, the update process may need to be performed or authorized by an IT team that might be resistant to bringing devices down even briefly.

Solutions and Recommendations: While some amount of downtime is unavoidable, organizations can at least minimize the impacts through good testing of firmware updates, update scheduling, and staged rollout processes. Testing firmware updates and rollback procedures against a test version of target systems can ensure IT teams will know the actual expected and worst-case downtime of a resource before applying the update. With this information, update teams can communicate with any affected teams and choose appropriate update windows to minimize the impact to the organization. A phased rollout plan can be particularly helpful when updating servers supporting elastic applications and services. By phasing in updates of the infrastructure, organizations can allow hardware to be updated while maintaining application availability. Additionally, organizations may want to consider developing a system similar to the one outlined in Intel's whitepaper, which establishes a standardized approach to firmware that includes input from a variety of functional teams, including IT.

BEST PRACTICES AND RECOMMENDATIONS

1. Establish an Organizational Strategy and Policy Around Firmware Updates

Develop a cross-functional policy toward firmware updates with similar principles to those established in Intel's whitepaper. The goal is not to replicate Intel's specific policy but rather to establish agreed-upon conditions about when firmware should be updated. These conditions should account for factors including the risk of not updating, the need for new features and the potential cost of downtime, and they should plan appropriately for the update process.

- Establish buy-in from all relevant teams and technical stakeholders.
- Establish processes for emergency updates needed for critical security issues.
- Leverage CVSS scores and risk-based vulnerability data when prioritizing hardware and firmware vulnerabilities.
- For regular updates, establish criteria to determine if an update should be deployed based on security impact, feature impact, risk of downtime, etc.
- Establish a firmware-inclusive risk assessment to assist with updating decisions. What attack vectors are relevant to a particular vulnerability? Is a device potentially exposed to those vectors?

2. Establish Firmware Visibility

Visibility into firmware can be achieved in a variety of ways. Review the capabilities of platform vendors in terms of identifying when firmware updates are available for various components. Additionally, consider a cross-platform approach capable of scanning devices to identify the firmware used, its vulnerabilities, and any available updates. Organizations may need more security tools in order to meet some of these requirements.

- Ensure visibility into UEFI, BIOS, integrated components, and peripherals
- Ensure coverage for all supported devices and platforms — laptops, servers, network infrastructure, etc.
- Implement a method to identify, track, and manage newly discovered vulnerabilities. This can include working with security vendors and accessing open-source intelligence (OSINT).
- Regularly scan for known firmware vulnerabilities in laptops, servers, and networking gear.
- Detect when updates are available for deployed devices.
- Maintain visibility for firmware version information.
- Check to see if intermediate updates are required between the currently installed version and the target version of firmware.
- Maintain visibility for firmware configuration, both before and after a firmware update (e.g., is USB is disabled? What is the status of Secure Boot? Is SGX enabled or disabled? Is BIOS protected by password?)
- Check externally connected devices (e.g., removable USB drives) for any potential compatibility issues with a given firmware update.
- Collect and monitor firmware from a vendor resource to identify if firmware is revoked due to stability issues or in case of supply chain attacks.

3. Develop Tooling and Skills Needed for Testing, Rollout, and Rollback

The testing and deployment of firmware will require an investment in both processes and tools. Staff needs to be trained, and testing should be customized based on the type of device being updated.

- Evaluate the process required for updating firmware of various components.
- Determine which components can be updated via encapsulation vs. a manual process or other methods.
- Ensure teams have the appropriate tools and reference systems to adequately test firmware updates for problems.
- Establish a phased rollout program with initial testing from a buy-in group such as IT and security staff. Stage updates to minimize the impact of downtime to server farms or local cloud assets. Phase rollout to smaller groups and lower-value assets for additional testing before completing rollout to the larger fleet of devices.
- Test all automated rollback and recovery capabilities from various vendors.
- Develop tools and processes for manual rollback to known good versions if necessary.
- Be aware of devices with anti-rollback measures for firmware. Some vendors prevent downgrading to an earlier version of firmware in order to prevent attackers from rolling back to earlier, vulnerable versions of firmware. While this can provide a layer of protection, it may mean that the device cannot be put back into a good state if the firmware update has adverse effects. Teams will need to take this trade-off into account when deciding to update.
- Be aware of large jumps in firmware versions (e.g., from version 1.x to 6.x), as this can have side effects or even brick the system.
- Check to make sure the firmware successfully upgraded to the expected version after an update.
- Analyze the configuration to make sure the firmware update didn't change any sensitive configuration settings following an update.
- Educate end-users about the importance of firmware updates and what to expect from the update process.

"All my server purchasing decisions are based on security and who has the best capability to automate hardware and firmware updates"

—Alex, Systems Engineer, Patch Management for a Fortune 500 company

4. Make Firmware Support a Priority in Hardware Purchasing Decisions

The fractured nature of the PC industry means that there can be a wide variance in terms of how vendors prioritize firmware security. Strong security practices are often not easy, and some vendors may eschew good security in order to reduce cost and overhead. However, small reductions in hardware costs are often dwarfed by the human manpower requirements an organization must meet to manually deal with poorly supported firmware in a device.

- Include a firmware updating process as part of platform evaluations.
- Demand strong firmware updating features as a critical feature for established vendors.

CONCLUSION

While updating firmware can be a daunting task, organizations should take solace in the fact that the industry has successfully conquered similar challenges before. In many ways, firmware is going through the same growing pains experienced by software and OS vendors in the past 20 years. By building the appropriate strategies, tools, and processes, and by selecting vendors that prioritize firmware management, organizations can build a reliable path to firmware security.

The simple fact is that, as firmware has become an increasingly targeted layer of the enterprise, organizations need to include firmware updating as part of their overall approach to security hygiene. While many of the basic goals of patch management apply to firmware, firmware presents some unique challenges that organizations will need to prepare for. Some vendors have made strides to make firmware updates (and rollbacks) more automated, but functionality varies considerably from vendor to vendor. The wide variety of firmware-dependent components within a device has further complicated matters, making it difficult for many organizations to even know what firmware they have in their environment.

To compensate, organizations will need to develop an overall firmware strategy as well as new skills, processes, and tools tailored to the unique requirements of firmware updating. As with many other disciplines of security, establishing visibility is a critical requirement. Teams need to be able to see what firmware is used across all of an organization's critical devices, including the many firmware-dependent components within those devices. Teams need to know when firmware updates are available, and they need established criteria to determine when an update should be applied.

Next, organizations must know the various mechanisms available to update firmware, whether driven by the operating system or applied manually. In order to cover all components, a team will likely need to support multiple updating strategies, which could impact the time and effort required for an update. Lastly, organizations will need the ability to test firmware updates and establish a process for phased rollouts in order to detect any problems related to the update. They should also have processes in place to roll back firmware as needed.

Naturally, every organization will be somewhat unique and have its own challenges. To learn more about building a firmware update program, or how to monitor enterprise firmware in general, please contact the Eclypsium team at **info@eclypsium.com**.

ACKNOWLEDGMENTS / THANK YOU

Eclypsium would like to thank the following organizations that contributed to this report: Criteo, Insyde Software, Linux Foundation, TAG Cyber, and Phoenix Technologies, as well as several other contributors and reviewers who have asked to remain anonymous.

ABOUT ECLYPSIUM

Eclypsium is the industry's leading enterprise firmware protection platform — providing a new layer of security to protect laptops, servers and network devices from firmware attacks. The Eclypsium platform provides full visibility into the firmware running on all the key components of enterprise laptops, servers and network devices. At a glance, you'll see if there are implants or backdoors in your firmware, if it's vulnerable to known threats, or if it's just out of date and in need of an update. You'll get expert guidance on the severity of vulnerabilities, and links to the latest firmware updates, so that you can mitigate threats and protect your assets. To learn more about how Eclypsium can help you improve your firmware security, contact us at **info@eclypsium.com**