# Definitive Guide™

## to

# *Enterprise Firmware Security*

Defending your fast-growing, unguarded and almost invisible attack surface


**Crystal Bedell & Michael Thelander**

**FOREWORD BY:**
**John Loucaides**

*Compliments of:*

eclypsium®

**About Eclypsium**

Eclypsium is the enterprise firmware security company. Eclypsium's cloud-based SaaS platform identifies, verifies, and fortifies firmware wherever it exists in your extended global networks: in laptops, tablets, servers, network gear, and connected devices. The Eclypsium platform secures against persistent and stealthy firmware attacks that your EDR tools miss, assures continuous device integrity, delivers firmware patching at scale, and prevents ransomware and malicious implants. Serving Global 2000 enterprises and state and federal agencies, Eclypsium was named a Gartner Cool Vendor in Security Operations and Threat Intelligence, a TAG Cyber Distinguished Vendor, and one of the World's 10 Most Innovative Security Companies by Fast Company. Learn more at eclypsium.com.

# Definitive Guide™
## to
# *Enterprise Firmware Security*

Defending your fast-growing, unguarded and almost invisible attack surface

**Crystal Bedell &**
**Michael Thelander**

Foreword by John Loucaides

CYBEREDGE
PRESS

**Definitive Guide™ to Enterprise Firmware Security**

Published by:
**CyberEdge Group, LLC**
1997 Annapolis Exchange Parkway
Suite 300
Annapolis, MD 21401
(800) 327-8711
www.cyber-edge.com



For general information on CyberEdge Group research and marketing consulting services, or to create a custom *Definitive Guide* book for your organization, contact our sales department at 800-327-8711 or info@cyber-edge.com.

ISBN: 978-1-948939-25-6 (Paperback)
ISBN: 978-1-948939-26-3 (eBook)

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

---

## Publisher's Acknowledgements

CyberEdge Group thanks the following individuals for their respective contributions:

**Editor:** Susan Shuttleworth
**Graphic Design:** Debbi Stocco
**Eclypsium Contributors:** Chris Garland and Joe Hopp

# Table of Contents

# Foreword

A 2019 report from analyst firm Gartner® made an ominous and somewhat surprising prediction: "By 2022, 70% of organizations that do not have a firmware upgrade plan in place will be breached due to a firmware vulnerability."[1]

Fast forward to today, and Gartner's prediction doesn't seem quite so far-fetched. Firmware-centered attacks have rocked SonicWall customers. Firmware in Accellion file transfer appliances became a beachhead for punishing ransomware attacks. APT actors used firmware to compromise Fortinet devices, and then sold or gave their code to criminal gangs.

Those are just a few examples of firmware-based attacks. Now Gartner's 70% prediction seems conservative.

Computers do not have to work the way we expect. We build layers of abstraction to avoid interacting with their myriad complexity. Those abstractions always include a set of assumptions. Somewhere within every supply chain is an assumption on the configuration and state of the hardware in every system. That assumption rests on the subject of this book: firmware.

Enterprise IT is complicated enough. Discovering assets, managing their risk, and taking quick action to remediate issues has driven many security teams to overwork and burn out. If we now consider the assumptions on firmware inside each component of a device, the complexity grows dozens or hundreds of times. There's no manual way to handle that at the scale of the enterprise.

That's why this book is so important. It provides a practical strategy for testing deep firmware and hardware assumptions at a scale that would otherwise be intractable. Like everything in cybersecurity, there will always be more vulnerabilities, more risky assets, more malware, and more attack campaigns than we can handle. However, the strategy of *Identify, Verify, and Fortify* laid out in this book carves out a path that will help you keep up with firmware risks by using your existing people and processes.

1. Gartner, "How to Mitigate Firmware Security Risks in Data Centers, and Public and Private Clouds," Tony Harvey. Published: 3 July 2019 ID: G00387620

This book could not come at a more critical time. Attackers will continue to cause damage one way or another, and constantly expanding the scope of risk cybersecurity staff must manage will only make the situation worse. Instead, the *Identify, Verify, and Fortify* approach will fit into existing organizational processes. It will allow security teams to focus on creative and exciting ideas while automated tools focus on the deeper mechanics and bubble up meaningful results.

I would like to thank Crystal Bedell and Michael Thelander for writing this book, and also Yuriy Bulygin, Alex Bazhaniuk, Steve Mancini, and the entire team at Eclypsium for making this effort possible. Visibility into the black box of firmware and hardware security no longer requires a team of researchers for every device. Automated tools, industry-wide awareness of firmware risks and threats, and shared expertise make it possible to defend the firmware foundations of our infrastructure throughout the enterprise.

**John Loucaides**
**Vice President**
**Eclypsium**

# Introduction

Over the years, security teams have evolved and strengthened their defenses to make it increasingly challenging for cyberattackers to successfully break into the IT environment. Unfortunately, while security teams have focused on securing operating systems and applications, firmware has become the unguarded attack surface of the enterprise.

Most organizations lack visibility into firmware. They can't see the vulnerabilities, threats, and backdoors that expose the enterprise to significant risk – never mind identify which versions of firmware are running. Fortunately, awareness of firmware security is increasing and security professionals like you are learning how to defend this attack surface. Picking up this book is your first step. This book will give you the tools and understanding you need to manage firmware risk and defend your enterprise against firmware attacks.

## Chapters at a Glance

**Chapter 1, "Understanding Firmware,"** explains what firmware is and the scope of the firmware ecosystem.

**Chapter 2, "Weaponizing Firmware,"** explores why attackers target firmware and why traditional security controls fail to protect this growing attack surface.

**Chapter 3, "Identifying Firmware,"** outlines the challenges and considerations that need to be taken into account when discovering firmware in the enterprise.

**Chapter 4, "Verifying Firmware,"** describes the importance of vulnerability assessments, zero trust, and regulatory compliance as they relate to firmware.

**Chapter 5, "Fortifying Firmware,"** explores the firmware update process, threat detection and response, and comprehensive monitoring.

**Chapter 6, "Scaling for the Future of Firmware,"** examines the firmware security in federal governments and the need for a vendor-agnostic firmware security solution.

**Chapter 7, "Selecting the Right Solution,"** reviews the 10 requirements for an enterprise firmware security solution.

## Helpful Icons


Tips provide practical advice that you can apply in your own organization.


When you see this icon, take note as the related content contains key information that you won't want to forget.


Proceed with caution because if you don't it may prove costly to you and your organization.


Content associated with this icon is more technical in nature and is intended for IT practitioners.


Want to learn more? Follow the corresponding URL to discover additional content available on the web.

Chapter 1

# Understanding Firmware

## In this chapter

- Learn what firmware is and what it does
- Understand how firmware represents many single points of failure within your IT environment
- Explore the breadth and scope of the firmware ecosystem

Organizations have a vast array of devices on the enterprise network. From servers and networking equipment to laptops and peripherals, all of these devices have one thing in common: *firmware*. This ubiquitous, component-level software is essential to the proper functioning of system hardware. In this chapter, we take a closer look at firmware, what it is, and what it does.

## Firmware All the Way Down

Today's servers, laptops, and networking equipment include dozens of components. Each of those components has its own complex software programming with millions of lines of code. This programming is called firmware and it exists in virtually every component of a device, from its chipset and processor, to drives, network adapters, graphics cards, memory, USBs and PCI buses, and more.

### *What is firmware?*

Firmware is the foundational code within a device. For endpoints and servers, firmware represents the code below the code – that is, the code that lies below the operating system (OS), the applications, and any updates or add-ons that are later layered on. In the broader sense, firmware is manufacturer-provided code shipped with any kind of device.

## *What firmware does*

You can think of firmware as the DNA of a device. It is the embedded code and microcode that, quite literally, tell the device how to act and how to behave.

Firmware operates at the highest level of privilege. It has direct access to critical resources and is a foundational dependency for the OS and application software. Firmware governs how a device boots and has the ability to modify the OS. Firmware largely runs in a layer that the OS doesn't see or control.

An exception can be found in the many appliance-type network devices that ship their "operating systems" as firmware. But the general distinction of highly privileged, manufacturer-provided code remains the same.


Hard drives, network cards, BIOS, and other components all have their own software stacks, forming a hidden attack surface beneath the OS layer. Don't underestimate or ignore this often-invisible attack surface.

### Cloud Considerations

You may be thinking, "We're a cloud-first company with no hardware. It's my cloud provider's responsibility to worry about firmware security of their underlying devices." This is an easy assumption to make and, possibly, a dangerous one.

Throughout this book you'll see sections labeled "Cloud Considerations." These notes highlight areas security professionals need to understand — and questions you can ask — to help you make the right strategic decisions about firmware security in cloud environments.

# The Firmware Ecosystem

Firmware is developed by a wide variety of manufacturers or their outsourced firmware experts. Each hardware device in your environment is produced by a collection of manufacturers.

## *Hardware and software*

Every hardware device relies on its own firmware. Indeed, every device has scores of firmware components. According to a Gartner® report, "Firmware has low-level kernel and hardware access, and modern PCs have 15 to 20 pieces of firmware software loaded into memory on every startup." The report further states, "It is rarely patched and often unpatchable."[2] Servers may have 20-30 components. Every network device has embedded firmware. That fact alone can be overwhelming when you multiply by the numbers of servers, switches, laptops, smartphones, routers, etc., that connect to the enterprise network. The vast majority of devices rely on an array of components and subcomponents, and each of those *also* has its own firmware and supply chain. The result: a very broad and deep attack surface into which organizations have very little – if any – visibility. We'll explain why in Chapter 2.

## *Single point of failure*


Independent from the OS, firmware is the first code to run and can modify or subvert the OS and applications running at higher levels. This makes firmware a single point of failure that, when compromised, can allow attackers to evade security controls at those higher layers and silently persist on a device.

Risk naturally increases as the attack surface grows, and supply chains amplify this "single point of failure" notion by concentrating firmware at the lowest level of the technical stack. Each component has an operating environment that builds up the assumptions on which the OS bases its work. If any one of the links in the supply chain is exploited, the entire system is compromised. Given the scope of the supply chain, there isn't just one single point of failure – there are

2. Gartner, Roadmap for Improving Endpoint Security, Peter Firstbrook. Refreshed: 17 November 2020 | Published: 19 June 2018 ID: G00343353

hundreds, possibly thousands, of single points of failure in an IT environment.

Even firmware that has not been tampered with has the potential to be a single point of failure. No code is perfect. Vulnerabilities may be discovered at any time. Firmware is often passed on and reused within a variety of products – causing those vulnerabilities to appear in a number of seemingly unrelated devices. For example, the Ripple20 vulnerabilities were found within a widely used TCP/IP software library. Over 30 vendors reused this code in devices ranging from laptops and servers to printers, medical devices, and critical infrastructure.

## Persistence

Firmware has another attribute that attackers love: it persists. Even after a system wipe, it persists. Even after re-provisioning and restoration of the base OS, the original firmware is, in most cases, still running.

Even in cloud workloads the underlying bare-metal equipment is often reassigned to other customers or teams. Some practitioners "re-flash" their firmware, or overwrite embedded code with trusted, known-good source code, as part of their re-provisioning process. But most do not, and whatever vulnerabilities or implants were riding along with the firmware will persist for the next users or the next workload.

## One device, dozens of components

The attack surface grows exponentially when you consider the full extent of the supply chain. A single device is composed of dozens of components. Each of those components has a supply chain of its own. Each link in the supply chain represents an opportunity for an attacker to modify or insert additional code.

The further away from the original equipment manufacturer (OEM) you go, the higher the likelihood that suppliers were selected based on price rather than the quality of their code, so it's possible for firmware code to ship with a vulnerability, such as a logic bug or buffer overflow.

Chapter 2

# Weaponizing Firmware

## In this chapter

- Learn why attackers are targeting firmware
- Understand the implications for your organization
- Explore current firmware threats and attack trends

For decades the security industry has focused on securing operating systems and the applications above them. And we've gotten pretty good at it. In most cases, attackers must work hard to get through our defenses. Unfortunately, we've done little, beyond establishing BIOS-level passwords, to secure the multiple firmware components that support those devices. In this chapter, we'll look at why that is and how attackers are using firmware against us.

## Firmware Is Under Fire


Firmware represents a prime opportunity for attackers who are looking for easy ways to increase the severity and scale of their attacks. Within the supply chain, firmware is the earliest and most fundamental way an attacker can compromise a device. The tools, knowledge, and vulnerabilities attackers need to attack firmware are also readily available. For this reason, this unmanaged attack surface is actively under attack.

Most organizations lack visibility into their firmware attack surface. They can't gain a holistic, deep view of what versions of firmware are running in every component of an enterprise device, or determine whether they're vulnerable to known threats... much less detect a hidden implant or backdoor. Once compromised, this blind spot allows attackers to subvert traditional security controls and persist undetected, leaving you

exposed to device failures, ransomware, and data breaches. It can also create a very long dwell time before anything is discovered. Some attacks, like MosaicRegressor and ESPecter, were only discovered and revealed years after the events took place.

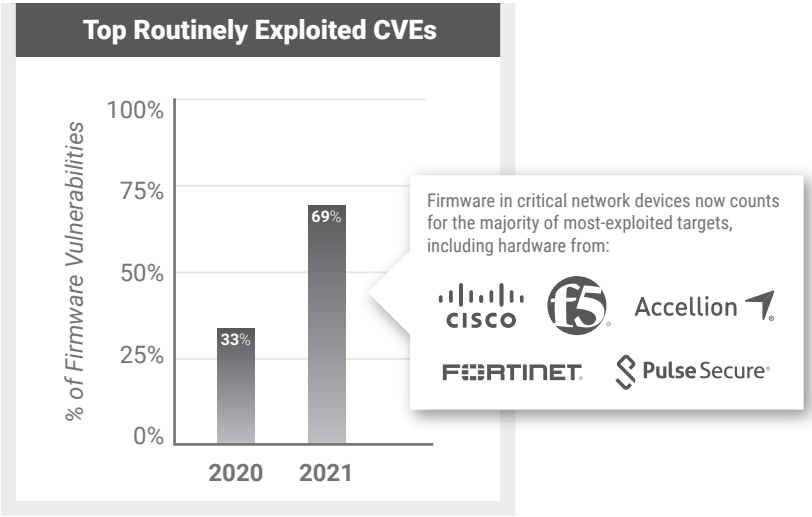



**Figure 2-1**: Firmware makes up a fast-growing percentage of the top routinely exploited vulnerabilities according to 2021 advisories from the U.S. Cybersecurity and Infrastructure Security Agency (CISA).

## Why Manage Firmware Risk


Every organization faces the significant risk of a firmware attack. As we explained in Chapter 1, the firmware landscape in any IT infrastructure extends far and wide, and, as shown in Figure 2-1, the risk of a firmware attack is growing. Unfortunately, most organizations aren't properly equipped to understand these risks.

Then there are the ramifications of a compromise. Malicious firmware can enable an attacker to subvert traditional security controls and silently persist without detection, even surviving a complete reimaging of the device or installation of a new drive.

Industry organizations and lawmakers are growing increasingly aware of the risk posed by firmware and the need to secure it. We explore standards and regulatory compliance in further detail in Chapter 4.

# Firmware Falls Between the Cracks

Until recently, and through no fault of their own, organizations implicitly trusted vendor firmware. This was by design. If a network device or endpoint is designed and built well, users shouldn't have to think about how it works. It just does. Not only that, but vendors don't want customers tampering with the code that tells their products how to operate. Firmware exists to be forgotten.


But firmware is built by humans. And just like any other code, it is subject to vulnerabilities and misconfigurations. If we consider firmware to be the DNA of a device, as we described in Chapter 1, then in the same sense that broken DNA can cause immeasurable harm to a life form, "broken firmware" — that is, firmware that's been tampered with, is out of date, or is misconfigured — can cause almost any hardware or software system to fail catastrophically.

## *Lack of security controls*

Firmware vulnerabilities, weaknesses, and exploits are an increasingly critical concern in existing security product categories, such as vulnerability management, endpoint management, supply chain security, network security, and cloud security. However, the technology solutions within these categories are not capable of finding, assessing, managing, and securing components at the firmware layer.

By compromising firmware, attackers can control and persist on a device in a way that's virtually undetectable by traditional security software. Traditional security controls are often limited to the OS and software layers and lack visibility into threats at lower levels, where the firmware resides. Limited visibility below the OS means that firmware threats go undetected — even in enterprises that routinely perform active software vulnerability scanning.


Unfortunately, vulnerability solutions don't address firmware threats and exploits. They're not built for it, and because of the underlying architectural challenges, vendors and their customers have been slow to build appropriate systems and safeguards to secure firmware. This is due, in part, to the difficulties involved. Firmware isn't designed to be easily monitored and updated. Remember, you're not supposed to have to think about firmware, and that includes actively monitoring and protecting it against cyberthreats.

## *Lack of firmware expertise*

For IT and security teams tasked with protecting infrastructure from attack, the challenge of keeping up with firmware updates has grown significantly, and the severity of the issue demonstrates the alarming gaps in firmware security. As firmware threats become mainstream, it is incumbent upon organizations to build the appropriate defenses against this growing area of risk.

Unfortunately, IT and security teams are already overworked and understaffed. Like every other technology these teams oversee, firmware is constantly changing. New firmware vulnerabilities and threats from advanced actors, as well as large-scale opportunistic campaigns, pop up every day. Most organizations lack the expertise to assess and defend against active firmware attackers.

The organizational structure of security teams also presents a challenge to managing firmware risk. Because firmware is everywhere, firmware-based attacks are also everywhere, targeting endpoints, servers, network devices, and Internet of Things (IoT) sensors. However, security teams are traditionally organized around platforms, applications, or types of compute or storage infrastructure. Few organizations are staffed and structured to secure firmware in a consistent way across these hardware, functional, and organizational boundaries.

# Firmware Threats and Trends

Firmware threats aren't new. Sophisticated attackers have used firmware backdoors and implants for years. However, in recent times firmware threats have become far more widespread and attacks more commonplace.

## *Five active attack vectors*

Firmware threats and vulnerabilities are often an unknown quantity for many organizations. It is critical that security teams understand how these threats work and their path into critical devices. Here are five of the most significant firmware threat scenarios based on current trends:


### Attackers exploit network and VPN firmware

The shift to remote work in response to the COVID-19 pandemic resulted in the rapid expansion of virtual private network (VPN) infrastructures to enable remote connectivity. Attackers took notice.

The U.S. Cybersecurity and Infrastructure Security Agency (CISA) issued multiple alerts in 2021 detailing how state-sponsored actors from China, Russia, and Iran were targeting vulnerabilities in enterprise VPNs and other network controllers. These vulnerabilities were remotely exploitable and directly linked to the integrated code and firmware running on these network devices.

Network infrastructure is also targeted by attackers. A variety of attackers have implanted firmware backdoors in compromised network devices. Large-scale Russian attacks have likewise targeted the network infrastructure of government and private sector organizations.

These threats can have a devastating impact on the affected organizations. By compromising the fundamental code of a network device, attackers can spread malicious code within the network and potentially manipulate traffic by copying or rerouting it, or by inserting a man in the middle.

### Firmware is leveraged in ransomware attacks

Ransomware is one of the most pervasive and highest-impact threats organizations face today. Attackers have found

strategic value in leveraging firmware as part of a ransomware attack. Ransomware operators target firmware in network devices to gain access to and spread malware within an organization. Attackers also use firmware to support core ransomware functionality. By compromising or controlling the firmware and master boot record (MBR) of victim devices, attackers can seize fundamental control of the device while maintaining persistence and evading security controls. An attacker can, for example, disable a device while ensuring that their malicious code always runs first and with the system's highest privileges.

### UEFI is a growing target

The strategic importance and ubiquity of the Unified Extensible Firmware Interface (UEFI) makes it a prime target. Criminal and nation-state groups leverage ransomware, rootkits, implants, and backdoors to maintain persistence and subvert security controls at the UEFI level.

UEFI implants are a rising trend among firmware attacks. In 2020, researchers at Kaspersky Labs identified a UEFI implant known as MosaicRegressor that was used to maintain persistence and deliver additional malware payloads to infected devices. MosaicRegressor is notable because it heavily reused publicly available components from a rootkit discovered five years prior! This is proof that attackers can easily re-package and reuse known implants for new malware campaigns.

But wait — it gets worse. The widespread BootHole vulnerability allows attackers to execute code during the boot process, even when Secure Boot is enabled. This vulnerability is pervasive, affecting most Linux distributions, Windows devices, and any device that uses Secure Boot with the standard Microsoft Third Party UEFI Certificate Authority. BootHole gives attackers a vast pool of potential targets for future rootkits and implants.

### Supply chain breaches are hard to detect

The complexity of technology supply chains introduces many opportunities for risk. Original equipment manufacturers (OEMs) depend on a network of suppliers that often source underlying components from other suppliers. A compromise

at any of these points can put the integrity of the device at risk. Vulnerabilities in any of the components could allow malicious actors to tamper with the device farther along in the supply chain, either during the manufacturing process or at a value-added reseller.

The supply chain can also introduce vulnerabilities. System components are often chosen based on price as opposed to security. Even worse, counterfeit devices, such as fake Cisco networking gear, are quite common and typically contain a wide variety of vulnerabilities. Even firmware within valid components often contains vulnerabilities that can easily be passed on and reused within a variety of products.

There are many examples of breaches in the technology supply chain. Backdoors have been found in enterprise firewalls, telecom gear, and IP security cameras, to name a few. Supply chain breaches are of such great concern that some governments have banned the use of certain technologies in sensitive areas or critical infrastructure.

Threats can also infiltrate the supply chain in the form of updates. In the Sunburst campaign, attackers compromised the update infrastructure of SolarWinds Orion software in order to deliver a malicious backdoor to over 18,000 SolarWinds customers. See more on the tasks of identifying and verifying supply chain firmware in Chapters 3 and 4.

### Connected devices create a growing attack surface

As an attack surface, firmware is growing rapidly thanks to increasing numbers of devices connecting to the corporate network. Most notably, we've seen personal computing and IoT devices continue to proliferate.

A study by VMware Carbon Black found that 91 percent of organizations reported an increase in cyberattacks as a result of employees working from home. A 2021 study from Microsoft showed that 83 percent of surveyed organizations had already experienced a firmware attack.

Organizations are also rolling out IoT initiatives. Every connected device expands the firmware attack surface. And some of these devices are used to monitor critical infrastructure. In

2020, the notorious Mirai botnet experienced a resurgence as attackers took advantage of a vulnerability in F5 BIG-IP controllers to infect IoT and other Linux-based devices.


These are just a handful of threats that put firmware at risk. Attackers continue to innovate by quickly assimilating advanced attack techniques into more-widespread malware and ransomware campaigns. Organizations must have better tools to find firmware vulnerabilities, missing protections, and both known and unknown implants.

Chapter 3

# Identifying Firmware

## In this chapter

- Understand the importance of obtaining visibility into the firmware landscape
- Learn how to gain visibility of devices that don't support a security agent
- Get tips and tricks for proactively managing firmware risk in the supply chain

Addressing firmware risks requires visibility into a wide range of devices, including laptops, servers, networking gear, and more. As security professionals are well aware, obtaining full visibility of IT assets is far from easy. Fortunately, in this case, it is possible. In this chapter we talk about the challenges of obtaining visibility into the firmware landscape and how to overcome them. Along with the next two chapters (on verifying and fortifying firmware), this information can be taken together as a blueprint for creating a sustained, enterprise-wide program to secure firmware in devices of all types.

## Discovery Challenges

The first step in reducing risk in any system is to achieve visibility. You can't protect what you can't see. Similarly, you must be able to see the firmware components that comprise your devices in order to identify vulnerabilities and anomalous behavior in them. New firmware vulnerabilities are common, but it does you no good to know about them if you don't know which of your devices are affected.

At a glance, you must be able to:

- ☑ get fine-grained insight into firmware components within a device, including the version of firmware that's running
- ☑ see the impact of a new vulnerability across all of your devices – both on-premises and remote assets such as devices used by employees working from home
- ☑ assess the severity of threats
- ☑ identify misconfigured hardware settings or out-of-date firmware
- ☑ find guidance on mitigation

## Agent and Agentless

Organizations today are experiencing a lot of change at the device level of the IT environment. Devices are no longer primarily corporate laptops and servers. Instead, security teams are navigating the risk of a constantly evolving landscape of networking equipment, connected devices, and personal employee devices, as well as devices in remote work environments.

Many of these devices can't be managed with traditional security tools. In fact, the vast majority can't support a security agent, which is traditionally used to obtain visibility of an IT asset. As we explained in Chapter 2, many of these devices are the prime targets of firmware attacks. Using an agent-based solution for firmware security will keep a significant portion of the threat landscape in the shadows.


One way to ensure complete coverage of your firmware landscape is to adopt a hybrid, distributed approach to the discovery and monitoring of devices on the network and in remote environments. By leveraging managed endpoint devices that have firmware security sensors, you can often triangulate information within a network segment to continuously discover and assess all networked and unmanaged devices in proximity to these endpoints. This helps you maintain a com-

prehensive, real-time inventory of all devices in your environment, even if introduced outside your procedures. This distributed approach offers comprehensive and reliable visibility into all devices.

# Supply Chain Identification

As we explained in Chapter 1, the supply chain for any given device can be quite complex and extensive. To fully inventory the components and subcomponents, organizations need the ability to identify the various parties in the supply chain.

However, it is imperative that organizations have their own independent visibility into the vulnerability of their device firmware. Because of the many dependencies within the supply chain and the common practice of reusing potentially vulnerable code, getting a code fix can take months or even a year or more.


Keep in mind, software typically only requires an update from a single vendor. Firmware issues, on the other hand, often require coordination across a variety of vendors, with each supplier having to do its own testing. If that vulnerability appears in code that's used across a variety of devices, then your organization remains exposed until each and every device is updated. If you have visibility into the firmware supply chain and can detect potential weaknesses and vulnerabilities, then you can make an informed decision on whether to accept or mitigate that risk.

## *Ensure security across the supply chain*

You can reduce the risk of introducing vulnerable or exploited firmware into your environment by incorporating the following procedures:

- ☑ Perform security scans on newly acquired hardware to verify firmware integrity and identify vulnerabilities.
- ☑ Establish processes to scan hardware introduced during a corporate merger or acquisition.

- ☑ Consider firmware security when evaluating prospective technology and service providers.
- ☑ Evaluate the robustness of vendor firmware update processes and infrastructure. Watch out for vendors that don't require code-signing certificates with their firmware or send update traffic in the clear (unencrypted).
- ☑ Regularly review vendor updates to ensure they are from valid sources and don't have any vulnerabilities.
- ☑ Regularly monitor firmware behavior to detect malicious or anomalous activity.
- ☑ Ensure your procurement and vendor engagement programs assign responsibility for the assurance of third-party components involved in the delivery of products or services. When delegating responsibility to groups outside of your organization, ask for details describing how they manage the firmware risk that you inherit.
- ☑ Add appropriate sections to your runbook on how your organization will deal with potential issues that extend to your vendors and partners via supply chain-related firmware concerns.

## Every SBoM Needs an FBoM

The Software Bill of Materials (SBoM) has become the standard tool for inventorying and tracking the different components, libraries, and code snippets that define a given build of software. In May 2021, the President of the United States issued the Executive Order on Improving the Nation's Cybersecurity, a call for increased cyber vigilance and resources.

A significant portion of Section 4 of the Executive Order is dedicated to defining the requirement that "critical software" must be accompanied by an SBoM to ensure it maintains its intended integrity: "… maintaining accurate and up-to-date data, provenance (i.e., origin) of software code or components, and controls on internal and third-party software components, tools, and services present in software development

processes, and performing audits and enforcement of these controls on a recurring basis…"

This has proven to be especially difficult with embedded firmware found in endpoints, servers, Internet of Things (IoT) devices, and network devices. Still, today's application, development, hardware, and security teams are being asked to cooperate and deliver the same level of transparency and traceability for firmware. This calls for vendors to effectively include firmware in the evolving definition of the SBoM. Think of it as a firmware bill of materials (FBoM).

## Risk Management

An enterprise firmware security strategy should be integrated into your organization's overall risk management program. The strategy should include identifying who is responsible for supporting different types of hardware and evaluating security tools to determine their function and reach into the firmware space. Given that most traditional tools miss the firmware layer, you will likely need to augment your tools and processes to account for firmware. Addressing firmware risks will require visibility into a wide range of devices and components that, in most cases, have been ignored by traditional risk management processes. We'll discuss the build versus buy approach to obtaining firmware security capabilities in Chapter 6.

## Cloud Considerations

There's no easy answer to gaining visibility of firmware in the cloud. It depends on who is taking ownership of and responsibility for securing firmware: you or the cloud service provider (CSP).

Because firmware is closer to hardware than software, you can make the argument that firmware security is the CSP's responsibility. But you need to verify that the provider accepts that responsibility and perform your due diligence during vendor assessment by asking how the CSP obtains visibility into the firmware layer of cloud infrastructure devices.

Unless you're spending millions of dollars with them, large CSPs may not be ready and willing to address your firmware

concerns. If the provider doesn't supply an answer, you may have to accept the risk up to the point where it becomes a compelling reason for a certification. At that time, you can derive comfort from the certification requirement for firmware baselines.

## Cloud Security Provider Relies on Firmware Integrity

In 2018, one of the world's largest providers of security solutions for web infrastructure — with over 100,000 customers, millions of end users, and well over $300 million in revenue — had a pressing problem. They'd read and followed stories highlighting supply chain attacks on major server manufacturers and had themselves been sourcing thousands of Quanta and Gigabyte server systems from overseas. These servers were created with the same kind of supply chains that had fallen victim to critical, firmware-embedded security exploits. In light of the flurry of news publications, the company wanted to establish a method for continuously verifying hardware- and firmware-level integrity in all their systems.

They turned to the firmware security experts at Eclypsium for answers. The result was a custom-designed firmware assessment agent built into the Linux kernels run by the company's Quanta and Gigabyte servers. This micro-agent allowed continuous assessment of firmware integrity and verification against implants. Best of all, the micro-agent had no noticeable impact on performance or availability, which was crucial to the cloud security provider's business.

As the review team reported in their final analysis to the Board of Directors, "Processor-level implants and vulnerabilities have been a nightmare waiting to happen. Eclypsium will help our CIO sleep better at night."

The company can now provide continuous assurance of firmware integrity to not only its 100,000+ customers, but also to its security teams, its head of compliance, and to Wall Street.

Chapter 4

# Verifying Firmware

## In this chapter

- Read about the importance of verifying firmware integrity
- Understand the implications of firmware risk for zero trust
- Learn about compliance requirements for firmware controls

Once you have an inventory of the firmware in your organization, you need to verify its integrity – ensure that it is what it claims to be, nothing less and nothing more. In this chapter, we look at how to verify firmware integrity.

## Integrity: Implants, Tampering, and Counterfeits

Implants, tampering, and counterfeits compromise the integrity of firmware. However, you can't know if firmware has been compromised if you don't know its expected state. Verifying integrity means ensuring that the firmware in your devices hasn't been modified from the authorized code and configuration.


To gain trust at the device level, you must be able to verify that the firmware and boot process of the device are secure and haven't been compromised. You can achieve verification by scanning devices with appropriate tools to ensure that all UEFI and component firmware is legitimate, vendor-approved code, and that it is free from vulnerabilities or implants.

Verifying the security posture and integrity of firmware within enterprise devices includes:

- ☑ Establishing baselines for all component firmware
- ☑ Ensuring firmware is valid, vendor-approved code
- ☑ Identifying devices with outdated or vulnerable firmware or with weak, non-compliant configurations
- ☑ Verifying that firmware has not been altered or compromised by either known or unknown threats


Security teams need the ability to verify the integrity of all newly acquired devices and identify any vulnerabilities before putting them into service. Ideally, this process should extend into the purchasing phase so teams can evaluate prospective devices based on their security posture. We outlined tips and tricks for the supply chain in Chapter 3.

It can seem daunting to extend security to the firmware layers of so many devices. However, there are firmware security tools available that can address these assets.

## Zero Trust Creates New Firmware Security Requirements

Whether due to pandemics and remote work or to the long slow dissolution of formal network perimeters, this is the era of zero trust networking. Zero trust asserts that every connection is untrusted until it has been verified. Each session is assessed from a security perspective before access is granted, and trust is constantly re-evaluated based on session context and risk insight. Even then, access and trust are highly granular, with users and systems only granted access to the specific resources they need at that moment and with the lowest level of privilege possible.

Device-level context and security posture have become a standard part of zero trust-based access decisions, verifying that a connecting device itself can be trusted as part of the access decision. The next evolution of zero trust will go deeper, requiring validation and verification of firmware before a device can be trusted.

An example: in 2021 security researchers discovered widespread high-scoring (cumulative CVSS 8.3) vulnerabilities in the UEFI firmware of Dell systems that affected 129 different models and at least 30 million units.

A working zero-trust implementation would:

- Immediately report the presence of any Dell systems affected by the vulnerable firmware within the organization's laptop fleet.
- Alert cyber security teams and begin limiting access of any systems deemed high risk or with highly privileged users.
- Assist patch and operational teams in verifying the binaries of the firmware updates, installing them, and updating device information.
- Report on the percentage of devices allowed on the network or allowed to have privileged access per the organization's zero-trust policies.

# Vulnerability and Threat Assessments

In addition to verifying the integrity of system and component firmware, organizations need the ability to detect known and unknown threats such as implants, backdoors, and rootkits. Unlike traditional software, firmware and device configurations should remain highly predictable. Any changes could indicate that the firmware has been compromised.


A firmware threat assessment should detect the following:

- ☑ Unknown binaries — leverage a library of firmware to find unexpected or altered firmware.
- ☑ Known threats — detect rootkits, hardware implants, and backdoors by defining your own firmware-specific YARA rules or by leveraging prepackaged rules from vendors.
- ☑ Unknown threats — identify anomalous behavior or functionality that can indicate a possible compromise.

# Firmware Verifications in Supply Chains

Verifying the integrity of firmware across a complex supply chain presents unique challenges. Self-assessment tools and capabilities of device original equipment manufacturers (OEMs), if they're available at all, vary widely depending on the vendor, model, and type of device in question. That makes these chores time-intensive and error-prone.

In addition, recent attacks force the question of how an organization can trust a vendor's tools and checks in the first place when the vendor itself (or one of its upstream component providers) may be compromised. This makes it critical for organizations to have an independent, consistent, and standardized way to verify the firmware in their devices.

- ☑ Independent — Organizations should develop their own vendor-agnostic methods and procedures for evaluating firmware security, and not rely on the reports of individual OEMs.
- ☑ Consistent — These methods and procedures should enable consistent application across different devices — endpoints, edge devices, network devices, and servers — even if the individual vendor-supplied tools are markedly different in their capabilities.
- ☑ Standardized — The same scoring methods for firmware verification should be applied not only across vendors and device types, but also different teams, like endpoint, network security, security operations, and data center teams.

In Chapter 3, we introduce the idea of a firmware bill of materials (FBoM). Supply chain verification of firmware uses automated methods, like binary analysis, to compare observed values against the values defined in an FBoM as a means of verifying the code. The FBoM, in conjunction with shared methods and procedures across defender teams and tools, provides the kind of independent, consistent, and standardized evaluations of firmware across supply chains that are impossible for traditional vulnerability management or endpoint tools.

## Cloud Considerations

If you and your cloud service provider (CSP) have determined that the responsibility for maintaining firmware security falls to you, the customer, then you should approach firmware verification and assessment processes the same way you would for on-premises assets. However, it's much more likely that the CSP is responsible for managing firmware risk. In that case, ask your provider the following questions:

- ☑ How do you verify the integrity of device firmware?
- ☑ What is your process for verifying the integrity of firmware on newly procured devices?
- ☑ How do you verify firmware integrity in reprovisioned hardware? Is firmware being reflashed between users?
- ☑ How often do you assess physical and virtual networking devices for firmware vulnerabilities?
- ☑ What artifacts can be delivered to you, the customer, summarizing this information for devices that will be critical to your operations?

To keep pace with these requirements, you should extend compliance efforts beyond operating systems and traditional software to include assessment and verification of firmware.

## Global Regulations and Frameworks Addressing Firmware

It was only a matter of time before standards organizations and lawmakers caught on to the risk posed by firmware. Sure enough, firmware is now in scope for most modern security frameworks and regulations.

For example, NIST's Security and Privacy Controls for Information Systems and Organizations (SP 800-53), has evolved over five versions to be increasingly inclusive (and wary) of firmware. While the previous Revision 3 had approximately 16 specific references for managing firmware, the latest update — Revision 5 — now includes over 160 references.

A new practice guide, NIST's Validating the Integrity of Computing Devices (SP 1800-34), goes even further by requiring a detailed analysis of the firmware components and security mechanisms to ensure device integrity.

Many other frameworks and regulations now address firmware, like the Cloud Security Alliance Cloud Controls Matrix (CSA CCM), Cybersecurity Maturity Model Certification (CMMC), Department of Defense Instruction 8531.01, FedRAMP, Monetary Authority of Singapore (MAS), and the ubiquitous Payment Card Industry Data Security Standard (PCI DSS Version 4), to name a few.

Chapter 5

# Fortifying Firmware

## In this chapter

- Learn how to overcome the challenges of updating firmware
- Understand why reimaging a system doesn't solve the problem of compromised firmware
- Consider the questions you should ask cloud service providers about their firmware update process

Chances are good that, like most enterprise teams, yours dedicates significant resources to patching and updating operating systems and applications. The same processes and rigor must be extended to the firmware that underpins the fundamental behavior of system hardware. In this chapter, we look at what it takes to implement a firmware update management process.

## Overcoming the Challenges of Updating Firmware

A firmware update management process is a critical piece of any modern security program. In fact, few programs can be successful without it. Unfortunately, carrying out a firmware update process is easier said than done. Updating firmware is time consuming. Infrastructure teams already face a deluge of alerts and updates from the rest of the IT landscape. Firmware exponentially increases their workload.

Updating firmware can also be risky because of the often-precarious relationship between configuration settings, planned use cases, required features and downstream functions. Firmware reconfigurations may require a system reboot and downtime, which can be anathema for critical operational

systems. Unlike operating systems and applications, reinstalling may not fix the problem if a firmware update goes wrong. The complex interactions between firmware, hardware, device drivers, and applications can make the debugging and testing processes long and frustrating. The majority of organizations lack an understanding of what firmware they have in their environment and whether updates are available, let alone the tools to safely test and roll out these updates.

### *Updating, patching, and configuration repair*

Just as you regularly perform software and application vulnerability scanning, you should scan for device-level weaknesses as an automated and ongoing part of your security practice. The scan should look for firmware vulnerabilities, outdated versions, hardware misconfigurations, and missing protections that can be repaired by patches or updates.


Once vulnerabilities have been identified with an established Common Vulnerabilities and Exposures (CVE) number, a firmware security platform can automate updates of firmware in endpoints, servers, and network devices, as well as device-level misconfigurations that can put the device at risk. (Misconfigured hardware settings can leave you open to attacks that are easily prevented.) Rather than simply alerting on issues, the platform can move your organization toward resolutions.

A security team can use the visibility provided by a firmware security platform to identify and prioritize devices whose security needs to be addressed, whether that's by applying updates, quarantining, or repairing firmware configurations. Also, by automating patching and quarantining of affected devices, you can save time while reducing your risk exposure.

## Threat Detection and Response

Real-time threat detection and response are also important elements of an enterprise firmware security strategy. You need the ability to detect and alert on threats that are unique to firmware, such as implants, backdoors, boot loaders, and other malicious code in real time, so that you can respond to them as quickly as possible.

## Detecting firmware threats

Security tools are available to extend threat detection and remediation to the firmware and hardware layers. These tools can validate that all firmware and device-level code match known, valid versions of code from the vendor. They can detect if bits in SDRAM memory have been "flipped," an indication that the machine might have been subject to attacks like Rowhammer or RAMbleed. They can also automatically detect the presence of any known backdoors, implants, rootkits, or other malware. By leveraging indicators of compromise (IOCs) and static, behavioral, and heuristic analysis, these tools can find known or unknown threats and deviations in firmware integrity that warrant deeper analysis.

## Responding to firmware threats

Once a threat is detected anywhere in the IT environment, speed and accuracy are the name of the game. Security analysts don't always have the time to analyze each threat, and regardless of which malware family the threat belongs to, the response is the same: clean the system to restore trust or wipe the system, reinstall the golden image, and send the machine back out for use. However, as malware in the wild increasingly targets firmware, it's essential that incident response (IR) and threat hunting efforts extend to firmware as well.


IR teams face a problem when it comes to infected firmware: reimaging a system doesn't produce a clean slate. System firmware as well as firmware within hardware components such as storage drives, network adapters, etc., all survive independently from the operating system. If an attacker can compromise any of these components, then they can easily persist despite a full reimaging of the system.

Teams need the appropriate tools for responding to firmware attacks. Security teams must be able to easily scan devices for signs of suspicious firmware activity that can be used for both attack mitigation and attribution. IR teams and analysts must be able to scan every device within the scope of an incident to verify that firmware hasn't been modified at either the system or component level. The ability to remotely scan laptops, servers, and even network devices is important to ensure the integrity of every device that was part of an incident's progres-

sion. In addition to installing a fully patched golden image of the OS, check the device to ensure the firmware is up to date.

Malicious implants should be analyzed by forensics teams, and a detailed analysis and reporting of any firmware images should be part of a complete incident response playbook. Analysis and reporting provide digital forensics teams with solid baselines and the evidence they need to investigate the context of an attack, as well as to identify and limit the exposure of a breach.

### Firmware Security Tips & Tricks: Incident Response

- Make it a standard practice to scan firmware in any devices that are potentially compromised.
- Before returning a device to service, use scanning to verify the integrity of all firmware.
- Give your threat hunters tools that monitor for unusual firmware behavior so they can further analyze suspicious devices.
- Identify any gaps in handling firmware-related alerts, both in existing security tools and in your security information and event management (SIEM) or security orchestration, automation, and response (SOAR) solution.
- Add firmware processes to standard IR triage and response runbooks.
- Ensure teams have appropriate tools or services to perform forensic analysis of firmware and to collect artifacts of a firmware attack.
- Evaluate and update the IR knowledge base to include firmware-related information.

## Comprehensive Firmware Monitoring

Continuous monitoring of device firmware helps you stay aware of your security posture at all times. Maintain a complete view of your entire environment or focus on a specific group of devices with insight into firmware and components. Gain visibility into weaknesses and threats to detect risks associated with hardware profile changes, tampering, and compromise.

Threats in the wild such as the ShadowHammer campaign have demonstrated the potential for valid, vendor-supplied updates to be compromised with malicious code. In these cases, monitoring a device's hardware and firmware would advise you of your exposure, while monitoring its behavior could key you into compromised firmware or hardware well in advance of public notice.


By verifying that only valid, trusted code is running on a device, and then monitoring the actual behavior of the code, you can actively confirm the integrity of the device. This information provides the necessary foundation from which all other device-related contexts can be supported.

A firmware security solution should actively monitor the behavior of valid code, rather than relying on third-party observations from the OS, to reveal any signs of unknown threats or malware. These threat-based contexts should be shared with other security tools.

## A Fortune 500 Financial Services Company Embraces Firmware Security

In 2019, the security teams at one of the world's largest financial services companies had a huge opportunity . . . but along with it came a new kind of problem. China opened up as a new market, but there were major concerns about key executives being hacked when traveling there. Gear returning from China could contain rootkits and implants that could spread into domestic networks.

The team recognized that the stealthiest and most persistent implants were those in firmware. The company's cybersecurity team was aware that their vulnerability, endpoint protection, and anti-malware tools couldn't inspect and assess firmware-level components. To take advantage of a global business opportunity, the company needed a fast solution in a new category: firmware security.

Through a trusted advisor, the company was introduced to Eclypsium. Eclypsium's firmware security solution uses lightweight agents to directly observe BIOS, UEFI, and critical chipsets. The solution can also be updated in real time with new firmware intelligence — profiles that determine which firmware is "good" and which is potentially "bad."

After a successful research effort, the company chose to roll out Eclypsium to thousands of high-risk devices that would be making the trek to China and back. Not only were new risks successfully managed and mitigated, but the company has begun exploring options to secure unprotected firmware in servers and network devices.

Chapter 6

# Scaling for the Future of Firmware

## In this chapter

- Understand what you can expect of firmware security in the future
- Learn why government regulations may impact your programs and how they address firmware security
- Explore the challenges of federated device responsibilities in large organizations

By this point, we hope we've reassured you that firmware risk is manageable. However, we understand if the prospect of adding an ever-expanding threat landscape to your risk management program feels daunting. In this chapter, we look at how to scale firmware risk management.

## The Future of Enterprise Firmware Security

It's a simple fact: the systems you verify today may have a vulnerability tomorrow. That's why periodic vulnerability assessments and continuous monitoring are security best practices. They apply as much to your firmware landscape as they do to your operating systems and applications, so you can expect firmware security to become a permanent addition to your risk management program.

However, like everything in technology, we can expect the firmware landscape to change. Yes, more IT services are moving to the cloud. But under every cloud platform are millions

of lines of firmware. You need assurance that cloud providers are doing their due diligence to reduce your risk.

While cloud estates are growing, we don't expect the device landscape to shrink. Security teams will always have network devices, user endpoints, remote and home offices, and personally owned devices connecting to the network. And let's not forget the Internet of Things (IoT). Every sensor deployed in an operational environment has firmware that must be identified, verified, and fortified.

All of the aforementioned devices present a growing threat landscape for enterprises and a lengthening list of opportunities for cyberattackers. Cyber threats and attack techniques will continue to evolve as cyberattackers seize the opportunity to take fundamental control of devices while maintaining persistence and evading security controls. The time to get ahead of firmware risk is now.

## Firmware Security in Government Environments

Governments – whether federal, regional, or local – are no stranger to firmware risk. Federal agencies and their partners are naturally prime targets for advanced state-sponsored attacks as well as opportunistic criminal campaigns. While federal governments have policies and processes in place for validating the supply chain, they are not necessarily scalable or practical.

Consider country-of-origin requirements. Limiting products to those made in your native country doesn't adequately reduce the risk of a firmware attack. A device from a company incorporated in the U.S. was not necessarily manufactured in the U.S. The product very likely has components or subcomponents that were manufactured or coded in other countries. Even if the component or device was developed in the U.S., it's likely that it used an international team of firmware developers or open-source code from developers around the world. Finally, as ShadowHammer demonstrated, security organizations can't always trust their vendors, regardless of origin.

Some federal agencies may be willing to put the extra time and effort into verifying the firmware in critical systems by deconstructing, analyzing, and reconstructing them. That approach simply doesn't scale, especially when a system needs to be revalidated with each software update. Federal governments need a better approach to firmware and device risk management.

# The Need for an Agnostic Solution

Given the scale, scope, and severity of firmware risk, security teams need a tool to help automate the processes we've described. Organizations often prefer to consolidate their technology investments under a single vendor. However, this is one instance where a vendor-agnostic solution is a must-have.

## *Standardizing firmware tools*

From a purely practical standpoint, a vendor-agnostic solution is the only acceptable option security organizations have for managing firmware risk. The heterogeneous nature of the IT environment won't permit anything else.


For any given device category, organizations tend to rely on more than one vendor. And for any given vendor, they tend to select more than one model – even when it's the same device. Vendors may offer tools to validate the supply chain, but they differ from vendor to vendor, and some even differ from one device model to another. The variety and scope of tools make it almost impossible to implement a consistent and efficient approach to verifying firmware. Managing firmware risk with a vendor-agnostic tool allows you to centralize risk management efforts.

## *Obtaining verification*


There's another problem with relying on vendor tools to verify the integrity of the supply chain, and it relates to the issue of implicit trust. As the Sunburst attack against SolarWinds reminded us: vendors are not infallible.

### Remove single points of failure

Attackers infiltrated SolarWinds and inserted malicious code into valid, properly signed software updates. Customers updated their software with the compromised code. The takeaway: you can't rely on a vendor to verify its own integrity when it might be compromised. Independent verification removes the single point of failure.

### Revalidate after each update

The SolarWinds incident also highlights the importance of revalidating firmware after each software update. Regular device updates can't be avoided. An independent view into the device and its firmware enables a third-party solution to verify the information self-reported by a device and to perform behavioral analysis to detect signs of compromise.

### Avoid vendor lock-in


A vendor-agnostic firmware security tool will also help you avoid vendor lock-in. This is important, considering the rapid changes occurring in IT. While vendor tools provide insight into source code and technology roadmaps, heterogeneous environments require heterogeneous solutions. A third-party solution will ideally support you through whatever changes come along.

## *Build vs. Buy*

The need for a vendor-agnostic firmware security tool raises the question: do you build it in house or buy a solution? As you consider your options, keep the following in mind:

- ☑ The vendor and device landscape is expanding. A third-party tool provides a consistent, automated approach across a heterogenous environment.
- ☑ Your firmware security tool will also come under fire. Building a custom solution will put an even bigger target on your back.
- ☑ New firmware vulnerabilities and attack methods appear in the wild almost daily. With a custom solution, it's up to you to correlate data to identify those vulnerabilities before it's too late.
- ☑ Firmware protection requires deep, automated analysis and vulnerability detection that continues to grow increasingly sophisticated.

Chapter 7

# Selecting the Right Solution

## In this chapter

- See a blueprint for creating your firmware security RFP
- Review the importance of enterprise-wide discovery
- Understand the value of component inventories and baseline profiles
- Learn about all the capabilities available in a single firmware security solution

If you decide to buy a firmware security solution rather than build a custom solution, then you face the task of evaluating products in a security category that is still young. In this chapter we advise you on the top requirements for an enterprise firmware security tool.

## A Global Database of Firmware and Hardware Profiles

Traditional software is always changing, but firmware should remain predictable. One way to ensure that firmware is in a "known good" state is to compare it to other firmware — the more the better. Look for a firmware security solution provider that checks firmware against millions of firmware hashes across scores of international hardware vendors in a regular, traceable way. This process will help identify correct versions, deviation from baselines, outdated firmware, and tampering.

## Support for Both Agent-based and Agentless Devices

To optimize your efficiency, look for a single solution that can provide complete visibility of your device landscape. This means choosing a solution with a multi-platform, hybrid architecture to support both agentless and agent-based devices running Linux, Windows, MacOS, Cisco IOS, and iOS, as well as network devices, connected OT and IoT devices, and servers.

## Wide Discovery of Both Managed and Unmanaged Devices

Anything short of complete visibility and security coverage leaves your organization exposed to a firmware-based attack. It's therefore important that you have visibility into every device connecting to your network – including personally owned and remote devices that users connect with from home or the field. Look for a solution that can discover a wide variety of devices and that leverages distributed discovery to remove the blind spots represented by connected but unchecked devices.


Recent, global events have obliterated traditional work/life and home/office boundaries. For knowledge workers especially, very little distinction remains between home networks, business networks, and cafe networks. To protect privacy, the discovery capabilities in your firmware security solution should have the option to be disabled when scanning public networks.

## Component Inventories and Baseline Profiles

A firmware security tool should provide a comprehensive view that simplifies firmware risk management. At a glance, you can see your entire estate and associated risks and understand your firmware security posture. You can see the impact of a new vulnerability across all your devices, assess the severity of threats, identify misconfigured hardware settings or out of date firmware, and obtain guidance on mitigation.

## Ability to Assess Firmware Integrity

A firmware security solution should automatically verify that firmware has not been unexpectedly altered or compromised by either known or unknown threats. A detailed integrity analysis provides assurance that firmware has not been compromised. The tool should also conduct an automatic risk analysis to identify critical vulnerabilities.

## Ability to Assess Compliance with Standards or Frameworks

The majority of regulatory requirements already apply to firmware but if your organization is like most, then you probably lack the tools and experience to assess and measure compliance. Take this opportunity to address those gaps. Consider a solution that equips you to assess your firmware security vulnerabilities and risks, take action, and demonstrate your compliance with regulations and industry standards down to the firmware and hardware level.

## Ability to Automate Firmware Updates

Tools can help streamline the firmware update process, which otherwise can be slow and laborious. Ideally, the same solution that validates firmware integrity also accelerates firmware patching and updating efforts, with the ability to roll back these changes if necessary.

## Ability to Configure Firmware for Security Best Practices

In addition to update and patch management, the firmware security solution should address configuration management. This means finding configuration issues, such as disabled BIOS write protections or unlocked components, which can put a device at risk. It also includes fixing configurations according to security best practices.

## Ability to Detect Active Threats

To accurately detect both known and unknown threats or changes to firmware integrity, a firmware solution leverages a variety of techniques, including indicators of compromise and static, behavioral, and heuristic analysis. The tool's analytics server should also leverage industry-leading threat and vulnerability research. The solution should alert you when implants, backdoors, and other malicious code have compromised your IT infrastructure. Look for a tool that can help prevent damage when threats are detected by, for example, by leveraging robust APIs to automate orchestration efforts such as quarantining affected devices.

## Enterprise-class Scalability and Integrations

To be effective, firmware risk management must become a normal part of your larger program. Seamless enterprise integrations will help you incorporate firmware security into existing workflows. For example, look for a solution that enables you to visualize event data through syslog or major security information and event management (SIEM) providers. A rich set of APIs will enable you to integrate with your other security solutions, your asset inventory and management systems, and your vulnerability management solution, and feed telemetry to MSSP solutions.

Enterprise-class scalability will ensure that visibility and coverage continue uninterrupted in periods of rapid growth.

## Choosing Your Next Steps

If this book has done its job, you have a much better sense of the threats, protections, and practices needed to get ahead of adversaries and secure the growing attack surface represented by your enterprise firmware. But what do you do next? Here are a few suggestions, depending on your role.

**If you're the CISO**

CISOs have multiple programs to maintain. It should be clear now that a number of these programs – like supply chain security, risk and vulnerability management, endpoint protection, network security, and threat detection and response – might be missing a critical firmware-centric perspective that's not being missed by adversaries. We suggest the following next steps:

- Survey your teams for their understanding of firmware threats and their plans for addressing them.
- Mobile them to research how their discipline can extend into securing firmware.
- Assess tools for gaining visibility into the security posture of firmware.

**If you're the security strategist**

When you think like an attacker you realize that organizations have done a pretty good job of securing operating systems and applications and the space "above" firmware and hardware. To attackers, firmware can look like the weakest link. Your next steps should be to:

- Assess your firmware supply chain and understand how firmware integrity will be assured.
- Evaluate the growth of your firmware attack surface: how many firmware components are in your endpoints? Your servers?
- Make sure the software bill of materials (SBoM) for your enterprise solutions includes a firmware bill of materials (FBoM) for hardware and devices (see Chapter 3).

**If you're the IT risk and compliance manager**

You already have regulations, standards, or frameworks – whether PCI DSS, NIST, or CIS 18 – to act as your roadmaps for maintaining good information security practices. But these are "living" documents that are constantly being revised. Take these next steps to address firmware:

- Review your standards for recent changes that include new firmware requirements and references.
- Ask your auditors about firmware considerations.
- Expand the scope of your assessments to make room for firmware inventories and security measurements.

**If you're the incident response analyst**

Remember that firmware is very similar to the software and programs

you are evaluating for threats and malicious activity, but far more opaque and often less observable. Going forward, do the following:

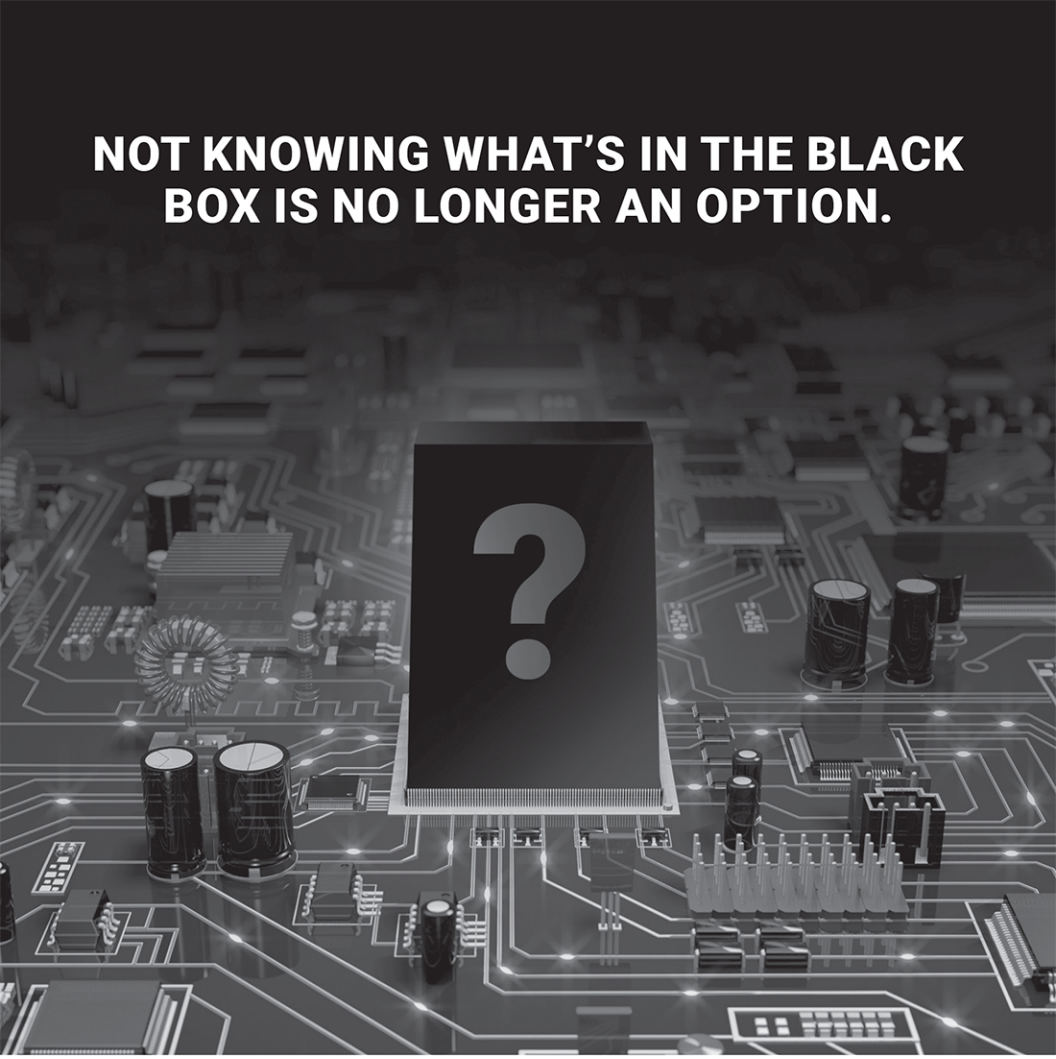
- Understand firmware indicators of compromise (IOCs) and tune your sensors and systems to detect these (See Chapter 5).
- Review incident response plans to account for reflashing of firmware, or to indicate what to do if firmware integrity can't be assured.
- Create a firmware compromise runbook for your most critical systems.

**If you're the product owner for VA, endpoint, threat, or network tools**

Hopefully this book has demonstrated that there's a growing gap between what practitioners believe tools can do and what they can really achieve.

- Ask your vulnerability management provider if they can identity and flag firmware vulnerabilities, and with what degree of accuracy.
- Use recent UEFI- and hardware-level vulnerabilities (like ESPecter and FinSpy) to test your endpoint and threat protections.
- Consider firmware in the context of the Identify, Verify, and Fortify approach: can you achieve these outcomes with your current tools? Some of them? Which ones?

# NOT KNOWING WHAT'S IN THE BLACK BOX IS NO LONGER AN OPTION.




Firmware is your organization's fastest-growing and most unprotected attack surface.

Eclypsium identifies, verifies and fortifies firmware throughout the enterprise, from endpoints and network devices to servers and virtual systems.

Try it! QuickScan is a free and easy way to get immediate details on firmware vulnerabilities and integrity in X86 systems.

**eclypsium.com/quickscan**

eclypsium®

## Discover the enterprise's vast, unguarded firmware attack surface, how to assess it, and how to defend it.

The security industry has made great strides in reducing cybersecurity risk in operating systems and applications. But the underlying firmware, the digital DNA of every device, has largely been ignored — by everyone but attackers. With firmware threats and exploits increasing, security teams need a way to defend this unguarded attack surface. This Guide shows them how to do just that.

- Understanding firmware — review what firmware is and what it does.
- Weaponizing firmware — understand why attackers target firmware and why traditional security controls fail to protect this growing attack surface.
- Identifying firmware — learn about the challenges of discovering and inventorying firmware in the enterprise.
- Verifying firmware — examine the importance of integrity and vulnerability assessment related to firmware.
- Fortifying firmware — explore firmware update and threat detection and response processes.
- Scaling for the future of firmware — know how to manage firmware risk to protect the enterprise today and tomorrow.

***About the Authors***

A former editor of SearchSecurity.com, Crystal Bedell is a senior marketing consultant specializing in cybersecurity. She's been helping technology providers create engaging content since 2000. Michael Thelander received CISSP training through SANS and has been a cybersecurity product manager and product marketer for over fifteen years. His articles and interviews have appeared in *SC Magazine, Cyber Defense Magazine, ITProfessional, Cyber Security: A Peer-Reviewed Journal,* and other publications.

CYBEREDGE PRESS

Not for resale

ISBN 978-1-948939-26-3