WHITEPAPER

Firmware Security: How to Identify and Prevent Vulnerabilities

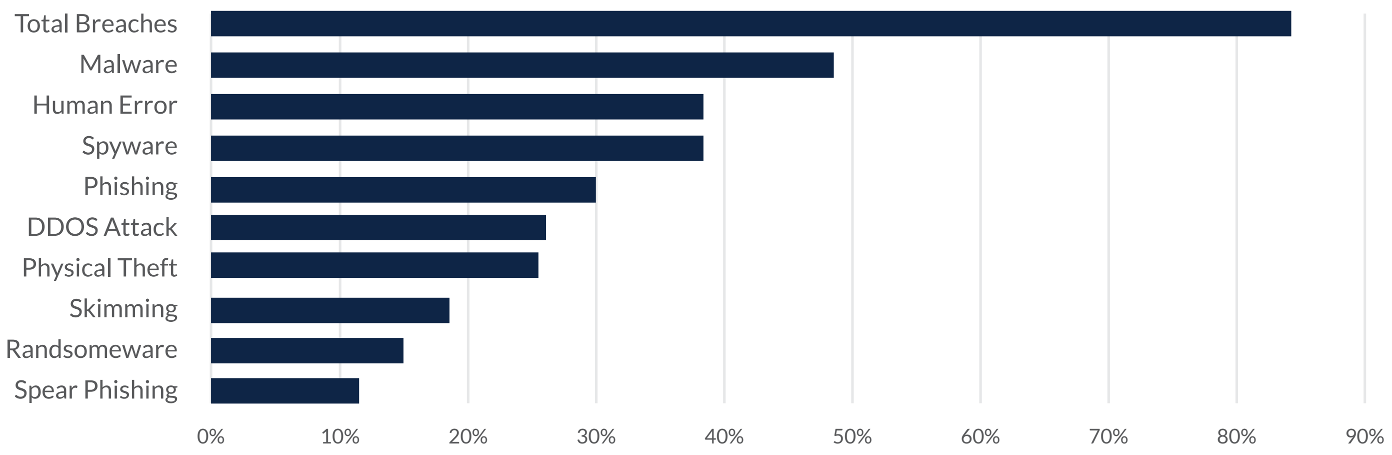
Security Compass

When engineering and software security professionals think of security, the focus is often on preventing adversaries from successfully attacking web applications. This makes sense, as web applications are the predominant target of hackers.

However, software is only a portion of the attack surface an adversary can target. Whether the goal is to steal sensitive information, disrupt normal operations, or cause harm to infrastructure, hardware and product security must also be considered. While attacks on applications and database often garner the most headlines, attacks on systems and hardware result in data and physical consequences:

- An attack on a steel mill in Germany resulted in an abnormal shutdown of a blast furnace, resulting in massive damage.
- Industrial control systems manage large scale manufacturing processes. In 2010, the Stuxnet attack on Iranian nuclear facilities resulted in damage to thousands of centrifuges used to refine uranium.
- Safety instrumented systems (SIS) are used across the Oil and Gas, Chemicals, and public utility industries. The 2017 Triton/Hatman attack on a safety monitoring system accidentally triggered the shutdown of an industrial process at an undisclosed organization.The attacks are not limited to nation state attacks on infrastructure. The rapid growth of Internet of Things (IoT) and Industrial Internet of Things (IIoT) devices brings new challenges. Each of these devices have an operating system and software or firmware, and each represents a potential attack vector. A recent report from Dell found that 63% of the surveyed organizations experienced at least one data breach in the previous year due to hardware security vulnerabilities.

84% of IoT adopters have experienced a security breach



***Source:** The Internet of Things: Today and Tomorrow, Aruba*

Secure software needs secure hardware

The NIST's Hardware-Enabled Security standard draft states that:

> "The foundation of any data center or edge computing security strategy should be securing the platform on which data and workloads will be executed and accessed."

Organizations building devices must look beyond traditional security testing methodologies to address product security and anticipate threats that are unique to this segment. This must include hardware security — protecting devices from cyber and physical attacks.

New threats require new thinking

It is clear the threat landscape has changed. As software adds more value to products and connected systems, security teams need to consider additional challenges.

Regulatory pressure is building

Complying with the myriad of regulatory standards can be challenging for software development teams. While most organizations are accustomed to complying with common regulatory standards like the PCI DSS, HIPAA/PIPEDA, and GDPR, a new regulatory landscape is developing around hardware security.

Some are very prescriptive. For example, the PCI-DSS for software processing credit card information requires organizations to test for specific types of vulnerabilities such as those enumerated in the CWE Top 25 and OWASP Top 10. Others provide no guidance at all. Section 5 of the FTC Act simply requires "reasonable security," California's SB-327 targeting IoT devices requires manufacturers of any connected device sold in California to have "reasonable security features," and GDPR requires both "Privacy by Design" and "Privacy by Default."

Hardware companies are, of course, in scope for multiple standards. Medical device manufacturers are subject to the UL-2900 standard adopted by the Federal Drug Administration (FDA) for network connected medical devices. More recently, the ISA/IEC 62443 security standard was developed to help build more secure industrial automation and control systems (IACS) and "define a set of engineering measures to guide organizations through the process of assessing the risk of a particular IACS and identifying and applying security countermeasures to reduce that risk to tolerable levels." Likewise the ISO/SAE FDIS 21434 Cybersecurity Engineering Standard is under development for road vehicles, no doubt prompted in part by the 2015 Jeep hacks.

Software and hardware development life cycles are not integrated

Hardware and software engineering teams usually work separately. Often it is a supplier/customer relationship (for instance, automotive infotainment, and navigation software and operating systems for controllers). This can result in a focus solely on functional requirements and use cases, ignoring security requirements and misuse cases.

Secure coding standards are difficult to follow

Many companies have secure coding standards, often in a static document that application and firmware developers can refer to for guidance. This suffers from a number of shortcomings:

- **Proprietary processors and languages:** Software embedded in hardware varies with the processors and runtime environment. These can also be written in C, B#, Embedded C++, and other less well-known languages compared to web applications. An organization's secure coding standards may be less well-defined for the unique needs of some languages.

- **Consistency:** In a perfect world, each development team will follow secure coding guidelines equally well, building a more secure, more easily maintained application. Consistency is difficult when attempting to follow standards maintained in a file or spreadsheet. Developers cannot always remember every rule for every use case, nor the regulatory requirements of overlapping standards.

- **Scalability:** A standard for a single application — with diligence — can work for one or two applications. However, standards are not the same for every application; an internal application with a limited attack surface processing public data does not warrant the same level of security as an internet-facing application processing personally identifiable information.

- **Auditability:** Written secure coding standards may seem like a simple way to track risks and controls, but, from a regulatory compliance standpoint, they lack auditability. Without a central, controlled repository for risk assessment data, with an auditable record of changes, it is difficult to prove compliance to an auditor or corporate board.

Testing for hardware security is different

Web applications can be launched in a staging environment, and tested for functional completeness and with a variety of application security testing tools. Testing hardware requires a different set of skills, as it must include information to ensure a device's life in a harsh environment (in the field, on the water, or in extreme temperatures). Some hardware may even require destructive testing to understand a device's mean time between failures.

Patching hardware vulnerabilities is a bad option

Software development organizations have embraced rapid development methodologies like Agile, Continuous Integration/Continuous Delivery, and DevOps. These allow teams to push dozens of new releases to production each day.

While this is the norm for web applications, it is the last choice for hardware. One cannot simply replace a circuit board, waveguide, or chipset in an existing design. Each change can require recalculating component specifications, sourcing new parts, reprogramming surface mount component pick-and-place assembly equipment, and revalidating QA processes. Once hardware is in the field, it stays there, barring a complete product recall.

Hardware and software security guidance

Teams building secure hardware and software require knowledge of both the functional and non-functional requirements of their respective engineering tasks — before they begin development. This includes:

- **Adopting secure development standards:** For medical devices, engineering must consider FDA guidance for premarket and post-market submissions. For devices like CPUs, Field Programmable Gate Arrays, and Application-Specific Integrated Circuits this can include required or acceptable key strengths, hash algorithms, cryptanalytic resistance, and physical attack countermeasures.

- **Anticipating threats:** Regulatory standards like HIPAA/PIPEDA as well as ISA/IEC 62443 require organizations to assess risk and adopt appropriate controls. Identifying threats or threat modeling is a process that draws expertise from security, architects, and development to identify threats in a project before the development process begins. While traditional threat models can take weeks to complete and scale poorly, DevOps tools like SD Elements can support the goals of threat modeling by identifying security weaknesses and corresponding mitigations for a particular project in a matter of hours.

- **Translating threats and requirements into controls:** By identifying threats up front, teams are able to identify security controls to mitigate risk. Comprehensive hardware security standards should include approved controls for each identified threat. This should include test plans to validate the proper implementation of controls.

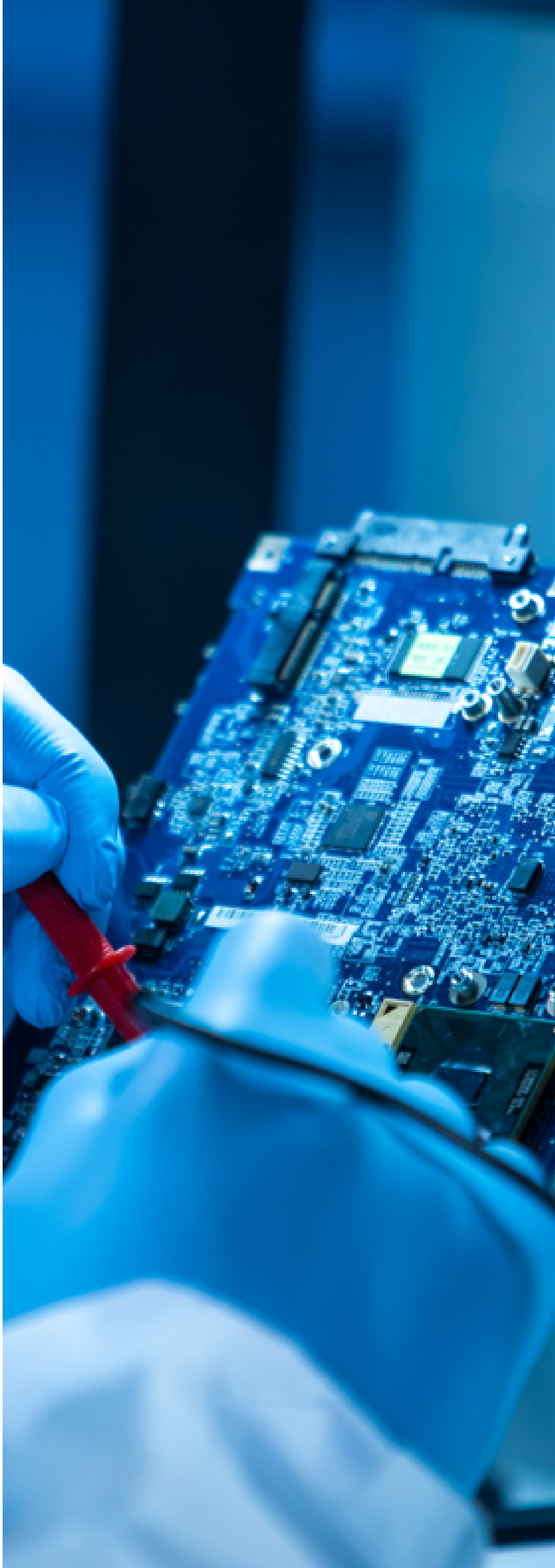
- **Building controls into engineering workflow:** Relying on hardware or software engineers to remember each required control for every project is not practical. Instead, teams should automate the process of assigning identified controls to engineering using their existing scheduling tools and test plans.

- **Using testing to validate controls:** Testing for security inevitably leads to friction between security and development teams. A better solution is to use test cases to ensure that all identified controls were properly implemented. By identifying each test case in advance, developers can better understand requirements and use/misuse cases.

Prioritizing firmware security to avoid breaches

Product companies need to prioritize the security of their systems, both software and hardware. SD Elements ensures that threats to hardware and software are identified, mitigated, and validated. It anticipates threats and provides product development teams, security, and operations with actionable tasks to mitigate risk. This means that security testing is primarily validating that prescribed controls were implemented correctly instead of acting as a primary vulnerability discovery activity.

The result is a balance between speed and security. SD Elements allows companies to build products nearly as fast as if they were being built without any security or compliance at all and as safely as if it were built under the guidance of human experts.



Security Compass

Go Fast. Stay Safe.

Security Compass, a leading provider of cybersecurity solutions, enables organizations to shift left and build secure applications by design, integrated directly with existing DevSecOps tools and workflows. Its flagship product, SD Elements, allows organizations to balance the need to accelerate software time-to-market while managing risk by automating significant portions of proactive manual processes for security and compliance. SD Elements is the world's first Balanced Development Automation platform. Security Compass is the trusted solution provider to leading financial and technology organizations, the U.S. Department of Defence, government agencies, and renowned global brands across multiple industries. The company is headquartered in Toronto, with offices in the U.S. and India. For more information, please visit www.securitycompass.com.

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS

OFFICES

GLOBAL HEADQUARTERS
1 Yonge Street
Suite 1801
Toronto, Ontario
Canada M5E 1W7

TORONTO
390 Queens Quay W
2nd Floor
Toronto, Ontario
Canada M5V 3A6

NEW JERSEY
621 Shrewsbury Avenue
Suite 215
Shrewsbury, New Jersey
USA 07702

CALIFORNIA
600 California Street
San Francisco, California
USA 94108

INDIA
#4.07
4th Floor, Statesman House
Barakhamba Road, New Delhi
India 110001