Product & Supply Chain Security for Device Manufacturers

Firmware isn't safe unless it's safe by design.

With over 20 billion internet connected devices today, connectivity is changing the world. For manufacturers, the ability to create new IoT, OT, and other connected devices and embedded systems has far outpaced their ability to keep those devices secure.

The problem starts with device firmware. This onboard software embedded in the memory of connected devices can perform many functions once controlled by physical systems. For instance, if you're driving a connected car, firmware may automatically control your brakes or steering. If you're using a wearable insulin pump, firmware controls the flow of medication.

As malicious actors continue to exploit weak links, device manufacturers and end users alike are scrambling to secure connected devices and embedded systems. The biggest hurdle for device manufacturers: a complex, opaque software supply chain.

Just like a car is assembled from parts, firmware is assembled from thousands of software components, including drivers, operating systems, bootloaders, and configuration files. Typically, device manufacturers source 80-95% of these components from third-party vendors or open source software. Once a component has been incorporated in a firmware version, device manufacturers often lack visibility into which versions and products it was used. When one component has a security flaw, that flaw is baked into the finished product without the device manufacturer's knowledge.

Here's the good news: **Almost everything we need to know in order to mitigate device risk can be found within its firmware.**

The Billion Dollar Question

If a zero-day vulnerability is discovered tomorrow in an open-source library that you believe was used in some device firmware, how long would it take you to determine which devices and firmware versions were impacted?

Key challenges:

Regulatory risk due to the ever-changing landscape of cybersecurity compliance.

Time-to-market challenges caused by customers and regulators demanding proof of security.

Costly and unscalable manual testing with little to no insight across product portfolio.

Third-party and open source risk, including legal risk from unknown or expired licenses.

Lack of device-specific tooling. AppSec tools don't have the ability to analyze and support embedded system architectures, tools, and binary formats.

Competitive pressure from companies with mature product security programs.

FINITE STATE

You can't protect what you can't see.

By analyzing binaries after they are compiled and ready to be flashed onto a device, Finite State is capable of detecting more types of security issues from more sources than traditional application security approaches. The Finite State platform can identify potential security issues including:

Hard-coded credentials

When passwords or other authentication is hard-coded in firmware, attackers can use these credentials to gain trust and access to key systems.

CVEs

Known vulnerabilities often exist for years in firmware code, leaving potential backdoors open to black hats or even nation-state level attackers.

Unsafe function calls

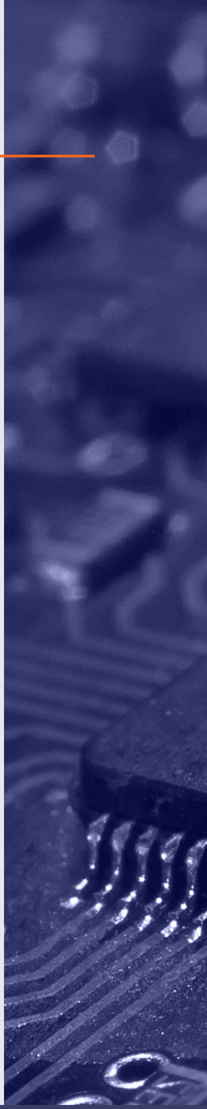
Code containing unsafe function calls can leave firmware vulnerable to injection attacks that can cause denial of service, privilege escalation, or full takeover of the device.

Cryptographic material

Poorly configured systems may contain files such as private keys that may serve as a backdoor or let attackers impersonate the system, and expired or self-signed certificates can present weaknesses attackers can use to compromise the system.

Insecure configurations

Some security issues are generated during the build process, such as not using compiler flags for exploit mitigations. Binary analysis allows these weaknesses to be identified and remediated. Other security issues may be introduced after the build process, such as network facing service configurations that leave the device open to attackers.



The Finite State Platform: Benefits At a Glance

Reveal hidden issues

Discover hard-coded credentials, known open source vulnerabilities, configuration errors, and crypto materials for every version of every firmware.

Create comprehensive SBOMs

Know the composition of every firmware version, so that components containing newly-identified zero-day vulnerabilities can be traced and patched quickly.

Manage supply chain risk

See where your code is coming from and how it could expose you to risk, allowing you to make informed component sourcing decisions.

Reduce time-to-market

See every firmware version for easy comparison, to ensure that new firmware versions are secure before firmware updates and security patches.

Prioritize remediation

Learn which security issues could have the biggest impact so fixes can focus on the problems most likely to pose major business and customer risks.

Comply with evolving standards

Prepare your product and organization for standards pursuant to Biden Executive Order 14028, as well as industry-specific standards and regulations.

www.finitestate.io

FINITE STATE