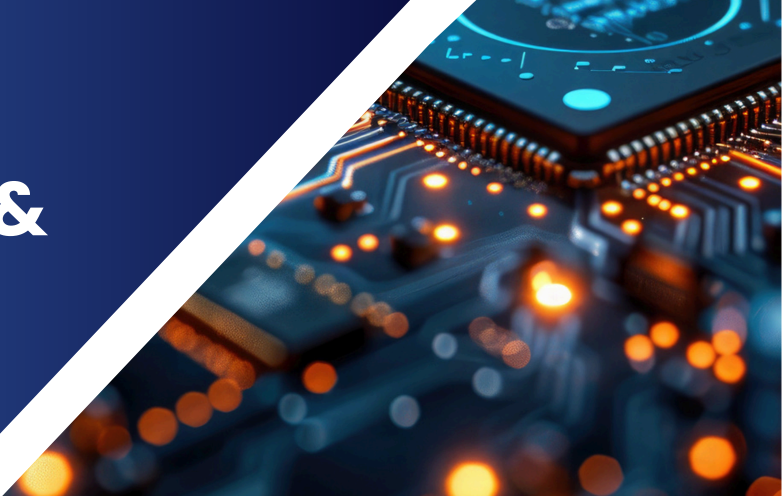
FINITE STATE

Security Scanning & Analysis with Finite State



Manufacturers face the critical task of ensuring their products are secure and compliant with global regulations. Yet traditional security tools are ill-equipped to handle the complexity of connected devices.

Visibility into Any Connected Device + Application across the SDLC

Finite State combines advanced binary software composition analysis (SCA), source code SCA, and binary static analysis security testing (SAST) for unique visibility into connected systems throughout the DevSecOps lifecycle.

Our Approach

> **Comprehensive Analysis**: Analyze virtually any type of code or binary - software, firmware, operational technology - regardless of its origin or format.

> **Deep Dive into Complexity**: Dissect even the most tightly integrated, complex software like monolithic and statically-linked binaries.

> **Unmatched Accuracy**: More true positives and fewer false positives, for the most accurate picture of device and connected portfolio risk.

> **Broad Coverage**: Protect your entire connected product portfolio, regardless of complexity or age. From individual components to complete systems, or from legacy products to cutting-edge IoT devices.

Compatibility

We cover a wide range of languages and architectures, including the most popular programming languages (Java, JS, .NET) and those preferred for connected device development (C/C++)

Key Capabilities

Binary SCA
Finite State dissects the structure of firmware images to extract critical information on components and their dependencies. Analyzing strings, symbols, function names, and control flow graphs, we precisely identify software elements, including their versions, and correlate them to known vulnerabilities. We do this while taking into account the diverse nature of software components, the nuances in naming conventions, and the intricacies of different software architectures. The process yields an inventory of software components within a binary linked to known vulnerabilities, helping you identify and proactively respond to emerging risks.

What You'll Uncover

> Vulnerabilities in custom code, third party software, and legacy systems

> License risk from unauthorized, insecure, or non-compliant software components

> Security weaknesses, hidden vulnerabilities not apparent in source code, and credential detection

> Conflicts, outdated versions, and potential performance issues within your dependencies

> Minimal false positives and more true positives

Source Code SCA
Complementing our binary methodology,
Finite State employs source code SCA to identify known security vulnerabilities and licensing issues in third-party libraries and dependencies, down to the transitive dependency. We automatically identify software components and their versions, analyze the relationships between components, and use this insight into the composition of your codebase to recommend actions to improve the code quality and maintainability of your software.

Static Application Security Testing (SAST)
Binary SAST looks into the internal structure of the binary to uncover additional weaknesses, beyond known vulnerabilities, in third party code. This process involves disassembly of binary files followed by in-depth data flow analysis. It is not intended as an alternative for source code SAST for first party code.

A Platform for Action, Not Just Visibility

> Policy enforcement that break builds or automatically creates policy violation tickets

> Integrations with 150+ security tools for faster remediation

> Upgrade recommendations with Auto PRs

> CI/CD pipeline workflows

Ready to Secure Your Connected Devices?
Learn more at finitestate.io