A Call for a Digital Golden Dome

Operational Software Assurance extends security from development through years of real-world operation.

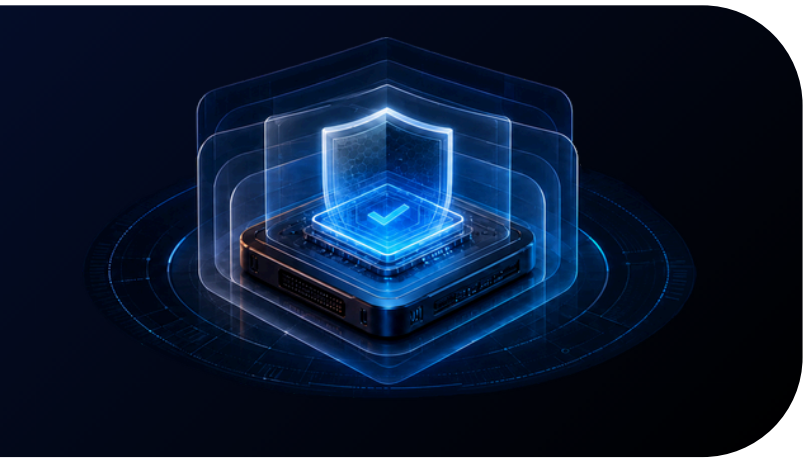
THE BOTTOM LINE: Software security cannot stop when a product ships. Medical devices, vehicles, industrial equipment, energy systems, and connected products may operate for 10-30 years while vulnerabilities continue to emerge. Operational Software Assurance helps companies identify risk, protect deployed software, and continuously prove security across the product lifecycle, including when patching is slow, disruptive, or impractical.

THE CATEGORY: OPERATIONAL SOFTWARE ASSURANCE

Operational Software Assurance (OSA) is the market focused on keeping deployed software secure and provable throughout its life cycle, delivered through three pillars: Identify, Protect, and Comply. Embedded Runtime Security (ERS) is the discipline within Protect that adds active defenses to deployed software.

THE PROBLEM

About 70% of serious software vulnerabilities are memory-safety defects (Microsoft and Google, via CISA), and they sit in the C/C++ code that critical infrastructure is built on. A newly discovered vulnerability can trigger months of engineering work across investigation, patch development, testing, validation, supplier coordination, and deployment. In regulated and safety-critical industries, changes may also require additional verification or recertification. Meanwhile, many deployed systems cannot be easily taken offline or updated at all. The result is a growing period when a vulnerability is known, but the product remains exposed.



WHY NOW: AI BROKE THE DISCOVERY CURVE, CYBER WARFARE, AND REGULATORY PRESSURE

Three forces are converging on long-lived software at once. In 2026, AI-assisted analysis of widely used open-source software confirmed memory-safety defects across major operating systems and browsers, including defects in code that had been audited for years. AI is finding these vulnerabilities and helping attackers develop exploits at machine speed.

At the same time, geopolitical tensions are putting critical systems in the crosshairs and regulations like the Cyber Resilience Act are making manufacturers responsible for security long after products ship. There is a widening gap between how fast risk emerges and how fast teams can patch, test, certify, and deploy fixes. Security has to keep working in that gap.

THE SOLUTION: PROTECT SOFTWARE ALREADY IN THE FIELD

Software memory protection makes an exploit fail at the moment it runs. Runtime techniques such as Load-time Function Randomization scramble a program's memory layout so that buffer overflows and return-oriented programming chains fail, with **no source-code changes, no compiler changes, and no disturbance to the system's certification**. The government already recommends it. The June 2025 NSA and CISA guidance endorses runtime mitigation for code that cannot be replaced, and FY2026 appropriators directed CISA to support memory-safe protections for critical open-source software. For products with long lifecycles, constrained update windows, or certification requirements, that means security teams can reduce exploitability while normal remediation continues.

WHAT THE FIELD EVIDENCE SHOWS

Measure	Result	Basis
Mitigated vulnerabilities, all severities	**49%**	More than 2,000 vulnerabilities were analyzed in a fielded medical device
Mitigated vulnerabilities, critical	**77%**	Same analysis; most critical findings were memory-safety related
Exploitability reduction	**greater than 98%**	Usable ROP chains on software with 1.6M candidate gadgets, CReASE method, Linköping University

Published field measurement. The first two rows count flaws neutralized; the third measures how far the odds of a working exploit fall. They are not interchangeable.

THE PAYOFF

Business Priority	OSA Impact
Product security	Reduce the likelihood that a vulnerability becomes a successful exploit
Engineering efficiency	Reduce dependence on emergency patching and extensive code changes
Product availability	Protect systems that cannot be frequently taken offline
Compliance	Maintain current SBOM, vulnerability, mitigation, and risk evidence
Product lifecycle	Extend security beyond development into years of field operation

THE EXECUTIVE ASK

Make software assurance part of the entire product lifecycle. Know what is inside deployed software. Understand which vulnerabilities create meaningful risk. Protect software while it is running. Maintain evidence that demonstrates how risk is being managed.

For any long-lived connected products, the goal is to build safer new software without forgetting to protect the software already operating today.