WHITE PAPER

Built on Trust:
How Leading Manufacturers Are Securing the Software Supply Chain

The software supply chain risks modern manufacturers don't see coming, and how industry leaders are closing the gap.

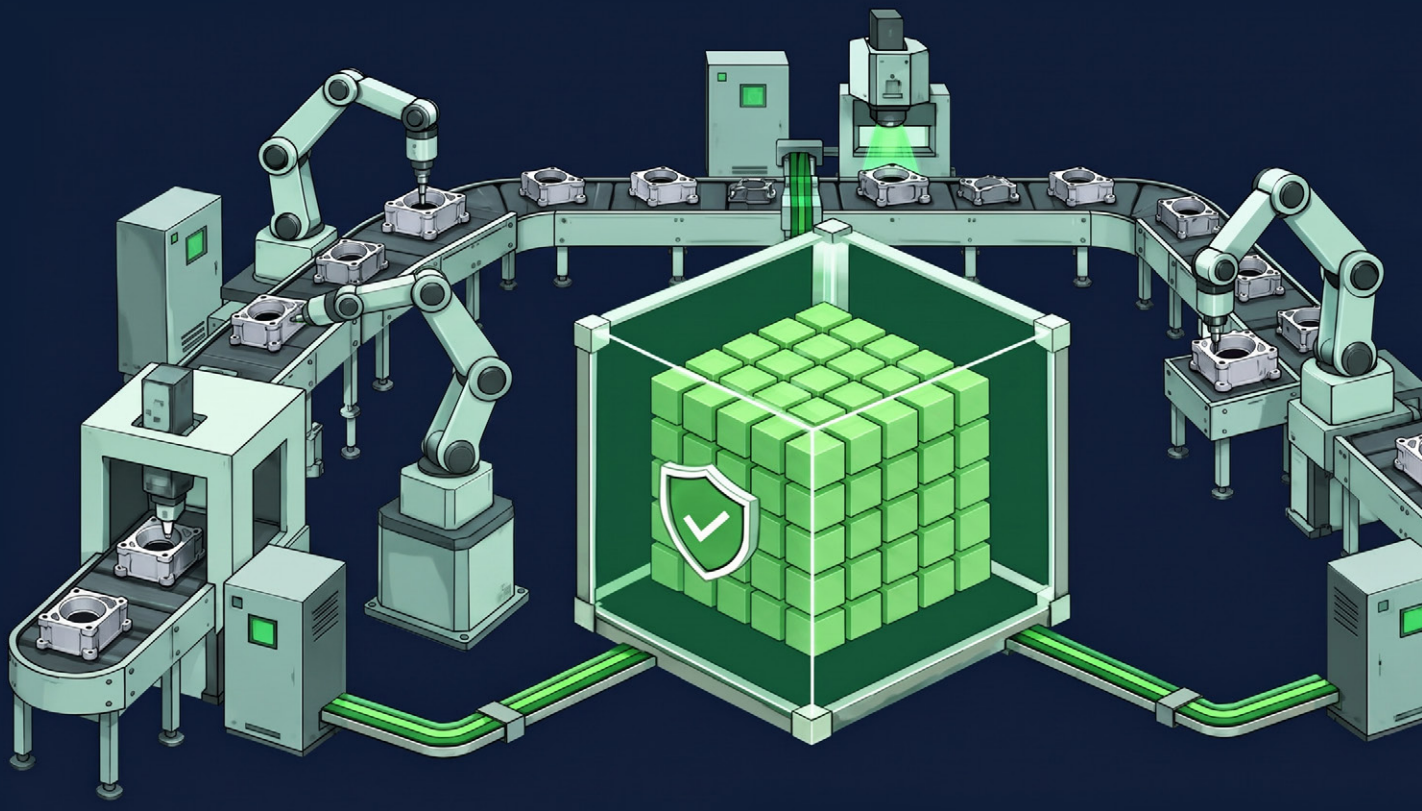
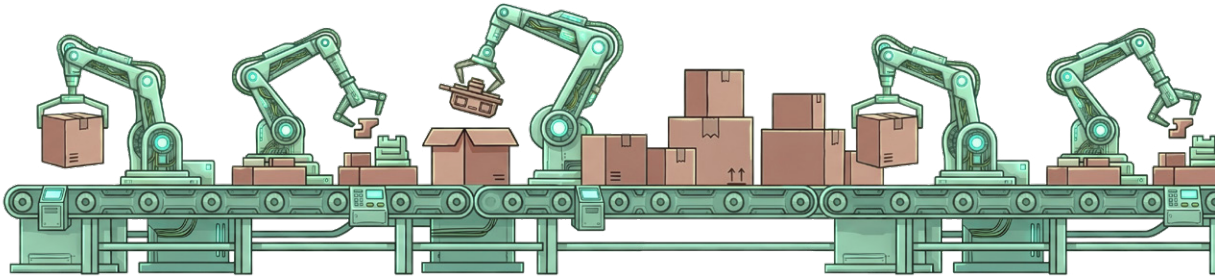


Table of **Contents**

Overview



Modern manufacturing has a software problem. While ransomware hitting headlines is the visible surface, there is a more pervasive fragility underneath: toolchains built from unvetted components, firmware shipping with known vulnerabilities, and engineering teams operating across acquired organizations with no shared security baseline. This guide examines the real challenges manufacturing CISOs, DevOps Directors, and AppSec leaders face, and how leading organizations are solving them.

#1

Most-Attacked Industry

Manufacturing: 27.7% of all cyberattacks in 2025[1]

$260K

Per Hour

Average cost of a cyberattack-induced production halt[2]

61%

YoY Ransomware Surge

Manufacturing incidents surged from 520 to 838 in 2025[3]

30%

Of All Breaches

Trace back to third-party software providers[4]

1. Euny Hong, "Why manufacturing companies are most vulnerable to hacking", *IBM*
2. Josh Connor, " How Attackers Are Exploiting Manufacturing Cybersecurity Weaknesses in 2026, *1KOSMOS*
3. "Escalating ransomeware threats to national infrastructure", *KELA*
4. Carlos Arcila, "Verizon's 2025 Data Breach Investigations Report: Alarming surge in cyberattacks through third-parties", *Verison*

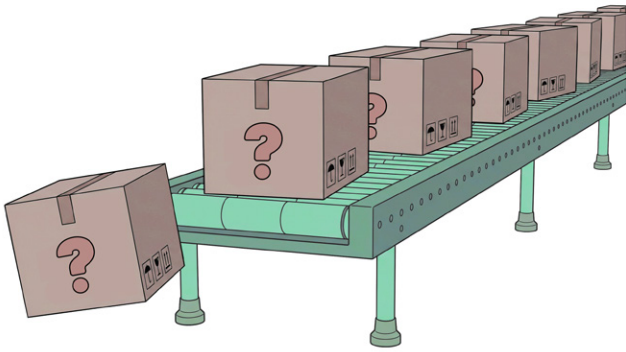
Manufacturing Is Now a Software Business, **Whether It's Ready or Not**

For decades, manufacturing security meant physical perimeter controls, proprietary OT systems, and air-gapped networks that rarely changed. That world is gone. The convergence of IT and operational technology has fundamentally redrawn the attack surface. PLCs, SCADA systems, and industrial controllers that were once isolated now run software that is built, updated, and distributed using the same pipelines as any enterprise application, and they inherit all of the same vulnerabilities.

Today's production floor runs on firmware in line controllers, containerized microservices managing factory automation, embedded software in connected IoT sensors, AI/ML models governing predictive maintenance and quality control, and supplier-provided components built directly into finished goods. Each is a potential point of failure. The question is no longer whether a manufacturing organization has a software supply chain, it's whether they have visibility and control over it.

CHALLENGE 01

No One Knows Exactly **What's Shipping**



Ask most manufacturing engineering teams to describe precisely what components are in the firmware that shipped to a production controller last quarter. In the majority of organizations, the answer is some version of, "We think we know." That uncertainty is a structural consequence of how most manufacturers have grown.

The dominant growth pattern in industrial manufacturing is acquisition. Each absorbed company arrives with its own artifact store, build conventions, and definition of an 'approved' component. Developers pull Docker images based on familiarity. Engineers reuse packages from previous projects without checking for newer, safer versions. No single team has cross-organization visibility. The result is build drift, production environments that diverge from staging in ways nobody fully tracks, and software bills of materials (SBOMs) that require days of manual effort to assemble from multiple sources, if they're assembled at all.

Why This Matters

The compliance cost

Without a continuous, automated SBOM process, manufacturing organizations are perpetually behind regulatory requirements. IEC 62443, EO 14028, and an increasing volume of customer contractual requirements all mandate documented software composition. Manual SBOM assembly at scale is not a sustainable answer.

The audit exposure

Build drift between development, staging, and production environments means that what security teams scan and what actually ships are frequently different things. In a regulated manufacturing environment, that gap is a material audit finding.

The acquisition multiplier

Each merger or acquisition resets the problem. Acquired teams onboard their own tools, their own repositories, and their own tribal knowledge about which packages are 'safe,' without any automated mechanism to integrate or enforce the acquiring organization's security standards.

How Leading Manufacturers Are Addressing This

Organizations solving this problem have replaced fragmented, siloed artifact storage with a single source of truth for every binary: container images, firmware binaries, embedded C/C++ packages (Conan), Python ML libraries, Java automation components, and more. With immutable, checksummed storage and geo-distributed replication, builds become reproducible, production environments become definitively defined, and SBOM generation becomes a continuous automatic output rather than a periodic manual exercise.

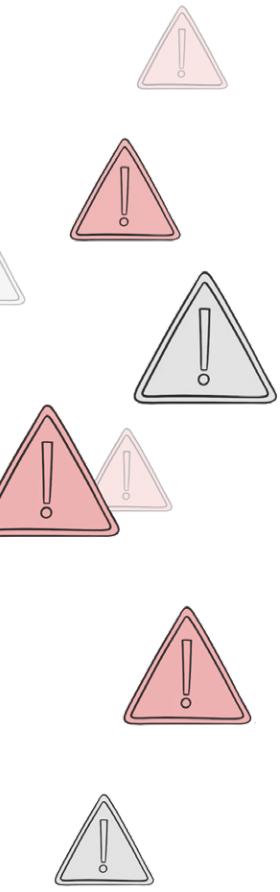
JFrog Artifactory provides universal binary management across 60+ package types, including Conan for embedded C/C++, the most underscanned ecosystem in manufacturing.

Immutable artifact storage with full build provenance answers regulatory audits with certainty: this is what shipped, to this environment, at this time.

Automated SBOM generation (CycloneDX and SPDX) as a continuous byproduct of the build process, always current, always tied to a known vulnerability profile.

CHALLENGE 02

Security Teams Are Drowning in CVE Noise **They Can't Act On**



Manufacturing security teams are not under-scanning. They're over-alarmed. A single embedded firmware build may contain hundreds of open-source dependencies, each with its own vulnerability profile. Automated scanners surface thousands of findings. The problem is not the volume of detection, it's the absence of context that makes the volume unmanageable.

When every finding looks equally urgent, none of them are. Security teams spend weeks chasing CVEs that are not exploitable in their specific environment, or they begin quietly deprioritizing alerts because the volume makes triage impossible. In either case, the genuinely critical issues (i.e. the reachable, exploitable vulnerabilities in actively used production components) get buried in the noise.

This is particularly dangerous in manufacturing, where firmware and embedded software release cycles are long and release windows are narrow. A vulnerability that misses a sprint may not surface for remediation until the next major release cycle, by which point it has been deployed to thousands of connected devices on a factory floor.

Why This Matters

Reason it happens

Most scanning tools report on presence, not applicability. They flag a vulnerable function in a dependency, but not whether that function is called in the actual build, whether the attack vector assumed by the CVE is reachable in the runtime environment, or whether the severity rating applies to the specific deployment context.

The compliance pressure

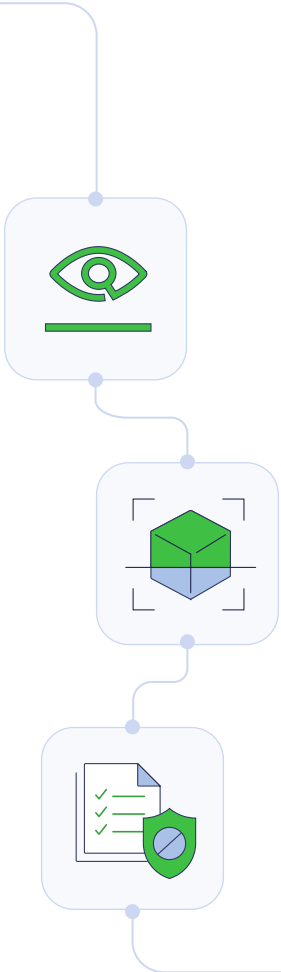
IEC 62443, NIST SP 800-82, and EO 14028 are simultaneously increasing the documentation burden on security teams. Teams are asked to produce more detailed SBOM outputs, respond to CVEs faster, and demonstrate traceable chain of custody, all while headcount stays flat and alert volume escalates.

The remediation math

Remediating a vulnerability at the package ingestion stage costs nearly nothing. The same vulnerability discovered after it has been built into firmware, tested, approved, and deployed to a production PLC costs orders of magnitude more, in developer time, operational disruption, and regulatory exposure.

How Leading Manufacturers Are Addressing This

The most effective approach treats scanning not as a reporting function but as a decision-support system. Contextual analysis determines whether a vulnerable function is actually called in the specific build, reducing the real remediation list from thousands to dozens. Continuous scanning ensures that vulnerabilities disclosed after a build is released are still caught automatically, without a new manual scan cycle.



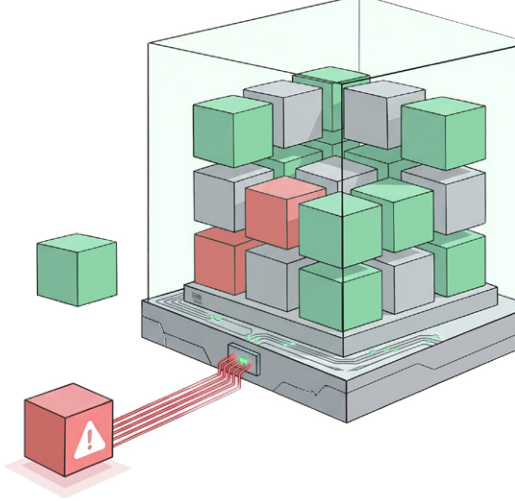
JFrog Advanced Security **Contextual Analysis** determines whether a CVE is actually reachable in the specific build, not just present in a dependency graph.

Continuous re-scanning automatically re-evaluates the full artifact landscape whenever a new CVE is published, with no manual trigger required.

Policy-driven blocking fails builds automatically when a component exceeds defined thresholds, enforcing decisions at the CI/CD layer, not in a spreadsheet.

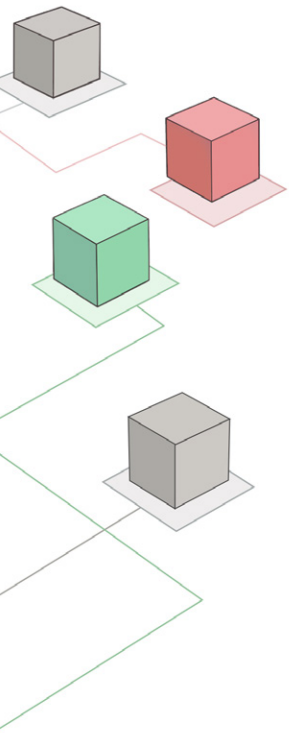
CHALLENGE 03

Open-Source Components Enter the Supply Chain **Without Any Vetting**



Manufacturing engineering teams consume open-source software at scale, across Maven, npm, PyPI, Docker Hub, Conan for embedded C/C++, NuGet for Windows automation, and more. The productivity gain is real. The risk is frequently underestimated and structurally unaddressed.

The threat is not limited to known CVEs in legitimate packages. It includes supply chain attacks that introduce malicious code into packages developers already trust: typosquatting attacks mimicking popular package names, dependency confusion attacks exploiting how package managers resolve internal versus public packages, and packages embedding malicious payloads in install scripts that execute before any scanner can inspect them. For manufacturers building firmware for industrial controllers, a compromised component doesn't mean a data breach, it means a backdoor with direct access to operational technology systems.



The Open-Source Governance Gap

Most manufacturing organizations have no formal process for approving open-source components before they enter the development environment. Developers pull whatever resolves their dependency, often the first result a package registry returns, with no automated check for known vulnerabilities, malicious content, license risk, or operational quality. By the time a security scan runs, the package is already in the codebase, already cached in a developer's environment, and potentially already in a build artifact heading to production.

How Leading Manufacturers Are Addressing This

Organizations addressing this risk have moved policy enforcement to the earliest possible point: before a component reaches any developer machine. Automated curation enforces approved component policies at the package ingestion layer, malicious packages, critically vulnerable components, and license-risky dependencies are blocked at the proxy before any engineer can consume them. The secure choice becomes the default choice.

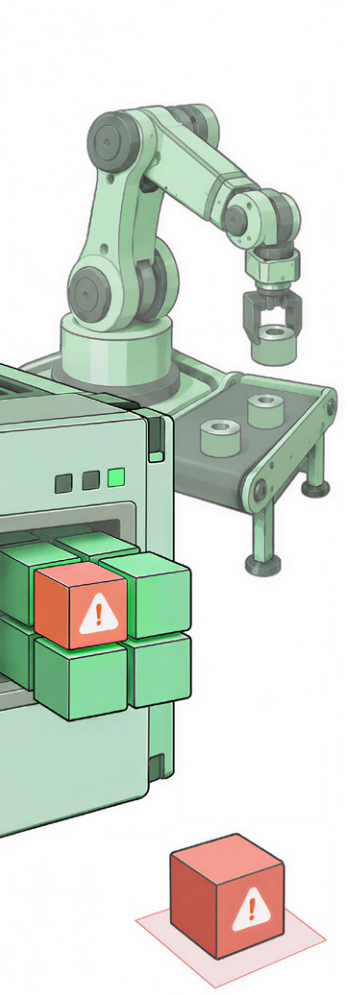
JFrog Curation operates as a policy enforcement gateway between public registries and internal artifact stores.

Malware detection, typosquat detection, and **license governance** are applied automatically at ingestion.

Security decisions are enforced by the platform, not by individual developer judgment under time pressure.

CHALLENGE 04

Getting Software to the Factory Floor **Without Creating a New Attack Vector**



Distributing software updates across a global manufacturing operation is a fundamentally different problem from pushing a web application release. Production line controllers run on constrained OT networks. Remote facilities may have intermittent connectivity. Air-gapped environments exist specifically to isolate critical systems. And the tolerance for a failed or corrupted update is close to zero; a bad firmware push to a PLC can halt a production line or, in a worst-case scenario, introduce a persistent backdoor into OT infrastructure.

At the same time, the pressure to update is intensifying. Connected devices have operational lifespans that far outlast the software stacks they launched with. A sensor deployed in 2019 may still be operating in 2031, accumulating unpatched CVEs across multiple release cycles. Manufacturers are caught between the operational risk of pushing updates and the security risk of not pushing them.

Why This Matters

Integrity in transit

Software distributed without cryptographic signing can be tampered with between the build system and the target environment. For manufacturers pushing firmware to remote facilities or contract manufacturers, there is no reliable way to verify that what arrived is what was sent, without a signed, immutable distribution mechanism that provides a chain of custody from build to deployment.

Multi-site version drift

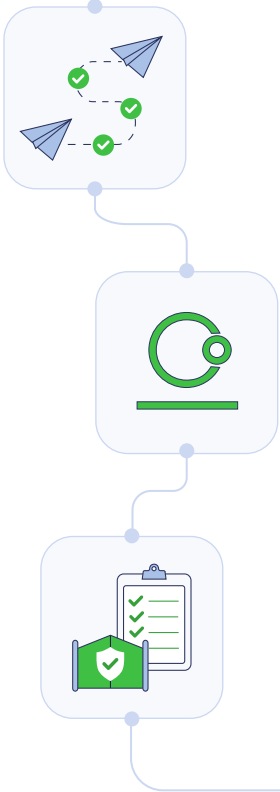
Global engineering teams working across time zones create coordination problems that compound distribution risk. When regional teams maintain local artifact caches without a synchronized source of truth, version drift is inevitable. Teams build against different dependency versions with no automated mechanism to detect or correct the divergence.

OT network constraints

Operational technology networks are frequently separated from corporate IT networks by design, exactly as manufacturing security frameworks recommend, and that same isolation often means limited or intermittent connectivity back to central infrastructure. Distribution systems that assume reliable, high-bandwidth connections fail in precisely these environments, the ones most in need of secure, dependable updates.

How Leading Manufacturers Are Addressing This

Organizations that have solved this problem have implemented cryptographically signed release bundles as the unit of software promotion, proving that a specific, validated set of artifacts has not been modified between the build system and the target environment. Edge distribution nodes deployed at remote facilities provide local artifact caches that operate independently of central infrastructure availability.



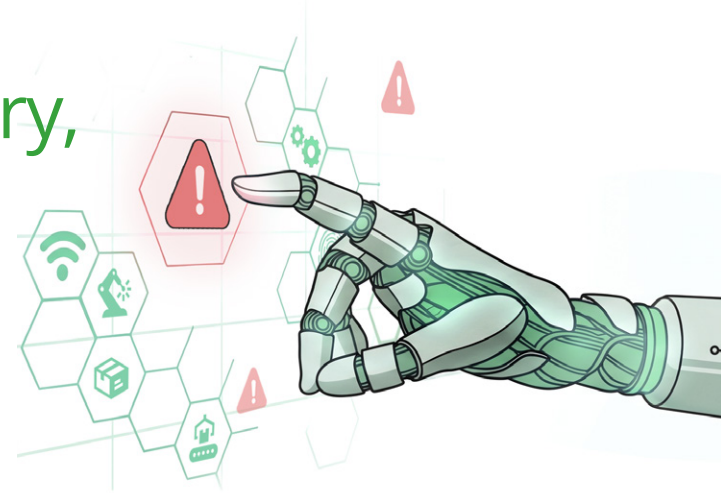
JFrog Release Bundles provide cryptographic proof of artifact integrity from build to deployment, including air-gapped and constrained OT environment.

JFrog Edge nodes deployed at remote sites maintain local caches that continue operating through WAN outages.

Automated promotion gates enforce security policy at every environment boundary; nothing reaches production without passing all configured checks.

CHALLENGE 05

AI Is Already in the Factory, **but Security Hasn't Caught Up**



Manufacturing was among the earliest enterprise adopters of AI/ML: predictive maintenance, computer vision for quality control, autonomous material handling, demand forecasting. The business case was clear. However, the security implications were not, and most organizations did not build governance frameworks alongside their AI deployments.

AI models are software artifacts with the same supply chain risks as any other open-source component. In some ways, they are more severe: a model downloaded from a public repository may contain malicious serialized payloads, depend on Python packages with critical CVEs, or have been trained on poisoned data designed to cause specific, targeted mispredictions in a production quality control system. Unlike a vulnerable npm package, a compromised AI model can be nearly impossible to audit without purpose-built tooling.

Most manufacturing security teams that have mature software supply chain controls have no equivalent governance process for AI/ML artifacts. Models move from data science environments into production systems through informal channels: no SBOM, no vulnerability scan, no approval workflow. This is the open-source governance gap from a decade ago, repeating itself in a new domain, with a shorter runway before both regulators and adversaries catch up.

What the Emerging AI Attack Surface Looks Like in Manufacturing

A quality control computer vision model running on a production line inspects components at speeds that human operators cannot match. If that model was pulled from a public repository without security vetting, it is a privileged, trusted process with direct access to manufacturing data and potentially to OT-adjacent network segments, with no SBOM, no known CVE profile, and no approval record. The JFrog Security Research team has identified hundreds of malicious models on public repositories including Hugging Face.

How Leading Manufacturers Are Addressing This

The JFrog Platform extends the same security scanning, governance, and traceability that organizations already apply to packages and containers directly to AI/ML models, giving security teams visibility into every model in their environment before it reaches production. The governance framework translates directly from software to AI.

JFrog AI Catalog provides discovery, governance, and security scanning for AI/ML models, the same controls applied to software artifacts.

Proxy and scan capabilities route public model repositories through governed internal infrastructure, scanning every model and its dependency chain before reaching production.

AI-Bill of Materials (AI-BOM) generation documents model provenance and security posture for every model promoted to production.

CUSTOMER PROOF

How Manufacturing Organizations Have **Closed the Gap**

The challenges described in this guide are not theoretical. The following organizations faced them, made a strategic decision to address them systematically, and documented measurable results. The common thread is a shift from reactive, siloed security tooling to a unified, policy-driven approach to software supply chain governance.

VIAVI Solutions

VIAVI Solutions — Global Manufacturing

The problem

Fragmented artifact stores across 20+ acquired teams. No SBOM process. Developers pulling Docker images on familiarity, not security. Vulnerability alerts with no prioritization context.

What changed

Single source of truth for all artifacts. Contextual CVE analysis with only real risks surfaced. Continuous, automated SBOM generation. Standardized onboarding for acquired teams.

Outcomes

1,000+
developer hours recovered

20%
output improvement

$1.2M
cost savings over 3 years

> "Because we're now able to operate more efficiently, our developers have more time to focus on meaningful work. We're able to standardize across processes, ensure security and, as a result, have much more confidence across the software supply chain."
>
> — Bill Goodrich, Principal DevSecOps Engineer, VIAVI Solutions

Siemens — Industrial Automation & PLCs

The problem

Tool sprawl and binary bottlenecks across 50,000+ developers building industrial PLC and process automation software. Fragmented security enforcement across a massive global engineering organization.

What changed

Unified DevSecOps across the entire engineering organization. Security policy enforced at scale via GitHub integration. Binary bottlenecks eliminated.

Outcomes

50,000+
developers on one platform

Faster
industrial software delivery

Consistent
security posture

Panasonic HVAC — HVAC Manufacturing Division

The problem

Manually tracking vulnerabilities in spreadsheets across multiple languages and environments. No unified artifact management. Reactive, not preventive, security posture.

What changed

Artifact management unified across all languages and environments. Shift-left security scanning embedded in CI/CD. Spreadsheet-based vulnerability tracking replaced.

Outcomes

99%
SLA uptime achieved

Verifiable
chain of custody

Proactive
security posture

Global Manufacturer — 15,000 Engineers, 5 Global Sites

The problem

Binary bottlenecks and zero visibility into third-party package vulnerabilities slowing a distributed 15,000-person engineering operation across 5 locations.

What changed

Artifact storage consolidated across all 5 sites. Full vulnerability visibility across 109 repositories. DevSecOps practice standardized globally.

Outcomes

2M+
artifacts secured

109
repos under one control plane

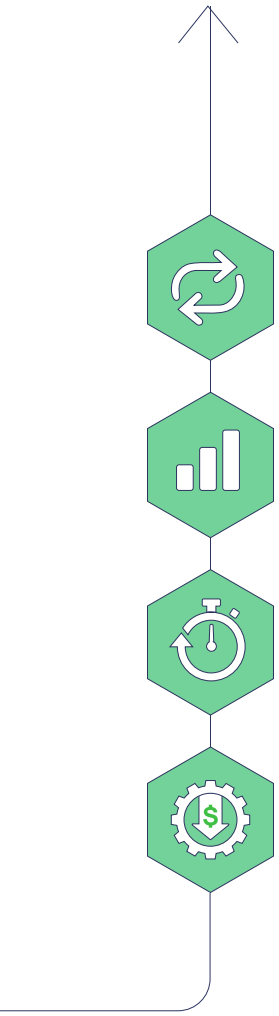
Eliminated
global bottlenecks

Analyst recognition

JFrog was positioned as a Leader in the inaugural **2026 Gartner® Magic Quadrant™ for Software Supply Chain Security** placing highest for Ability to Execute.

BUSINESS IMPACT

The Cost of the **Status Quo vs. the Cost of Addressing It**



The conversation about software supply chain security in manufacturing is often framed as a cost. The Forrester Total Economic Impact™ study of the JFrog Platform reframes it: the real cost is in maintaining the status quo.

282%
ROI
Forrester Consulting Total Economic Impact™ study of JFrog (2026)

$5.4M
3-Year Benefit
Risk-adjusted present value across enterprise deployments

80%
Faster Remediation
Reduction in mean time to remediate CVEs

71%
Lower Tool Spend
Achieved by consolidating point solutions onto one platform

The figures above reflect a composite enterprise organization.

Compliance Frameworks Manufacturers Need to Address

IEC 62443

Industrial cybersecurity standard for OT environments. Requires documented software composition, access controls, and security management across the OT lifecycle.

NIST SP 800-82

Guide to OT security. Mandates secure software development lifecycle practices, build traceability, and policy-driven environment controls.

EO 14028 / SBOM

Requires SBOM generation for software sold to the US federal government. Manufacturers in the defense and critical infrastructure supply chain are directly affected.

FDA Cybersecurity

Connected medical device manufacturers must provide a cybersecurity bill of materials and demonstrate a secure product development framework for 510(k) and PMA submissions.

See How **JFrog Secures** the Manufacturing Software Supply Chain

If you are a DevOps, security, or compliance professional in manufacturing, take a self-guided tour, schedule a demo, or start a free trial to see how JFrog can help you manage, secure, and document your software supply chain.

JFrog empowers thousands of DevSecOps organizations globally to build, secure, distribute, and connect any software artifact to any environment using the universal, hybrid, multi-cloud JFrog Platform.