

WHITE PAPER

Path to Zero Vulnerabilities

Bad actors are infiltrating already vulnerable open-source software with malicious intent and weaponizing AI to exploit vulnerabilities.

Open-source software (OSS), including OSS LLMs, MCP servers, and AI agent code, has become a foundational component of modern software and AI application development, enabling rapid innovation by allowing developers to build on existing, community-driven code. With 70–90% of today's applications composed of OSS components, organizations depend heavily on this collaborative ecosystem to keep pace in a competitive technology landscape. The good intentions, culture and collaboration of the open-source community provides a strong backbone for innovation. But while efficient, open-source software introduces exponential risks, and the innovation-driven open-source community can only do so much to keep existing software secure.

The open and transparent nature of OSS, while core to its success, inherently introduces security concerns. Because the source code is publicly accessible, cyber threat actors have the opportunity not only to study it for vulnerabilities but also to manipulate or compromise it for malicious purposes. This openness can provide attackers with pathways into organizational networks and systems, especially when vulnerable components are deeply embedded in critical applications.

95%

of risk is sourced

**>85%**

of vulnerabilities can be exploited by LLMs



4 Open-Source Software Risks to Consider

1 Dependencies introduce challenging complexity

Software includes extensive webs of open-source dependencies, many of which pull in additional libraries behind the scenes. Direct dependencies drag in transient dependencies up to 60 levels deep that even open-source developers don't have visibility into. This creates long, interconnected chains that organizations often struggle to map or monitor. When even a small or obscure dependency contains a weakness, the issue can ripple upward into countless applications. A single flaw in the supply chain can therefore have outsized consequences, especially when it is deeply embedded and widely reused.

3 There is no dedicated support or standard security workflows

Most of the open-source ecosystems rely on volunteer contributors to identify, report, and patch security issues rather than having dedicated support teams. Also, due to the nature of rapid iteration or community-driven feature development, comprehensive security practices are not always embedded in the workflow. Formal elements like threat modeling, secure coding guidelines, or automated security scanning may be absent or inconsistently applied. Without these guardrails, vulnerabilities can enter the code early and persist unnoticed, ultimately propagating into the products and services that rely on them. More than 70% of critical and high vulnerabilities are never fixed but they continue to live in organization's code.

4 Developers can't fix what they did not build

The sole reliance on a community effort can also lead to delays in updates and inconsistent availability of security fixes. This leaves organizations scrambling to find fixes for code they did not create. In the absence of timely remediation, known vulnerabilities remain exposed, giving threat actors opportunities to infiltrate systems, escalate attacks, disrupt operations and compromise sensitive information.

2 Code quality and trustworthiness varies

Not all OSS undergoes rigorous testing or expert security review, and there is no universal standard ensuring that contributors thoroughly vet changes before they are released. 8.65% of open-source software is of dubious origin, which means its provenance is provably incorrect and unattested. Additionally, many open-source libraries begin with active development but gradually lose contributors or fall out of maintenance. Organizations may continue to rely on these aging components without realizing that updates have slowed or stopped entirely. As a result, organizations may unknowingly adopt components with undetected flaws and vulnerabilities.

Backlogs Are Relentless

Code Velocity > Vulnerability Prioritization Velocity

Even daily reprioritization cannot keep up with hourly code changes. Reachability changes with every code changes and detection tools cannot keep up.

All Vulnerabilities Are Reachable

If an attacker can load and execute vulnerable code, it does not matter that a developer did not call that function. It's reachable!

Unfixed Vulnerabilities are Exploitable

Three quarters of open-source critical and high vulnerabilities are unfixed. A developer can't fix them, VM tools ignore them and DevSecOp teams give them a pass.



Vibe Coding is Imperfect

Developers increasingly use tools like Cursor, Claude Code and CoPilot to rapidly generate code, which expands risk and drags in open-source dependencies without developer being aware.

Traditional Appsec Tools Can't Keep Up

Modern application security tools consistently fall short because they weren't built for today's rapidly evolving, highly connected and AI-fueled development practices. Most tools generate noise instead of clarity, bog teams down with alerts that lack meaningful context, and often interrupt developer workflows. As applications grow more complex and AI accelerates code creation, these limitations leave organizations with an increasing pile of unresolved security issues and an illusion of safety.

Why These Tools Miss the Mark

Built for a slower era

Most traditional security platforms were designed around older, monolithic release cycles. They can't adapt to the constant change and automation of cloud-native development or handle the distributed nature of modern systems.

Too much noise, not enough insight

Security teams are often buried under mountains of alerts that lack business or environmental context. Critical issues that directly affect exposed services can be buried beneath false positives or minor findings.

Fragmented visibility

Legacy solutions inspect isolated pieces of the development lifecycle but fail to provide an end-to-end view from source code to running workloads. Without this holistic perspective, organizations miss how vulnerabilities actually manifest or interact in real environments.

Increases backlog accumulation

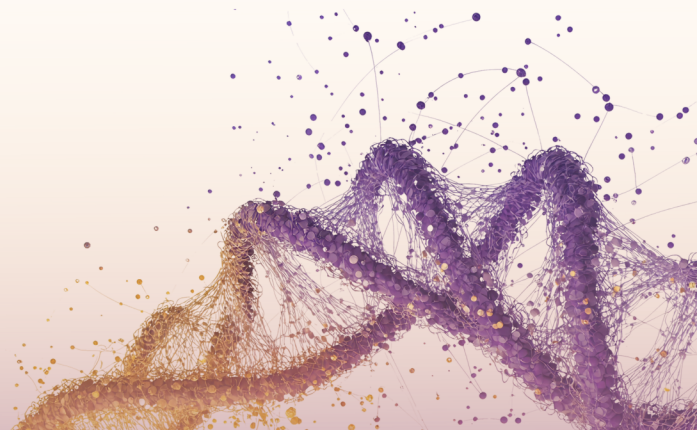
Slow scanning, high rates of false positives, and poor workflow integration leave many issues unaddressed. Over time, this compounds into significant security debt that teams struggle to manage.

Unprepared for the AI era

As AI-generated code and machine-learning components become more common, new categories of risk such as prompt-based attacks or model manipulation have emerged. Traditional security tools can't match the velocity at which AI has accelerated software development and increased the related attack surface.

Breaks code

Most tools focus on fixing vulnerabilities in the latest versions of dependencies, and latest versions frequently break existing applications. On an average, a dependency is used by 2-3 developers in their code. Coordinating their changes simultaneously is hard, creating additional breakage when one developer updates and others do not.



Source Safe Open-Source Software and Auto-Remediate Vulnerabilities with Lineaje

Lineaje provides full-lifecycle software supply chain security for critical software, continuously, autonomously and at scale.

With firm roots in software supply chain security, Lineaje is poised to take on software security in the age of AI. Leveraging technology that is purpose-built to secure the evolving software development lifecycle, organizations can combine sourcing safe software with auto-remediation to get on a viable path to zero vulnerabilities.

95%

Reduction in critical and high vulnerabilities by sourcing safe software

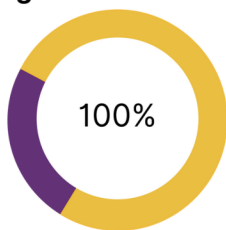
20%

Reduction in developer toil with auto remediation

100%

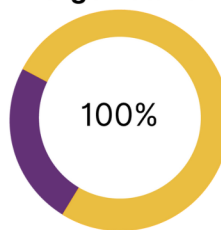
compatible software and fixes that do not break builds

Large Product Vendor



■ Safe open-source ■ Premium fixes

Large Retailer



■ Safe open-source ■ Premium fixes

Source Safe Software

Lineaje Gold Open Source (GOS) is an enterprise-grade subscription service that continuously provides attested, critical and high vulnerability free, malware-free, untampered versions for all packages developers use in applications and all images they depend on in containers. GOS maintains a curated, vetted, safe, and high-integrity repository of popular, secure open-source packages and images. Compatible packages and images can also be autonomously created, on-demand. Additionally, advanced policy controls ensure that corporate security guidelines are met before risks can enter an organization. Developers, platform engineering, and DevOps teams can use only trusted open-source packages and images, without disrupting the software development process.

Source safe open-source packages, before risks can enter the organization or vulnerability exploits escalate.

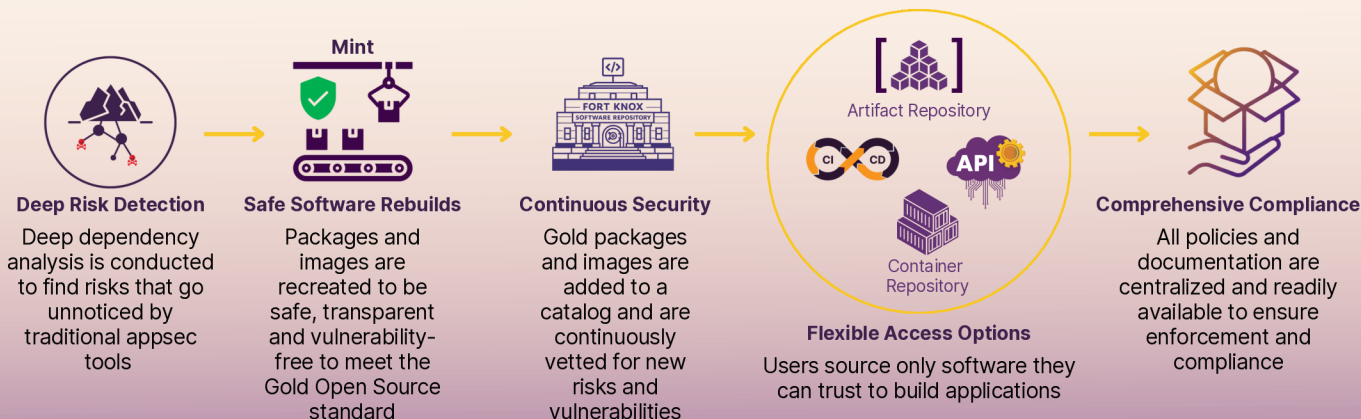
3 MILLION +

Vulnerability-free packages available


Automate container hardening and guarantee application compatibility to avoid breaking builds and interfering with productivity.

3 THOUSAND +

Hardened container images available



Self-Heal Source Code and Containers

 **Lineaje's SBOM360** leverages agentic AI to autonomously and continuously secure software development factories across the software development lifecycle. Get comprehensive visibility into what's in your applications through deep dependency analysis and intelligent risk scoring. Understand sources of risk to eliminate sourced risk, and autonomously generate intelligent fix plans for vulnerabilities in source code, and containers. Auto-remediate vulnerabilities in minutes, without breaking builds, and prove continuous compliance to achieve a self-healing software supply chain.



Know What's in Your Software

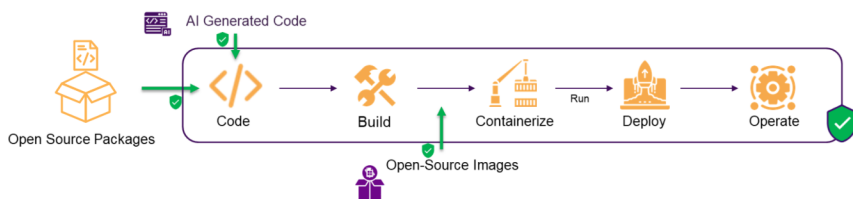
Unify all software scanning into a central view and get continuous analysis of applications, source code repositories, container images, and SBOMs to gain complete visibility into all dependencies.

- ✓ Vulnerabilities
- ✓ Licenses
- ✓ SBOMs
- ✓ Provenance
- ✓ Geo-provenance
- ✓ Tamperers
- ✓ Threat intelligence
- ✓ And more, in multiple form factors



Auto-Fix Source Code & Containers

Agentic AI analyzes dependency chains and auto-generates intelligent fix plans for packages and containers in minutes. Choose from major, minor, custom, or compatible upgrade paths.



Continuously Comply

Enforce policies and generate complete SBOM, VEX, and CSAF documentation for your existing software; demonstrate compliance with comprehensive attestation for every component.



[Learn more](#)



[See it in Action](#)



[Try It](#)

About Lineaje

Lineaje provides full-lifecycle software supply chain security for critical software, continuously, autonomously and at scale. It's portfolio of products allows users to source safe software, contextualize risks, auto-secure builds, and manage risk and compliance for all software that you source, build, buy and sell. The Lineaje portfolio provides unmatched discovery, deep assessment, compatible fixes and comprehensive compliance to eliminate risks and reducing developer maintenance effort by 40%.



Lineaje AI Agents

Continuous, Autonomous and at Scale



BOMbots

Deep Discovery and Assessment



FIXbots

Trusted Auto-Remediation



COMPLYbots

Comprehensive Policy Enforcement & Compliance

[Book a Demo](#)