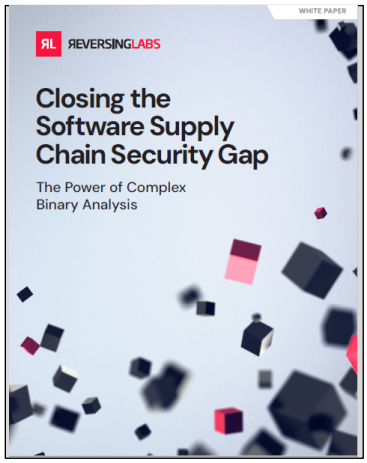


Closing the Software Supply Chain Security Gap

Thank you for downloading this ReversingLabs resource. Carahsoft is the distributor for ReversingLabs Cybersecurity solutions available via TX DIR and other contract vehicles.

To learn how to take the next step toward acquiring ReversingLabs solutions, please check out the following resources and information:

For additional resources:
carah.io/ReversingLabsResources

For upcoming events:
carah.io/ReversingLabsEvents

For additional ReversingLabs solutions:
carah.io/ReversingLabsSolutions

For additional Cybersecurity solutions:
carah.io/Cybersecurity

To set up a meeting:
ReversingLabs@carahsoft.com
844-445-5688

To purchase, check out the contract vehicles available for procurement:
carah.io/ReversingLabsContracts

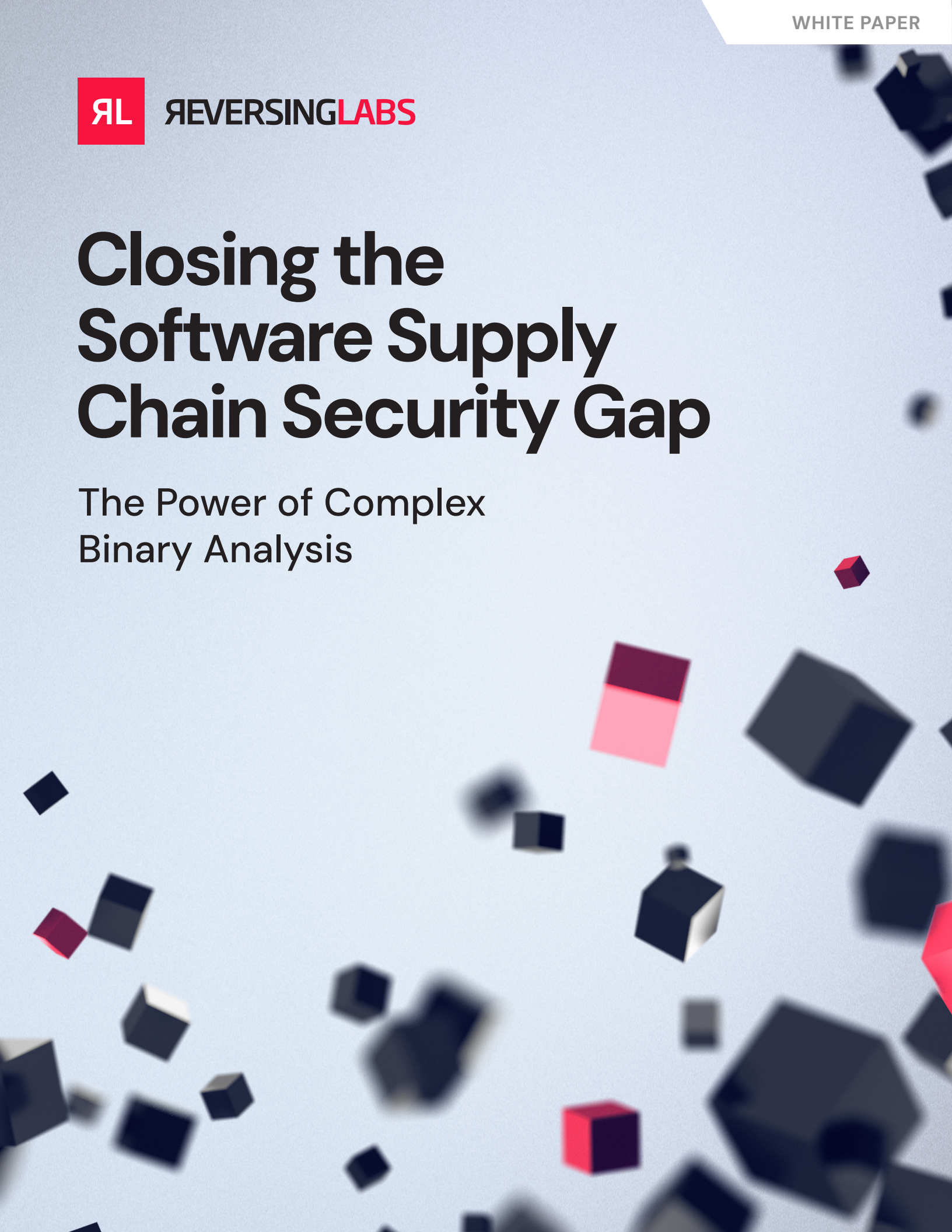
For more information, contact Carahsoft or our reseller partners:
ReversingLabs@carahsoft.com | 844-445-5689

WHITE PAPER

REVERSINGLABS

Closing the Software Supply Chain Security Gap

The Power of Complex Binary Analysis



The Hole in the Software Supply Chain

As the application security testing landscape evolves, it confronts a wave of sophisticated attacks that infiltrate software supply chains that capitalize on advanced tampering techniques. In this complex and volatile environment, the inherent limitations of traditional application security (AppSec) tools are becoming starkly apparent. Rooted primarily in technologies like software composition analysis (SCA), static application security testing (SAST), dynamic application security testing (DAST), and interactive application security testing (IAST), these conventional tools are increasingly inadequate in addressing modern software supply chain attack vectors. While excelling at detecting security defects, like those outlined in the OWASP Top 10, they fail to detect and mitigate the advanced, deeply embedded threats and risks that effortlessly circumvent legacy detection mechanisms. Each of these tools are segmented and isolated in their respective areas.

When it comes to exercising software supply chain security (SSCS) best practices, many organizations make the mistake of focusing solely on open-source vulnerability testing. The reality that confronts enterprises today is that modern software packages are complex assemblies of proprietary, commercial, and, yes, open-source components, each introducing its own set of issues. This holds true for not only the software your organization builds and ships to customers, but the software you buy and deploy across your entire organization as well. This multifaceted nature of software demands a level of analysis far beyond what traditional tools offer.

Traditional application security testing (AST) tools, once the mainstay of security, are now proving less effective in addressing modern software supply chain attacks. Their siloed nature focuses on vulnerabilities within a specific subset of the codebase, be it open-source libraries, source code, or running web applications. This approach fails to address attacks where malicious actors embed malware and other threats within software components or artifacts that are added to software packages during the final build. Furthermore, they lack a robust foundation of threat intelligence, a crucial element for identifying and addressing potential risks.

Without a deep, historically informed understanding of what threats look like, how they behave, and how they evolve, traditional AST tools fall short offering only limited visibility into software risks (mostly vulnerabilities), while missing the increasingly sophisticated and hidden software supply chain threats. In short, they cannot see or stop software supply chain attacks.

It is clear that enterprises are facing a new wave of attacks that infiltrate software supply chains with advanced obfuscation techniques. As we navigate the modern software attack surface, a new approach is essential – one that moves beyond the capabilities of traditional AST tools to embrace a wider scope of software supply chain risks such as software tampering, embedded malware, and exposed secrets.

Understanding Legacy AppSec Limitations

In the rapidly evolving realm of application security, traditional tools like SAST, DAST, IAST, and SCA have been cornerstones of traditional software security strategies. However, as cyber threats evolve into more sophisticated and stealthy entities, the limitations of these long-established methods are increasingly evident. These tools, primarily designed to flag vulnerabilities through signature-based detection or predictable behavioral patterns in small slices of code, struggle to keep pace with the advancing tactics of modern cyber adversaries.

“Software supply chain attacks have seen triple-digit increases, but few organizations have taken steps to evaluate the risks of these complex attacks.”

READ THE REPORT

Gartner®, Mitigate Enterprise Software Supply Chain Security Risks
Dale Gardner, 31 October 2023

The inherent inefficiencies of these legacy technologies become apparent when confronted with the complexities and dynamic nature of the modern software attack. For example:

- **Static Application Security Testing (SAST)** is invaluable for identifying vulnerabilities in code, often falls short when it comes to understanding the business logic, which encompasses the custom rules and processes that dictate how the application operates and data flows, leading to a high number of false positives and overlooked vulnerabilities
- **Dynamic Application Security Testing (DAST)** tests the exposed interfaces of running web applications but might miss deeper, more intricate security issues due to its surface-level analysis
- **Software Composition Analysis (SCA)** is crucial for identifying open-source components with known vulnerabilities, but doesn’t test proprietary (internal) or commercial (third-party) code. SCA also lacks the depth of analysis required to flag malicious packages (malware) or tampering, leaving gaps in detection

	SCA	SAST	DAST	SSCS
Malware & Threat Intelligence				●
Tampering Detection				●
Complex Binary Analysis				●
Extensive Binary Format Analysis				●
Extensive Artifact Coverage & Analysis				●
Software Bill of Materials (the list)	●			●
SBOM w/Assessment of 6 Risk Categories				●
Pre-Production Scanning	●	●		●
Production Scanning			●	◗
Third-Party Software Risk Management				●
Version Differencing				●
Secrets Detection & Prioritization	◗	◗	◗	●
CVE Detection	●	●	●	◗
Policy Fnforcement	●	●	●	●

Figure 1: Legacy AST tooling fail to address the full spectrum of software supply chain attack vectors

Lack of Visibility into Build Artifacts

With many AppSec organizations adopting a “shift left” approach to securing their applications, there is a heavy emphasis on integrating AST tools pre-compilation to prevent code vulnerabilities from making it to the final build. However, this creates an enormous visibility gap as build artifacts can increase the size of the total software package from 10 to 1,000 times or more. This creates an opening for attackers to inject malicious code into the application, as it would get lost in the deluge of artifacts added much later in the build process.

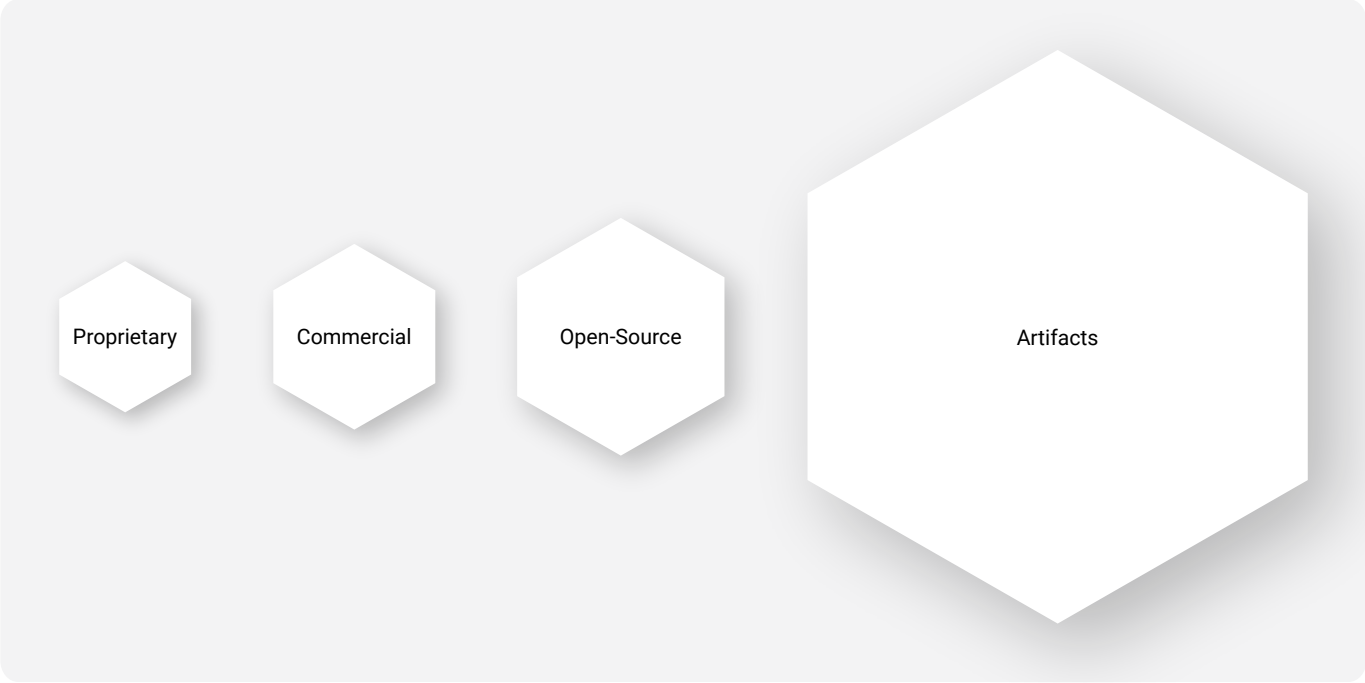


Figure 2: The components that make up a full software binary once compiled.

As outlined in Figure 1 above, the modern software supply chain security solution must consider all the parts that make up the final software including proprietary, commercial, and open source code, along with artifacts added during compilation.

Lack of Visibility into Commercial Software Packages

For Enterprise Software Buyers, specifically those in charge of Third-Party Risk Management (TPRM), visibility into commercial-off-the-shelf (COTS) software deployed across your IT environment is just as critical as securing the software your team builds in-house. We’ve already seen real-world examples of malicious actors exploiting the good reputation of commercial applications to exfiltrate sensitive data as was the case with SolarWinds, 3CX, CyberLink, CircleCI, and CodeCov to name a few. The complex mechanisms of these attacks were undetectable by traditional AST tools and approaches.

Because these tools require source code for analysis, cyber risk professionals do not have the means to assess for embedded threats in purchased software and, thus, are beholden to cumbersome, manual techniques such as security surveys and pen-testing. Without deeper and contextual insight into the inner workings of COTS applications in the form of a comprehensive and actionable software bill of materials (SBOM), cyber risk professionals lack the assurance that every proprietary, commercial, and open-source component within their vendor’s software is securely maintained and sourced by trusted sources.

Malicious Actors Get the Memo

These shortcomings haven't gone unnoticed. Indeed, sophisticated attacks and active exploits of software development organizations make it clear that Advanced Persistent Threat (APT) groups, and other threat actors, have gotten the memo on the limitations of existing application security measures. These threats, often deeply embedded within complex file structures or software components, can easily bypass detection mechanisms that rely on known signatures or observable malicious behaviors.

The SolarWinds software supply chain attack case is a prime example, showcasing how threat actors can exploit even the most fortified systems by circumventing traditional detection methods through advanced techniques like time-based payload execution or other delay tactics. In that incident, attackers believed to be Cozy Bear/APT29, a nation-state backed threat actor group associated with Russia's SVR, compromised SolarWinds' software build and code signing infrastructure, directly modifying proprietary source code to include a malicious backdoor that was then delivered to thousands of SolarWinds customers in the form of a signed Orion software patch through the company's software release management system.

SolarWinds isn't the only example, of course. CodeCov suffered its own software supply chain attack when a Docker image exposing sensitive credentials allowed attackers to clandestinely compromise CodeCov's Bash Uploader script, exposing the CI environments of a portion of their 29,000 customers.[1]

In 2023, state-sponsored North Korean threat actors breached Taiwanese multimedia software company CyberLink by hijacking and modifying a legitimate installer to push malware that was detected on over 100 devices worldwide.[2]

The lesson from these examples is clear: focusing on vulnerabilities is not enough. As the application security threat landscape evolves, it is increasingly evident that traditional tools and methods are inadequate to manage the multitude of threat categories affecting both software producers and buyers like malware, tampering, and exposed secrets. As a result, an expanded and more robust, comprehensive, and forward-thinking approach to ensuring the integrity of software supply chains and software artifacts is needed.

The Necessity of Complex Binary Analysis

The fragmentation of traditional application security testing methods underscore the need for a more refined approach to assessing the security and integrity of applications. Complex binary analysis is key to such efforts and is a linchpin technology for understanding and addressing modern software supply chain attacks such as SolarWinds and 3CX.

Complex Binary Analysis: Digging Deep into Application Risk

Complex binary analysis delves into a software package's binaries without requiring execution or source code. It enables a deep inspection of the software's structure, code, and dependencies in a post-compilation, pre-deployment state. This level of scrutiny provides a significant edge in identifying potential threats, risks, and issues, even identifying threat indicators that serve as a warning sign for potentially novel malware threats. Beyond running siloed tests of the application in a controlled environment, as is common with legacy AST tools like SAST, DAST, IAST, and SCA, complex binary analysis offers a proactive, comprehensive approach to security.

---

[1] Bleeping Computer, Hundreds of networks reportedly hacked in Codecov supply-chain attack, April 2021
https://www.bleepingcomputer.com/news/security/hundreds-of-networks-reportedly-hacked-in-codecov-supply-chain-attack/

[2] Bleeping Computer, Microsoft: Lazarus hackers breach CyberLink in supply chain attack, November 2023
https://www.bleepingcomputer.com/news/security/microsoft-lazarus-hackers-breach-cyberlink-in-supply-chain-attack/

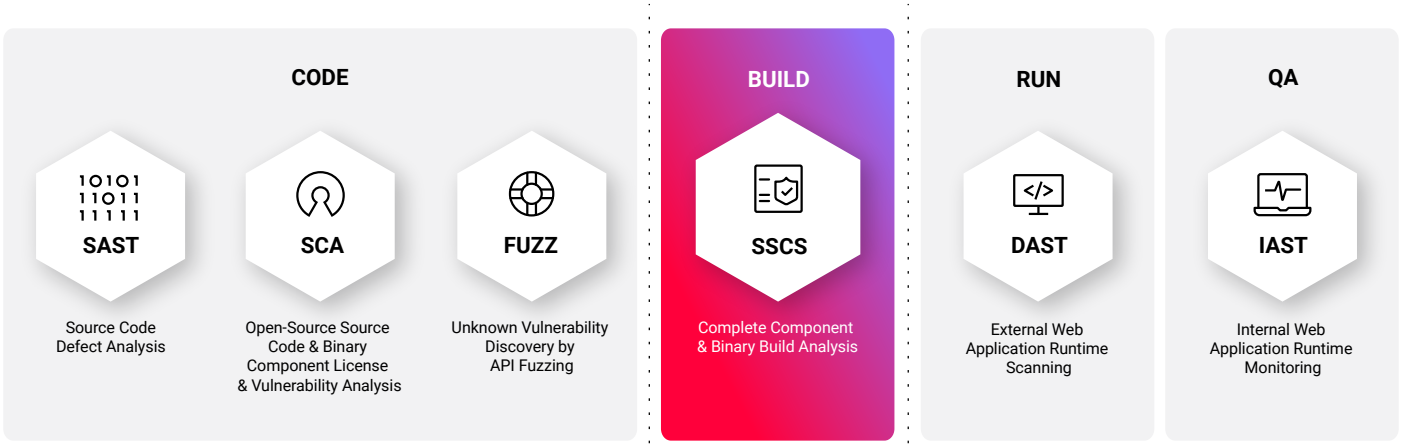


Figure 3: Complex binary analysis can flag and contextualize risk within the entire software package by plugging into the SDLC at the post-compilation, pre-deployment state.

The advantages of complex binary analysis are particularly evident when considering the real-time insights it provides. It enables teams to understand how the software will interact with its environment without having to execute the program itself. This granular observation uncovers hidden threats like tampering, exposed secrets, vulnerabilities, abused signatures, licensing issues, and malicious code that traditional application security testing tools miss.

Why do we crash test the fully assembled vehicle?

Consider the automotive industry and how it goes about conducting crash safety tests on new vehicles. Vehicles undergo exhaustive crash testing from all angles on fully assembled vehicles. Testing individual components would not give a clear indication as to how the vehicle would fare in a real-world accident.

The same principle applies to software and illustrates why looking at the complete package is so critical. For example, if malicious code is embedded in the ReadMe PDF of a software package, neither SCA, SAST, nor DAST would see it. Analyzing the entire software package in its final executable state, including every commercial, proprietary, and open-source component, is the only way to check for hidden threats that SCA and other AST tools are not designed to find.

Below we can see a comprehensive risk assessment of the complete software package that security and risk professionals can expect from implementing complex binary analysis on the software they build, buy, and ship:

Malware
Complex binary analysis deconstructs software packages down to individual file format and object types without requiring source code or execution to find specific malware signatures and threat indicators.

Tampering
By deconstructing and analyzing the package down to individual components, complex binary analysis enables threat hunting by identifying human-readable threat indicators that signify potential novel malware and tampering threats. Identifying components that have been added, removed, or modified, coupled with flagging irregular behavior across reproducible builds is how complex binary analysis flags tampering before release.

Exposed Secrets
Complex binary analysis can identify exposed secrets like login credentials, API keys, certificates, or encryption keys that are exposed due to being hard coded as plain text, weak cryptography, packaging automation mistakes, insider threats, or other means.

Insufficient Hardening
To ensure proper safeguards exist within the software package, complex binary analysis will validate that proper security mitigations exist within the compiled code. For Windows and Linux binaries, complex binary analysis will flag missing vulnerability protections, insecure coding practices, outdated toolchains, inadequate prevention methods, and missing fortified functions.

Licensing Risk
Software supply chain risk can also stem from copy-left software licenses that pose company IP and legal risk. Complex binary analysis will verify that any open-source or third-party components are free of risky copy-left licensing terms that can compromise the compliance standing of your proprietary software.

Vulnerabilities
Not to be ignored, complex binary analysis will also analyze the entire executable software package for known vulnerabilities. To facilitate efficient remediation, the analysis prioritizes vulnerabilities that are exploited by malware patching has been mandated by CISA (Cybersecurity and Infrastructure Security Agency). It also automatically triages CVEs (Common Vulnerabilities and Exposures) identified in package dependencies when vulnerable code is absent from the SBOM, which minimizes the burden of manual review.

Complex Binary Analysis in Action: SolarWinds Orion

As an example, an analysis of the tampered SolarWinds Orion update conducted by ReversingLabs using Spectra Assure identified important static code behavior changes between the last non-tampered Orion update, the first tampered version (released months before the actual attack), and the final malicious update that contained the malicious backdoor code. Those changes included the addition of features for enumerating and tampering with user and account privileges; code for reading information about one or more running processes; and the addition of references to the kernel32.dll / advapi32.dll native Windows API.

While such changes are not malicious, per se, they are highly suspicious - especially in the context of a minor application update. As noted in RL's analysis of that hack: modifications like updating an application to reference native Windows APIs from .NET library all of a sudden is very unusual - what developers refer to as "code smell." Similarly, looking up user or account privileges is typically the first step in having them elevated. Running code at elevated privileges is done to perform a limited action, like copying files to restricted folders, manipulating running processes, changing system settings, etc.

In the context of analyzing a software update, these are all actions that must have a firm reason and explanation behind them. Procedurally, developers should be made aware of changes like this and should have to sign off on them. Adding them to a mature code base absent an explanation or approval is a big red flag.

The ability of complex binary analysis to detect the elusive threats hidden from software vendors is paramount. By analyzing software binary packages in real-time, complex binary analysis reveals more data than legacy technologies can, enabling software producers to understand the assembled software package's security posture fully.

It's Time to Rethink Your Software Supply Chain Security Strategy

By now it's understood that legacy application security testing techniques are ill-equipped to handle the multitude of threat vectors plaguing enterprise software supply chains. Focusing solely on vulnerabilities is a myopic approach that omits a huge portion of software supply chain attack categories like malware, tampering, leaked secrets, and others. Even tools like SCA that are designed to detect open-source vulnerabilities are only measuring a small portion of your organization's third-party risk profile - open-source software.

Institute the Critical Exam for the Applications You Build or Deploy

Complex binary analysis has the distinct advantage of serving as a critical final exam for software packages that your organization builds and buys. Unlike the fragmented, legacy approach to application security that requires a suite of tools to test only small subsets of code, complex binary analysis takes the final executable into account and deconstructs the entire software package down to the individual file, container, and component, including any artifacts added during the build phase. Assessing the entire software package in its final executable state is the only way to check for hidden threats that SCA and other AST tools are not designed to find.

For AppSec teams, complex binary analysis can easily be integrated into existing CI/CD processes to test the final software package between the post-compilation, and pre-deployment stages. By analyzing the final executable application as it would be deployed to production, we get full visibility into additional components or changes that are red flags for advanced threat exposures like malware, code tampering, suspicious behaviors, or tampered signatures. Instituting these measures as a stop-ship release gate gives AppSec and Development teams runway to address any critical issues at minimal cost, contrary to the millions that would be paid towards damages and brand loss in a breach.

For risk management organizations, complex binary analysis decouples vendor risk analysis from cumbersome processes, surveys, and pentesting that fail to provide full-risk visibility into COTS applications. Deconstructing the application to its core parts can help populate an actionable SBOM that enables cyber risk professionals to make more informed risk decisions before deploying COTS applications that host sensitive company and customer data. And because no source code is required, enterprise risk professionals can decouple themselves from the cumbersome processes and security surveys that usually come with vendor risk assessments.

Spectra Assure: Software Supply Chain Security Powered by Complex Binary Analysis

With software supply chain attacks on the rise, ReversingLabs is moving beyond traditional application security testing tools by applying this foundational technology to power Spectra Assure™, the first software supply chain security solution powered by AI-driven complex binary analysis. Spectra Assure delivers comprehensive visibility into the most complicated software packages, including commercial, proprietary, and third-party code, all without the need for the source code.

Just as a prism separates a single beam of light into its constituent colors, Reversing Labs Spectra Assure's AI-driven complex binary analysis deconstructs and reveals the facets of hidden threats, risks, and more. This advanced technology transforms a singular entity – a complete software package – into a spectrum of insights, uncovering malware, tampering, vulnerabilities, exposed secrets, and malicious behavior.

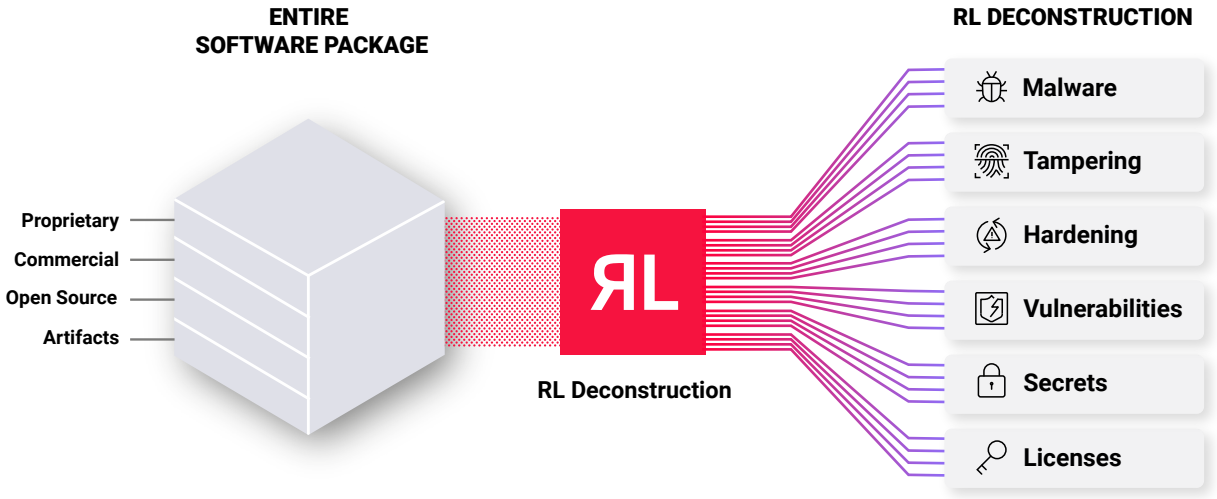


Figure 4: AI-Driven Complex Binary Analysis powering Spectra Assure dissects custom, commercial, and open-source components to find embedded software supply chain threats.

How Spectra Assure's AI-Driven Complex Binary Analysis Delivers Results

The Complex Binary Analysis powering Spectra Assure doesn't require any sort of code execution or even source code to conduct analysis. Instead, Spectra Assure thoroughly analyzes binary code to unveil risks and threats that would otherwise remain hidden. How Spectra Assure leverages complex binary analysis to deliver such rich and contextual insights is the result of a few key components:

The World's Fastest Software Deconstruction

Every executable package that is uploaded to Spectra Assure's complex binary analysis engine undergoes a process called recursive unpacking. Recursive unpacking uses high-speed binary analysis to identify, de-archive, deobfuscate, and unpack the underlying object structure including embedded executables, libraries, documents, resources, and icons. Think of this process as unstacking a Matryoshka, or Russian doll, with each layer of the doll symbolizing another layer of hidden risk.

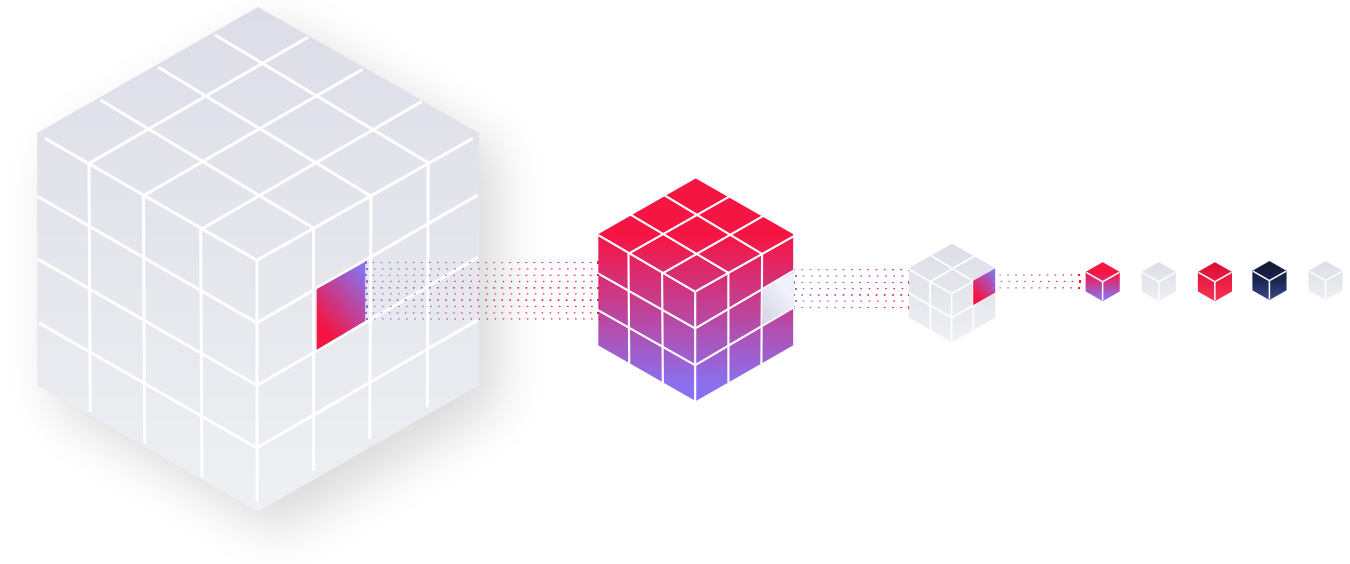


Figure 5: Complex Binary Analysis recursively unpacks the software binary to identify and classify components and threats.

Spectra Assure can unpack over 4800 file types down to individual DLLs, containers, and other post-build artifacts that can inflate the size of the application several times over. Beyond the files themselves, Spectra Assure also extracts proactive threat indicators (PTIs), and correlates each against a repository of over 3000 threat indicators and assigns a risk ranking accordingly.

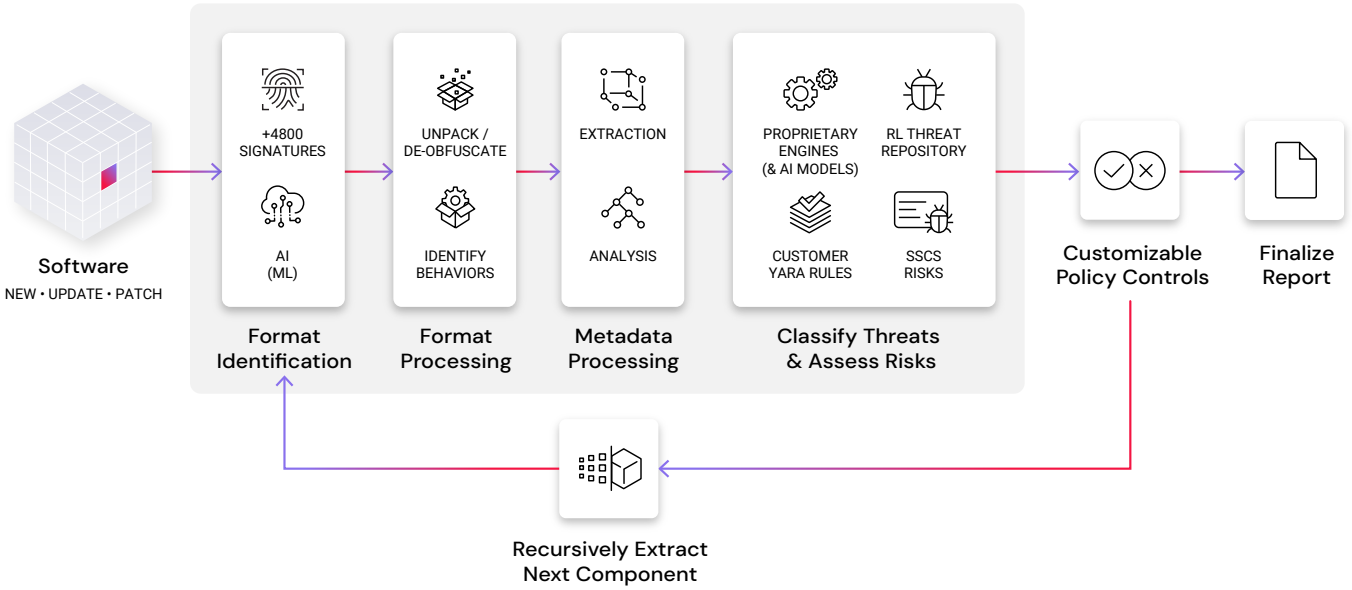


Figure 6: The AI-Driven Complex Binary Analysis powering Spectra Assure has a rigorous process for recursively identifying components and assessing embedded risks

And because Spectra Assure's complex binary analysis engine doesn't require any sort of execution, it is able to deconstruct even the largest and most complex packages in minutes instead of hours. Based on RL tests, a 1 GB package can take as little as five minutes to deconstruct. This rapid turnaround minimizes the impact to developers' release schedules and the time it takes to fully assess vendor applications for safe deployment.

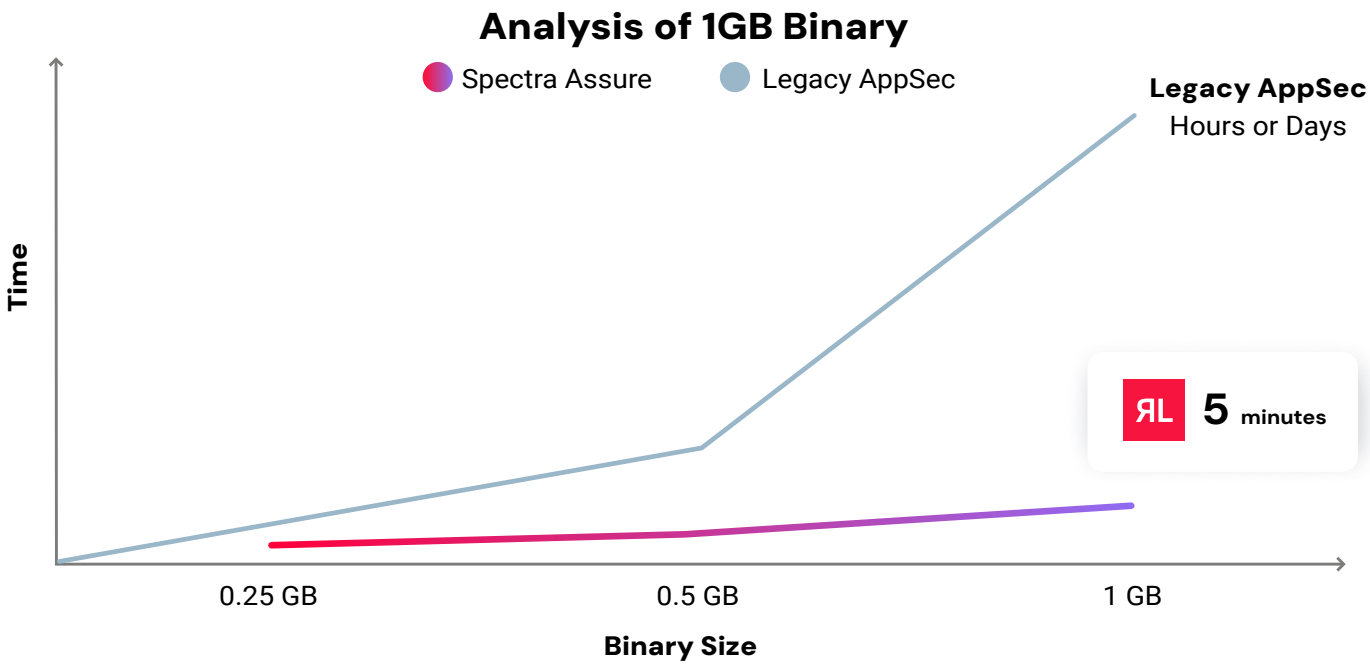


Figure 7: Complex Binary Analysis processes and classifies each individual component in milliseconds, with analysis of the entire software package taking only minutes.

Explainable Artificial Intelligence – xAI

Undocumented, novel malware requires a level of analysis that looks at specific threat indicators to determine whether the code is malicious or not. This is why Spectra Assure includes Explainable Artificial Intelligence (xAI) learning as part of its complex binary analysis engine. xAI helps to classify unknown malware, while providing security analysts a deep and actionable understanding of "why" the detection was determined. Spectra Assure's AI classification is based entirely on human-readable threat indicators, coded to identify which of these indicators apply to a piece of potential malware. For example, in the case of malware, Spectra Assure will, in human readable terms, outline threat indicators such as whether the code can read, write, or encrypt files, or iterate certain disk drives. A perfect illustration of this concept is how Spectra Assure will explicitly call out threat indicators that mimic the attack that impacted SolarWinds in 2019.

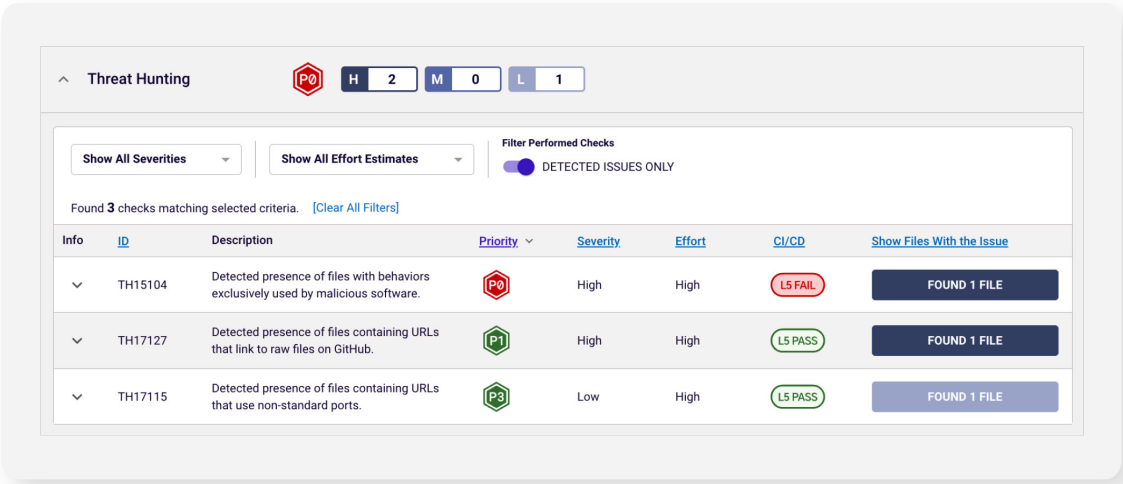


Figure 8: Spectra Assure's xAI identifies human-readable threat indicators that signify potential novel malware and tampering threats, enabling threat hunting within business-critical software packages.

The World's Largest Threat Repository

Of course, any threat analysis engine must also draw from a repository of known malware. When deconstructing and analyzing a software package for embedded threats, Spectra Assure's complex binary analysis engine draws from Spectra Assure's Threat Repository containing over 40 billion searchable pieces of malware and goodware. Beyond that, however, Spectra Assure also catalogs attack intelligence data such as malicious behavior and characteristics that could be an indicator of a software supply chain attack. In fact, 60 of the world's leading cybersecurity vendors rely on our Threat Repository as a feed for the solutions they provide to customers.

Differential Analysis

In many large development organizations, new code is pushed several times per day. With that much change happening within a single software application, it becomes vital to benchmark what normal behavior looks like and flag any deviations immediately. Spectra Assure's complex binary analysis engine provides a version-to-version differential analysis to help security practitioners flag new threats that may have been introduced with code changes. Spectra Assure compares subsequent versions of the application to check if any files have been added, deleted, or modified, or if the package is exhibiting erroneous behavior that could be an indicator of tampering or malware injection. This capability is also vital for security teams who wish to detect indicators of build environment tampering by validating build behavior integrity as part of their workflows. DevOps teams that have implemented two separate and isolated build environments can integrate automated behavior differential analysis to flag inconsistencies. In the end, differential analysis benefits security teams on two fronts: AppSec teams can flag new threats as they emerge on subsequent versions while tracking remediation statuses, and SecOps teams can proactively set up policies in their monitoring tools to mitigate the fallout from a potential attack.

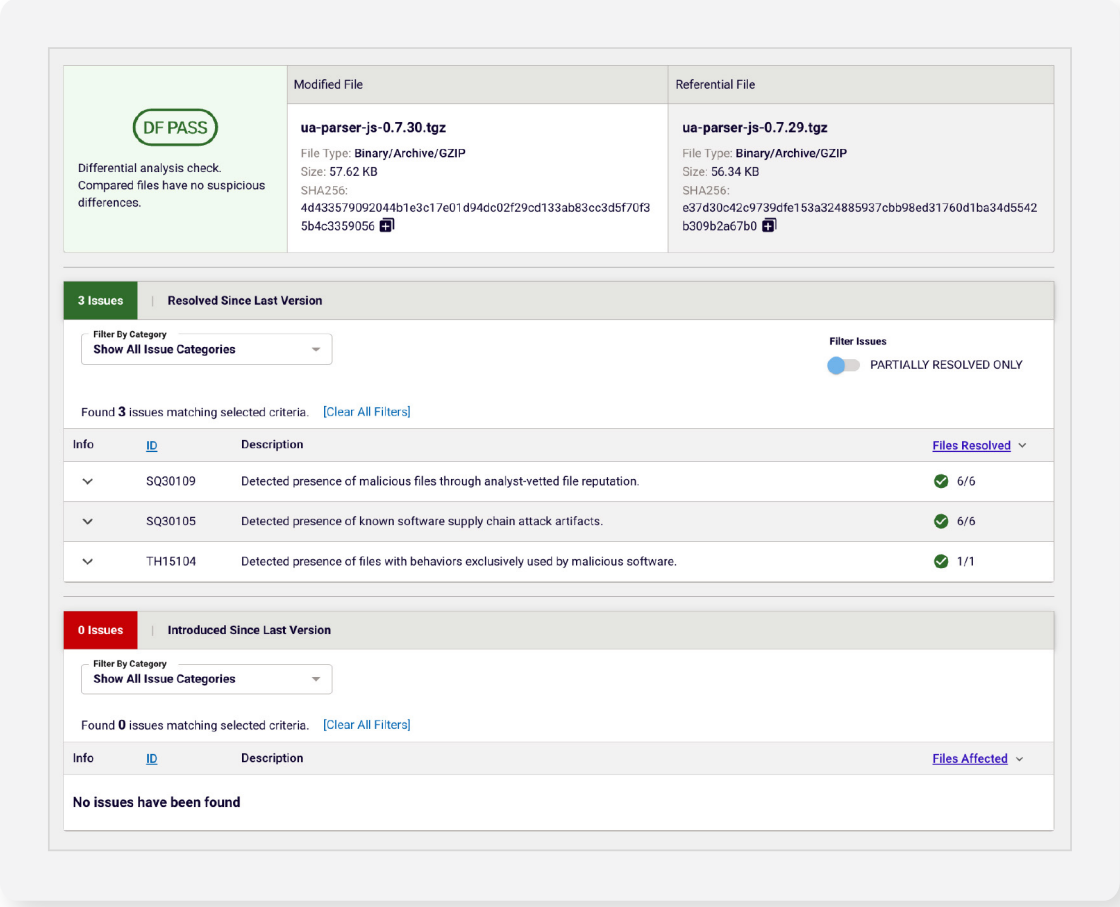


Figure 9: Spectra Assure enables AppSec teams to track new threats that emerge with new version releases, and verify the integrity of software build environments by identifying indicators of build tampering.

Spectra Assure's AI-driven complex binary analysis capability enables software producers and enterprise buyers to go beyond legacy application security tools with precise and scalable visibility into the most complicated software packages. By deconstructing binaries down to their core components and identifying software security issues before release or deployment, the insights provided enable software producers and enterprise software risk managers with the data they need to address software security visibility and ever-evolving challenges.

Spectra Assure: Software's Final Exam for Enterprise Producers and Buyers

By integrating Spectra Assure's AI-driven complex binary analysis into the software development lifecycle, organizations not only detect but also stop attacks before they materialize, establishing a robust and proactive defense mechanism against the sophisticated tactics of modern threat actors.

The impact of complex binary analysis on software supply chain security is profound. Spectra Assure enhances visibility and security within the SDLC by providing a comprehensive solution that addresses the complexities of modern software supply chain attacks. By assessing software packages at a post-compilation, pre-deployment state, Spectra Assure fills a major gap within modern CI/CD workflows left behind by legacy AST tools. It does this by combining complex binary analysis and a cutting-edge artificial intelligence model to effortlessly unpack even the most complex software packages and flag potentially dangerous threat indicators. The end result is full visibility into threat categories that legacy AST tools like SAST, DAST, IAST, and SCA are simply not built to detect. This robust security oversight across subsequent software releases means that security practitioners are empowered to ship secure software at scale without imposing cumbersome roadblocks.

Spectra Assure excels in establishing control stage gates to prevent attacks from materializing. Finding the proper oversight and control enforcement level is a strategic challenge for CISOs, who strive to deliver high-security assurance levels without impeding business agility. Spectra Assure achieves this balance by leveraging its complex binary analysis engine to analyze commercial (COTS) software packages and subsequent updates before deployment. By populating an actional SBOM that encompasses the entire executable package, not just open-source libraries, TPRM professionals can make more informed security decisions when assessing vendor risk and, subsequently, work with SecOps teams to establish the proper compensating controls post-deployment. This "final exam" for fully compiled software packages establishes trust, prevents the introduction of unforeseen risks, and strikes a harmonious balance between robust security and operational efficiency.

Spectra Assure Is Driving the Future of Complex Binary Analysis

While legacy AST tools have their place in preventing vulnerabilities from making it to the build, AI-driven complex binary analysis goes beyond vulnerability detection to flag more advanced threat categories like malware, tampering, suspicious behaviors, and secrets leakage. Adopting complex binary analysis as a critical final exam in the software development and procurement processes provides the depth of coverage to fully safeguard business-critical and customer data.

As the software supply chain attack surface evolves, the tools used to defend must also advance. Spectra Assure, powered by its AI-driven complex binary analysis engine, transforms how application security and risk management professionals approach software supply chain security, delivering the critical insights and data necessary to stay ahead of the most advanced threat actors.

About ReversingLabs

ReversingLabs is the trusted name in file and software security. We provide the modern cybersecurity platform to verify and deliver safe binaries. Trusted by the Fortune 500 and leading cybersecurity vendors, RL Spectra Core powers the software supply chain and file security insights, tracking over 40 billion searchable files daily with the ability to deconstruct full software binaries in seconds to minutes. Only ReversingLabs provides that final exam to determine whether a single file or full software binary presents a risk to your organization and your customers.

Get Started!

To learn more about ReversingLabs Software Supply Chain Security capabilities and solutions

REQUEST A FREE TRIAL

www.reversinglabs.com

REVERSINGLABS

Worldwide Sales: +1.617.250.7518
sales@reversinglabs.com