REVERSINGLABS

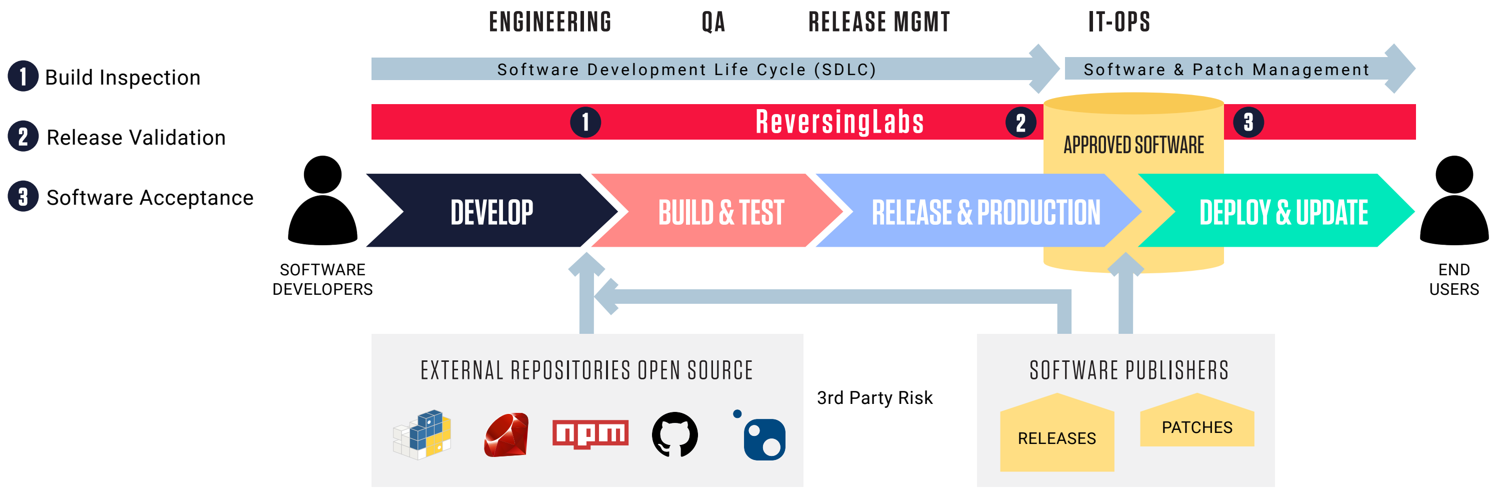
SOLUTION BRIEF

ReversingLabs Solutions for Managing your Software Supply Chain Risks

BRINGING TRUST TO YOUR DIGITAL BUSINESS

The software development lifecycle (SDLC) is not immune to compromise. In fact, it has emerged as a favored attack vector, acting as the perfect Trojan into your organization and your customers as it is inherently trusted, has access, and is not inspected by other security controls.

As a digital business, you are both developing and deploying software to optimize your business processes. Whether sourced from commercial Trusted Publishers, Open-Source Software (OSS), or through Internal engineering efforts, this software supply chain isn't always vetted to the level it should be. This ecosystem of third-party software suppliers is not accountable for the risk they could pose to your business, you are.



Challenges

The compromise of SolarWinds' Orion software is the latest example of how advanced attackers can successfully circumvent traditional security controls, and in this case place backdoor software into unsuspecting organizations through an otherwise trusted channel. Unfortunately, existing security solutions are limited to the discovery of vulnerabilities, open source licensing violations, or coding defects. They are not addressing the actual malware that may be unsuspectedly built into the code, maliciously injected into the code, or abused certificates intent on exploiting trust. When this software is placed into production, malware has successfully infiltrated the organization. And these cyber risks can lead to operational downtime, productivity loss, data loss, and reduced trust.

So where are these gaps?

- **Scanning source code is NOT enough**
  Application Security Testing (AST) technologies such as Static and Dynamic AST are simply not addressing malware that may unknowingly be present in published software applications. They focus on examining the integrity of the software code itself and identifying potential vulnerabilities. So if your source control was not compromised, SAST/DAST/SCA (Software Composition Analysis) and VM (Vulnerability Management) tools won't help.
- **Scanning binaries for malware is NOT enough**
  Software packages and installers represent large, complex files which often are not within scope of many security technologies such as Anti-Virus (AV) or dynamic analysis (sandbox) tools. In the case of the Orion package, a file in excess of 1GB with over 600K elements requiring analysis, these tools would not have inspected them and subsequently allowed software publishing and deployment to proceed. So if malware was well hidden (either the malware blended in with the code base, or the host package was too large and complex thus skipped by the scanners) or was novel with no malware signature, AV and sandbox tools won't help.

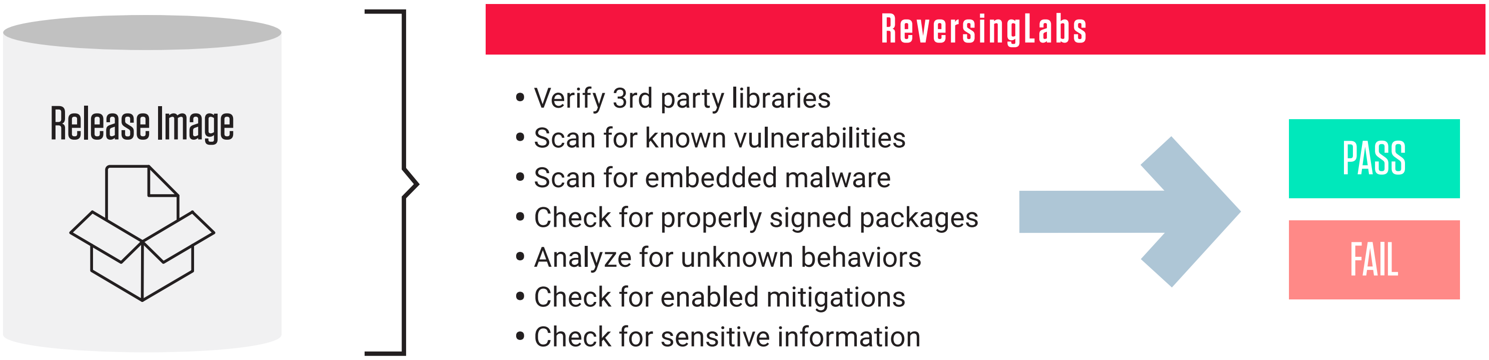
Solutions	Vulnerability insights	Malware insights	Certificate & Code Signing insights	Other critical insights e.g. software mitigations
File Analysis ЯEVERSINGLABS	🟨	🟩	🟩	🟩
AV & Dynamic Analysis	🟨	🟨	🟥	🟥
SCA, SAST & DAST and VM	🟩	🟥	🟥	🟥

- **Trusting certificates and code signatures is NOT enough**
  As a matter of expediency most security vendors assume the chain of trust to be in place. So if the code is properly signed, they don't bother inspecting. However, should attackers circumvent this trust, as was the case with SolarWinds, and inject malware into the code by leveraging the presumption of trust, they can easily bypass security controls and deliver their payload into the development process and code base itself.
- **Trusting software at a point in time is NOT enough**
  Should malware be detected in publisher's software after it's been published, how do you know whether you've been impacted? What if software at one point in time was approved, verified gold and free of malware, but later reclassified with malware as threat intelligence improves. Do you have the right tools in place to retrospectively hunt for the compromised software and identify the exploited hosts?

  As a producer, you'd have an obligation to thoroughly investigate any suspected compromise and inform your customers, and as a consumer, you're responsible to ensure what software that has been compromised is quickly remediated and the scope of the breach understood.

Solution

For those involved with software, whether part of an engineering team building software for internal or commercial use, or part of IT operational teams responsible for the deployment and management of 3rd party software, the challenges remain- is your software safe? The defense of today's digital business requires a new software analysis process, one that includes:



Release Image

ReversingLabs

- Verify 3rd party libraries
- Scan for known vulnerabilities
- Scan for embedded malware
- Check for properly signed packages
- Analyze for unknown behaviors
- Check for enabled mitigations
- Check for sensitive information

PASS

FAIL

How ReversingLabs can Help

Secure what you Build

as a Software producer or publisher

Continuously inspect your Builds

- **Open-Source Repositories -** Automate assessment of packages and libraries from GitHub, NPM, Ruby, PyPI, etc for malicious threats.
- **Large Scale Application Development -** Automate assessment of .NET, JAVA, OSS (Ruby, Python, etc) environments for malicious threats.
- **Incremental Build Checks -** Automate reviews of incremental builds to provide timely checks during the evolution of each software release candidate, and hunt for threats across the SDLC should the case arise.

Verify your Releases

- **Gold Image Verification -** Inspect the finished product for malware presence prior to publishing, and provide an in-depth analysis report where analysts can verify software is fit for purpose based on several suggested criteria.
- **Large Binary and Executable Support -** Support ability to scan release packages and code installers >1GB for malicious threats.
- **Container Support -** Inspect workload containers and dependencies in flight to production clusters.
- **Malicious Code Differential -** Automate assessment and comparison of code iterations against a standard gold image for malicious code changes.

Verify what you Buy
as a Software consumer or buyer

Verify and accept your Software

- **Independent Acceptance -** Automate analysis of final release packages without requiring access to source code or partnership with the software producers, thus you have completely independent assessments.
- **Application Catalog Validation -** Automate the inspection of newly published software prior to inclusion in the master software catalog for provisioning.
- **Audit Reports -** Provide complete software bill of materials (BOM) and audit record that all files have been inspected and undergone in-depth analysis for presence of unwanted and malicious intent.
- **Software, Patches and Updates -** Continuously monitor all incoming software enroute to Remote Monitoring & Management (RMM) and Software Configuration Management (SCM) solutions for deployment.

Case Study:
A Large Investment Organization

- **Their Challenge**
  Compliance mandates required they maintain an "enterprise software library" or what is often referred to as the approved software catalog, including "commercial off-the-shelf" (COTS) software that are typically procured, provisioned and managed as part of a software asset management program for financial and compliance reasons.
- **ReversingLabs Solution**
  The Enterprise Software Library was hosted on a global file share, with a web portal to facilitate access control and support document management. Via an integration connector, ReversingLabs was directed to automatically inspect all files added to their staging area within the storage volume. This has ensured the integrity of files before moving into an accessible segment of storage reserved for production use.
- **Benefit/Value Realized**
  By automating this particular business process, the organization has streamlined their software acceptance processes, advanced their software management practices and increased their security posture against supply chain risks.

Benefits

As digital business becomes more reliant on software, it's critical that organizations ensure both the software they use and the software they sell is free from risk, particularly as the exchange of information is quickly.

- **Accelerate delivery and deployment** by continuously monitoring all your digital channels for file and certificate based threats to alleviate risks before threats have a chance to propagate.
- **Improve software quality** by verifying the integrity of all third party dependencies, identifying vulnerabilities and potentially unwanted intent in the code, and ensuring software mitigations are in place.
- **Mitigate vendor risk** in your digital supply chain from third-party software by inspecting all software and associated dependencies.
- **Reduce time and expense** of manual oversight by automating the detection, analysis and classification of all your files.
- **Reduce brand risk** due to compromise and data loss, and jeopardizing your's or your customer's trust.
- **Increase confidence** in ongoing business operations among management and audit & compliance teams.
- **Eliminate exposure doubts** and reduce response times to get the answers. Should a compromise happen, you'll be the first to know everything about its impact on your environment, as well as your active customers.