Malware detection isn't enough to stop software supply chain attacks

Software development and supply chains are the new battleground for sophisticated adversaries. Development teams need to up their game on release acceptance testing to prevent compromises.

The common wisdom in information security circles is that cyber defenses need to "shift left" - moving from post-deployment "layered protections" into the application development process. Compromises of vendors like SolarWinds and CodeCov make clear that software development organizations and software supply chains are the new battlefield on which sophisticated cyber criminal and nation state groups are operating.

In response, organizations developing software are looking to fortify their CI/CD systems and workflows against supply chain threats and risks. As incidents like the SolarWinds "SunBurst" attack show, these risks, if undetected, can severely impact customers and damage both bottom lines and corporate reputations.

One common approach to supply chain risk is to simply extend endpoint- and perimeter anti-malware scanning tools into the development process. Anti-malware engines that are already scanning email attachments or file downloads can also inspect release packages and container images produced by development systems and workflows. That's a sensible approach - and malware scanning of development binaries can reduce cyber risk. However, the range of attack types and threat models observed in known supply chain attacks make clear that relying on anti-malware alone will not address some of the biggest cyber risks to development organizations, leaving them with little real protection against cyber adversaries.

Here are a few reasons that antivirus isn't enough to secure your development process:

KNOWN MALWARE IS A FRACTION OF SOFTWARE SUPPLY CHAIN RISK

There are millions of known malware families circulating on the Internet. But known malware accounts for a fraction of software supply chain compromises. That means anti-malware engines are likely to miss the vast majority of non-malware attacks. For example, attackers like Cozy Bear, the Russian APT group behind the SolarWinds hack, masked their attack using behavioral changes that mimicked developer's coding styles. Such changes are invisible to malware detection and have proven to be very hard to spot even with code reviews and vulnerability scans. Additionally, nation-sponsored hacking groups have the time and resources to develop and test novel or company-specific attacks to evade detection based on understanding malware families used in past attacks.

YOU WANT TO STOP ATTACKS BEFORE MALWARE IS PLACED IN SOFTWARE, NOT AFTER!

The attack on CodeCov's customers began with developer credentials that were left exposed in a container image (see Table 1). Assessing container images for exposed secrets like credentials or private keys before release provides the opportunity to stop data theft and avoid future CI/CD or build system compromises. But anti-malware tools aren't designed to conduct such assessments or detect exposed secrets, making this important early detection impossible.

SOFTWARE BUILDS AND RELEASE PACKAGES CAN BE TOO BIG TO SCAN

Many anti-malware tools aren't well suited to the task of scanning large files. Anti-malware engines have file size limits for effective scanning. While the upper limit varies with each product, container images and release packages exceeding 1 GB, which are now very common for enterprise applications, will cause scanning problems unless they can be broken up into multiple files so that each can be scanned separately. Correctly resizing binaries is time-consuming, error prone, manual work for what should be straightforward tasks of submitting the package or container for assessment.

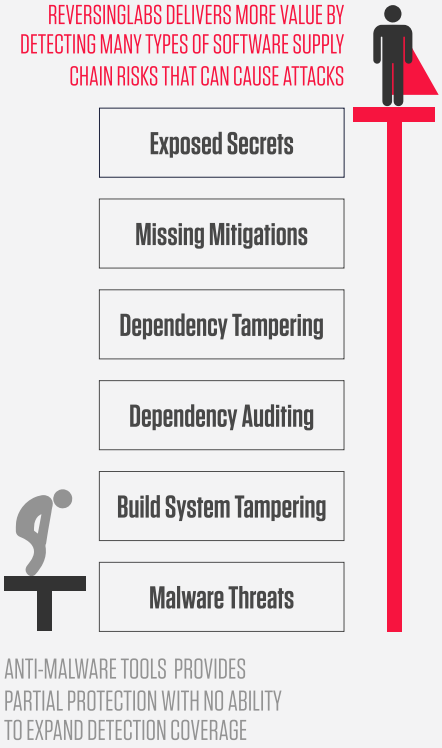
SCANNING WITHOUT COMPONENT REPORTING CREATES A FALSE SENSE OF SECURITY

Applications today are composed of scores or hundreds of components and libraries - some proprietary, many more licensed or pulled from open source repositories. However, some threats are buried so deep in an application dependency tree that many anti-malware engines are unable to detect them which leads to incomplete results. Beyond that, most malware tools are not transparent with exactly which software components have been scanned or what type of scan was performed as different options will lead to different results. This means there is no way to validate whether all, most or some of a software build or release package has been inspected for malware. Without this transparency, scan results can create a false sense of security.

Conclusion

As incidents like SolarWinds have shown: the consequences of inadequate security around software development processes and software supply chains are severe. Trust in the security of an application or platform - like trust in general - is difficult to earn back once it has been lost. Breaches of software development organizations, software build- and distribution infrastructure can turn legitimate software providers into unwitting purveyors of ransomware, nations-state spy software, data stealing malware and more. Such incidents erode your brand, encourage flight of customers to competing vendors and lead to costly software dependency upgrades, and more onerous compliance requests from customers and regulators.

To prevent such undesirable outcomes, organizations need to adopt a more holistic approach to securing developed code that expands software threat detection beyond known malware to encompass the full range of supply chain attacks that are in use by malicious actors.The value of such investments will increase with each secure release and satisfied customer.



A checklist for securing software supply chains

1. Provides a multi-layered approach, enabling discovery and remediation of a wide range threats and risks
2. Detects malicious intent with high accuracy, which eliminates time wasted trying to remediate false positives
3. Identifies build system compromises, software tampering and supply chain attacks that are invisible to anti-malware tools
4. Uncovers exposed secrets and credentials, which can prevent future attacks on your CI/CD
5. Identifies all components scanned to provide transparency and confidence in the results
6. Sees deeper into software dependencies to find threats on any layer, enabling you to discover hidden threats and manage risks that other tools miss
7. Delivers prioritization and remediation advice to focus on future development sprints
8. Tracks improvements with automated comparison against subsequent builds, making it easier to demonstrate compliance with secure development processes.
9. Supports the wider variety of file types and formats used in software development environments
10. Supports larger file sizes enabling container images to be scanned and saves time on manual file resizing

Software Supply Chain Threats	ReversingLabs Software Assurance	Malware Solution Capabilities
Compromised build system pulled in attacker-supplied source code files Webmin	Identifies build system compromise and software tampering	X
	Detects injected vulnerabilities	X
	Detects known malware	Detects known malware
Compromised build platform injected malicious behavior during each build. SolarWinds	Identifies build system compromise and software tampering	X
	Detects known malware	Detects known malware
Compromised source control platform and injected two malicious commits. PHP self-hosted git server	Identifies software tampering	X
	Detects injected vulnerabilities	X
	Detects known malware	Detects known malware
Added an innocuous dependency and later versions added malicious behaviors Nodejs_net_server	Detects software new behaviors between subsequent builds	X
	Detects known malware	Detects known malware
Leaked credentials & other secrets in release packages or container images CodeCov Part 1	Uncovers exposed secrets and credentials	X
Uploaded an attacker supplied script not built by the CI/CD system CodeCov Part 2	Detects changes in network communications	X
	Detects changes in software behaviors between builds	X
	Detects known malware	Detects known malware

Table 1. Detecting many types of threats and risks with a single scan delivers more value

Get Started!

www.reversinglabs.com

REQUEST A DEMO

Sources

Webadmin: https://www.webmin.com/exploit.html

SolarWinds: https://blog.reversinglabs.com/blog/sunburst-the-next-level-of-stealth

PHP self-hosted git server: https://news-web.php.net/php.internals/113838

Nodejs_net_server: https://www.theregister.com/2021/07/21/npm_malware_password/

CodeCov: https://about.codecov.io/apr-2021-post-mortem/

REVERSINGLABS

Worldwide Sales : +1.617.250.7518
sales@reversinglabs.com