EBOOK

3 Ways Contrast Helps Safeguard The Software Supply Chain

The Proliferation of Third-party Code in Applications Requires a Comprehensive Approach

Executive Overview

When it was revealed in mid-December 2020, a massive cyberattack on SolarWinds Orion infrastructure management software changed the lives of the IT and cybersecurity teams at thousands of organizations overnight. Characterized by one analyst as among the worst breaches of the past decade,[1] the attack left a "backdoor" on more than 18,000 networks[2] that gave the perpetrators—and apparently other cyber criminals—advanced access[3] to return with other attacks. Among the victims were multiple agencies of the U.S. federal government, including groups that manage top-secret information.[4]

This attack was a wake-up call for organizations to prioritize application security for all parts of their software supply chain, which has four elements:

- What you write: Custom code developed by in-house and contracted development teams
- What you build with: Software development tools that aid in-house developers in their work
- What you buy: Commercial off-the-shelf (COTS) Software-as-a-Service (SaaS) applications in use by an organization
- What you use: The numerous third-party libraries that most applications depend on

At the same time, the sheer number of libraries and the complexity of transitive dependencies make this difficult. Solutions by Contrast Security help organizations manage this complexity while bringing comprehensive observability of the entire software supply chain. This helps organizations respond effectively to zero-day attacks—even when patches are unavailable or not feasible to install.

> [W]hat if an organization could leverage the software factory's amazing skills at innovation, efficiency, and automation to 'manufacture' security along with software?
>
> Jeff Williams, "Security As An Output Of The Software Factory," Forbes, May 11, 2021

[1] Michael Novinson, "SolarWinds Hack 'One Of The Worst In The Last Decade': Analyst," CRN, December 17, 2020.

[2] Michael Riley, et al., "Russia-Linked SolarWinds Hack Snags Widening List of Victims," Bloomberg, December 18, 2020.

[3] Jason Lemon, "SolarWinds Breach Potentially Gave Hackers 'God Access': Ex-White House Official," Newsweek, December 16, 2020.

[4] Zachary Cohen, et al., "Massive hack of U.S. government launches search for answers as Russia named top suspect," CNN, December 16, 2020.

Table of contents

01

Complexities in the Software Supply Chain

Organizations in all industries have undergone massive digital transformation over the past several years, and software has been at the center of those initiatives. Software engineers face increasing pressure to deliver applications more quickly, and they have responded by dramatically streamlining and accelerating their work using approaches like Agile and DevOps.

One strategy that speeds development cycles is the use of open-source libraries and frameworks. According to Contrast's 2021 State of Open-source Security Report, the average application now contains 118 open-source libraries.[5] This approach pays significant dividends: A recent McKinsey report found that open-source adoption was the biggest differentiator for organizations in the top quartile of their Developer Velocity Index (DVI).[6]

SECURITY RISKS FROM OPEN-SOURCE LIBRARIES

But a proliferation of third-party code in applications also has its pitfalls. It introduces multiple complexities that are compounded by the velocity of development. In addition, the sheer volume of third-party code from a growing number of unknown sources far outweighs proprietary, custom code in the typical application today. This, in turn, introduces gaps in visibility and governance. Too often, the result of these deficiencies is the inability to respond quickly to zero-day exploits when they occur.

In the wake of a series of high-profile cyberattacks by nation-state threat actors, U.S. President Joe Biden issued an executive order in May 2021,[7] placing strict new standards on cybersecurity protections for any software purchased by the federal government. Protecting applications—and the software supply chain that makes them possible—is prominent throughout the order. And while application security professionals are responsible at most organizations for implementing the necessary safeguards, developers must continue to meet strict deadlines. This means that application security protections must be integrated directly into their native toolset.

> [T]he federal government must take action to rapidly improve the security and integrity of the software supply chain, with a priority on addressing critical software.
>
> "Executive Order on Improving the Nation's Cybersecurity," The White House, May 12, 2021

[5] "2021 Contrast Labs Open-source Security Report," Contrast Security, April 8, 2021.

[6] Shivam Srivastava, et al., "Developer Velocity: How software excellence fuels business performance," McKinsey, April 20, 2020.

[7] "Executive Order on Improving the Nation's Cybersecurity," The White House, May 12, 2021.

02 Keeping Pace with Proliferating Third- Party Components

When working against tight sprint deadlines, developers almost always default to using third-party open-source or commercial off-the-shelf (COTS) components to provide ready-made functionality for a particular business use case. Developers pull these components from a variety of sources, including public repositories and project sites. The result is complex dependency trees that can be difficult to track. Application security teams can quickly become overwhelmed with this volume of technical debt. In many cases, they struggle to answer the simple question, “What is in the application?”

This state of affairs would be difficult enough if every vulnerability in every library posed equal risk to an organization. But Contrast Labs research shows that 62% of libraries found in the typical application are inactive—that is, not used by the software.[8] What is more, within active libraries, fully 69% of library classes are inactive. This means that while third-party libraries comprise most of the lines of code found in many applications, only 9.4% of code in the typical application is from active third-party libraries and library classes. Common Vulnerabilities and Exposures (CVEs) in the remaining third-party code pose no risk to an organization.

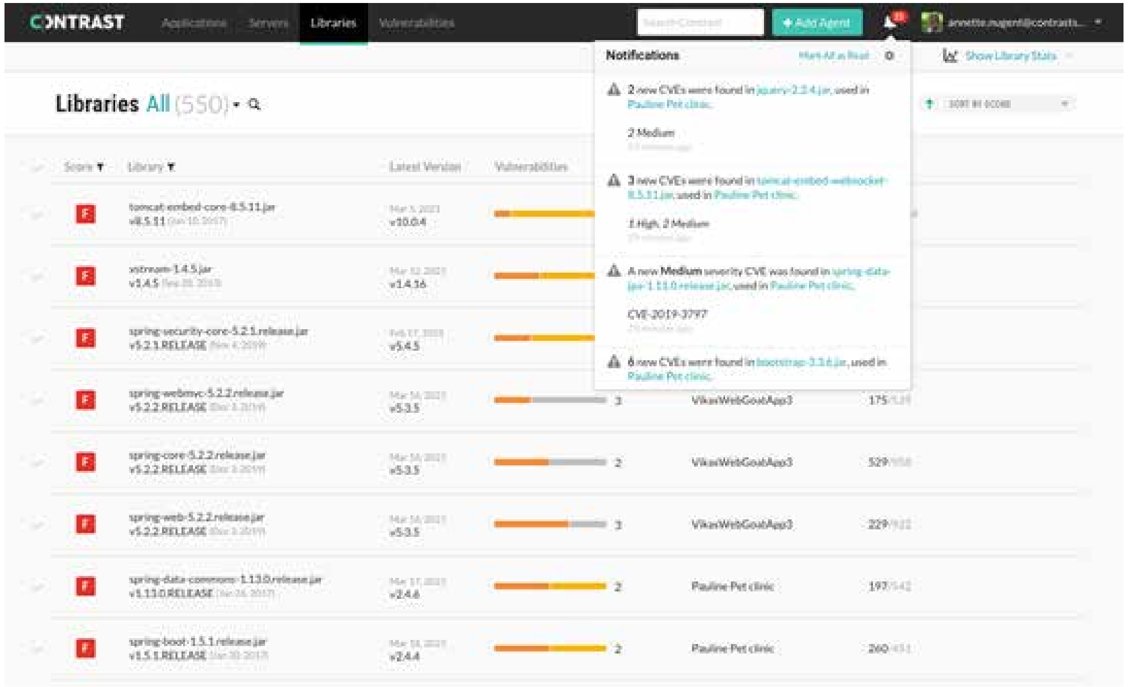
CONTRAST BRINGS ORDER TO THE CHAOS—WITHOUT SLOWING DEVELOPMENT CYCLES

Detailed insight into which libraries and classes are used by the software is essential to prevent application security teams from becoming overwhelmed, and from wasting time addressing vulnerabilities that pose no risk. They need a way to benchmark what third-party software assets are present in applications (both direct and transitive dependencies), understand where they are potentially exposed from both a security and licensing perspective, and prioritize what needs to be fixed.

[8] “2021 Contrast Labs Open-source Security Report,” Contrast Security, April 8, 2021.

The Contrast solution addresses these issues with the following features:

- **Contrast OSS** prioritizes the libraries that pose the greatest risk based on which libraries are actually used during runtime—down to the specific class, file, or module. It also triggers alerts about new vulnerabilities in already-deployed libraries (Figure 1).
- Contrast OSS fits into existing continuous integration/continuous deployment (CI/CD) workflows to benchmark against third-party software risk whenever new code is introduced, with no cumbersome scanning required. The Contrast OSS agent creates a real-time inventory of all third-party software assets mapped to their respective application and server environments.



Libraries All (550)

Notifications

2 new CVEs were found in jquery-2.2.4.jar, used in Pauline Pet clinic.

2 Medium

3 new CVEs were found in tomcat-embed-websocket-8.5.11.jar, used in Pauline Pet clinic.

1 High, 2 Medium

A new **Medium** severity CVE was found in spring-data-jpa-1.11.0.release.jar, used in Pauline Pet clinic.

CVE-2019-3797

6 new CVEs were found in bootstrap-3.3.6.jar, used in Pauline Pet clinic.

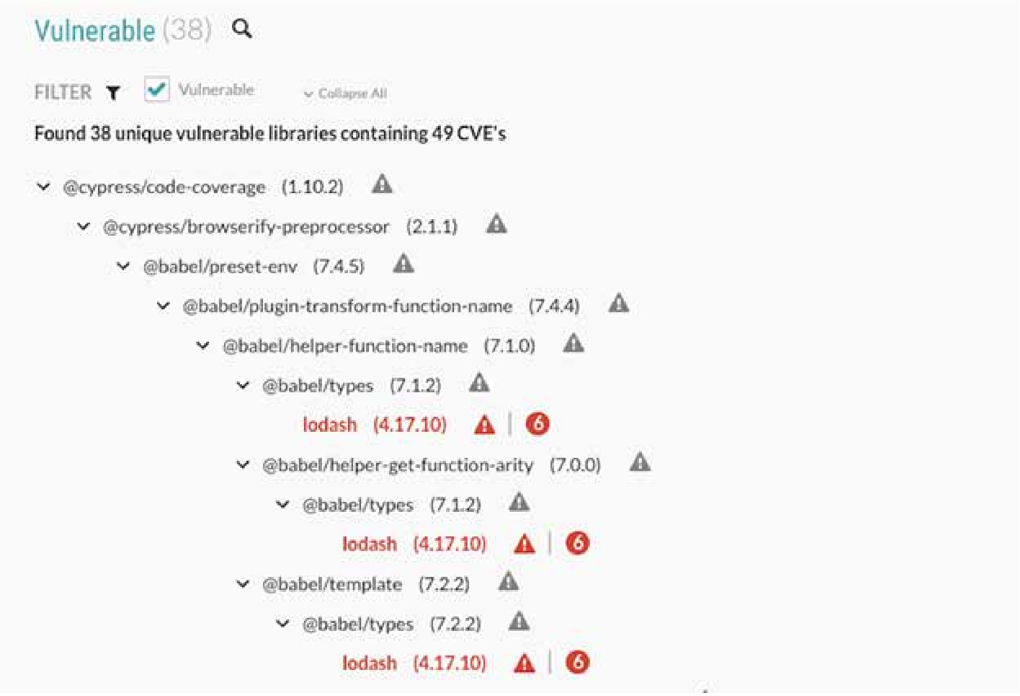
Score	Library	Latest Version	Vulnerabilities		
F	tomcat-embed-core-8.5.11.jar v8.5.11 (Jan 10, 2017)	Mar 5, 2021 v10.0.4			
F	xstream-1.4.5.jar v1.4.5	v1.4.16			
F	spring-security-core-5.2.1.release.jar v5.2.1.RELEASE	v5.4.5			
F	spring-webmvc-5.2.2.release.jar v5.2.2.RELEASE	v5.3.5	3	VikasWebGoatApp3	175/[illegible]
F	spring-core-5.2.2.release.jar v5.2.2.RELEASE	v5.3.5	2	VikasWebGoatApp3	529/[illegible]
F	spring-web-5.2.2.release.jar v5.2.2.RELEASE	v5.3.5	3	VikasWebGoatApp3	229/[illegible]
F	spring-data-commons-1.13.0.release.jar v1.13.0.RELEASE	v2.4.6	2	Pauline Pet clinic	197/[illegible]
F	spring-boot-1.5.1.release.jar v1.5.1.RELEASE	v2.4.4	2	Pauline Pet clinic	260/[illegible]

Figure 1: Contrast OSS provides automated alerts for new vulnerabilities in libraries already deployed.

- Using the **Contrast Command Line Interface (CLI)** within Contrast OSS, developers can run quick tests on their source code to check for vulnerable top-level libraries prior to committing code. Rapid testing for developers ensures that the code they ship into production is free of exploitable CVEs.

- Contrast CLI also highlights transitive dependencies introduced during the build process by populating a dependency tree within Contrast OSS (Figure 2). This provides much-needed observability and context into the deepening layers of dependency risk.

- Application security managers can institute scalable policy enforcement for vulnerable open-source libraries and license risk without forcing developer teams into manual code reviews. Contrast CLI enables specific rules that will block a build from entering production if specific CVE severity rules are not met.



Vulnerable (38)

FILTER ☑ Vulnerable ⌄ Collapse All

Found 38 unique vulnerable libraries containing 49 CVE's

```
⌄ @cypress/code-coverage  (1.10.2)  ⚠
    ⌄ @cypress/browserify-preprocessor  (2.1.1)  ⚠
        ⌄ @babel/preset-env  (7.4.5)  ⚠
            ⌄ @babel/plugin-transform-function-name  (7.4.4)  ⚠
                ⌄ @babel/helper-function-name  (7.1.0)  ⚠
                    ⌄ @babel/types  (7.1.2)  ⚠
                        lodash  (4.17.10)  ⚠ | 6
                    ⌄ @babel/helper-get-function-arity  (7.0.0)  ⚠
                        ⌄ @babel/types  (7.1.2)  ⚠
                            lodash  (4.17.10)  ⚠ | 6
                    ⌄ @babel/template  (7.2.2)  ⚠
                        ⌄ @babel/types  (7.2.2)  ⚠
                            lodash  (4.17.10)  ⚠ | 6
```

Figure 2: The dependency tree contextualizes how vulnerable dependencies are pulled into the application.

MANAGING COMPLEXITY BRINGS BUSINESS VALUE

The insight into all dependencies enabled by Contrast OSS and CLI can result in these tangible benefits:

- Reduced developer friction through prioritization of the libraries that are actually used by the application, saving hours of needlessly validating security results
- Full observability into third-party software assets—including layers of transitive dependencies—which contextualizes how vulnerable third-party libraries are introduced into applications
- Robust governance within a rapidly expanding third-party inventory with scalable security and license policy management

> The typical application contains 118 libraries, but only 38% are active. And within active Libraries, only 31% of library classes are active.
>
> “2021 Contrast Labs Open-source Security Report,” Contrast Security, April 8, 2021

03

Alleviating Observability Gaps in the Software Supply Chain

While open-source software is a common culprit for supply chain risk, visibility into the code businesses use to build, buy, and ship all feed into closing software supply chain security gaps. That is especially true for software vendors that ship commercial code as a revenue stream, further adding to their customers' software supply chain.

Supply chain attacks can take many forms, but when they do occur, the effects are vast. Attackers often attempt to use brute force to infiltrate organizations by targeting open-source components with known security defects. But more recently, attackers have begun to target the software supply chain as a vehicle for delivering malicious code to penetrate vulnerable systems. For example, the attack on SolarWinds Orion software, which was revealed in December 2020, impacted at least 18,000 networks,[9] some of which even belong to non-SolarWinds customers.

DEPENDENCY CONFUSION: MALICIOUS LIBRARIES MIMICKING LEGITIMATE LIBRARIES

Dependency confusion is a recent and prominent example of how attackers leverage the software supply chain as an alternative attack vector for accessing sensitive data.[10] Similar to typosquatting attacks that rely on erroneous spelling mistakes to trick users into installing malicious software, dependency confusion attacks provide an avenue for malicious code to be pulled into the application by tricking the package manager into pulling in code from a public repository that uses the same naming conventions as internally maintained software packages.

As developers become increasingly reliant on the software supply chain to meet tight deadlines, visibility into potential security gaps can become more obscure. At the same time, attackers such as hacktivist groups or nation-states have come to realize that the most efficient avenue to a breach is often through a backdoor. Businesses frequently lack the safeguards necessary to ensure the software they ship and the software they import is not a vehicle for a targeted software supply chain attack.

[9] Michael Riley, et al., "Russia-Linked SolarWinds Hack Snags Widening List of Victims," Bloomberg, December 18, 2020.

[10] Matt Austin, "Dependency Confusion: A New Third-party Risk for the Software Factory," Contrast Security, February 24, 2021.

CONTRAST BRINGS COMPREHENSIVE OBSERVABILITY ACROSS THE SOFTWARE SUPPLY CHAIN

Security teams need an automated, scalable software composition analysis (SCA) solution to benchmark supply chain risks across custom, commercial, and open-source code. Conversely, development teams need to be armed with the means to ensure that the software supply chain they rely on contains the proper security checks to close any possible attack vectors for bad actors.

The Contrast solution brings observability in a number of ways:

- The **Contrast Application Security Platform** enables application security teams to test custom code and third-party libraries through a single-agent deployment. Aggregating custom and third-party software risk into a single platform helps streamline remediation efforts and drastically lowers the total cost of ownership (TCO) associated with managing an application security program.
- Contrast CLI within Contrast OSS gives developers the assurance that they are not inadvertently opening attack vectors for malicious code. The tool alerts on all internal libraries that are at risk of dependency confusion and highlights suspicious versioning.

OBSERVABILITY ADDS TO BUSINESS VALUE

Comprehensive visibility brings several positive business outcomes:

- Bolstered security for not only the third-party components within the software supply chain but also for custom code, through an aggregated platform that ultimately saves valuable resources managing separate tools
- Improved security posture against targeted supply chain attacks such as dependency confusion, ensuring that native development tools are not erroneously introducing unnecessary risk

> “When a library is downloaded from the package manager, the first thing the library does is run the post-install script. This is one method by which a malicious payload can be delivered in a dependency confusion attack.”
>
> Matt Austin, “Dependency Confusion: A New Third-party Risk for the Software Factory,”Contrast Security, February 24, 2021

04

Enabling Quick Response to Zero-Day Threats

When new CVEs are published for open-source libraries, they are often accompanied with an explanation of how to exploit them as a way to validate the finding. The caveat with this is that attackers have access to this same information, leaving unpatched libraries open to exploitation. Many businesses cannot react fast enough when zero-day vulnerabilities are disclosed, creating a window of attack that can last for days or even months depending on the application exposed.

Companies often do not have the means to replace or patch a vulnerable library when a zero-day vulnerability is disclosed. This could be because a patch is unavailable or that an existing patch could potentially break the application. Organizations lack a scalable means to protect against targeted CVE attacks in the event a patch or update is not feasible.

CONTRAST PROTECTS AGAINST ZERO-DAY THREATS WITH CONTINUOUS VISIBILITY

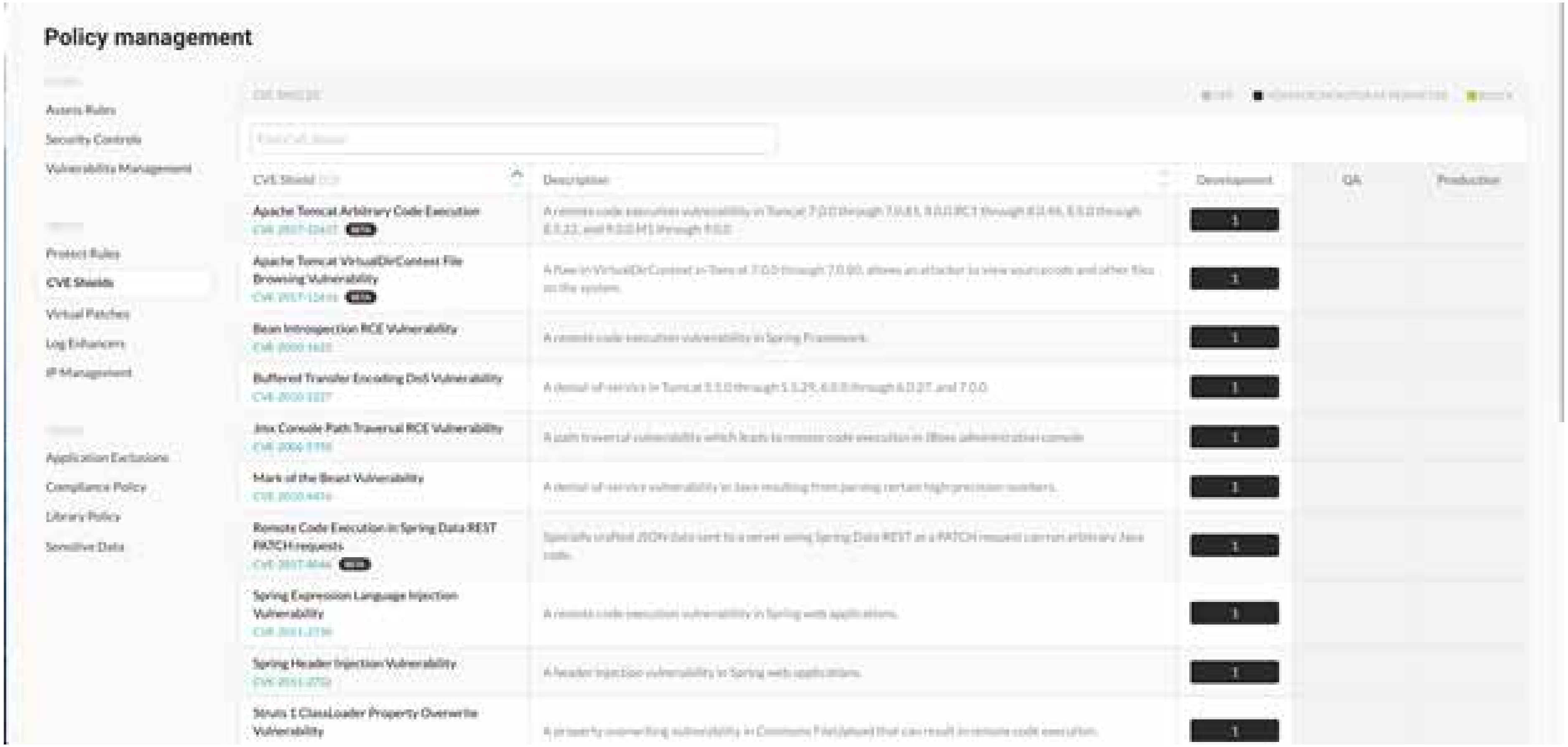
Vulnerabilities in third-party code are discovered every day—either by security researchers or by bad actors who gain access to a network. Businesses need to protect applications deployed in the wild by leveraging continuous visibility into third-party software assets and implementing controls against targeted attacks.

The Contrast solution provides protection against zero-day exploits in a number of ways:

- Contrast's single-agent Application Security Platform enables integrated open-source testing within native CI/CD workflows while also protecting against targeted CVE attacks in production. Runtime protection in production environments allows for security controls against targeted CVE attacks and vulnerability classes when patching or replacing a library is not an option (Figure 3).

- Contrast OSS sends automated email and messenger alerts when a previously secure version of a specific library is found to contain a new CVE. This allows the application security and development teams to coordinate a fix.

- Data shared between Contrast OSS and **Contrast Assess** allows for the detection of zero-day vulnerabilities before they are disclosed to the National Vulnerability Database (NVD).



Policy management

- Access Rules
- Security Controls
- Vulnerability Management

- Protect Rules
- CVE Shields
- Virtual Patches
- Log Enhancers
- IP Management

- Application Exclusions
- Compliance Policy
- Library Policy
- Sensitive Data

CVE Shield	Description	Development	QA	Production
Apache Tomcat Arbitrary Code Execution CVE-2017-12617	A remote code execution vulnerability in Tomcat 7.0.0 through 7.0.81, 8.0.0.RC1 through 8.0.46, 8.5.0 through 8.5.22, and 9.0.0.M1 through 9.0.0.	1		
Apache Tomcat VirtualDirContext File Browsing Vulnerability CVE-2017-12616	A flaw in VirtualDirContext in Tomcat 7.0.0 through 7.0.80, allows an attacker to view source code and other files on the system.	1		
Bean Introspection RCE Vulnerability CVE-2010-1622	A remote code execution vulnerability in Spring Framework.	1		
Buffered Transfer Encoding DoS Vulnerability CVE-2010-2227	A denial of service in Tomcat 5.5.0 through 5.5.29, 6.0.0 through 6.0.27, and 7.0.0.	1		
Jmx Console Path Traversal RCE Vulnerability CVE-2006-5750	A path traversal vulnerability which leads to remote code execution in JBoss administration console.	1		
Mark of the Beast Vulnerability CVE-2010-4476	A denial of service vulnerability in Java resulting from parsing certain high-precision numbers.	1		
Remote Code Execution in Spring Data REST PATCH requests CVE-2017-8046	Specially crafted JSON data sent to a server using Spring Data REST as a PATCH request can run arbitrary Java code.	1		
Spring Expression Language Injection Vulnerability CVE-2011-2730	A remote code execution vulnerability in Spring web applications.	1		
Spring Header Injection Vulnerability CVE-2011-2732	A header injection vulnerability in Spring web applications.	1		
Struts 1 ClassLoader Property Overwrite Vulnerability	A property overwriting vulnerability in Commons FileUpload that can result in remote code execution.	1		

Figure 3: Runtime protection from Contrast institutes compensating controls for targeted CVE attacks and vulnerability classes.

ZERO-DAY PROTECTION HELPS THE BUSINESS

Protecting against unknown threats brings a number of benefits::

- Greatly diminished likelihood of a costly breach resulting from the unavailability or unfeasibility of a patch.
- Continuous visibility into the third-party software layer removes the overhead of continuously cross-referencing vulnerability databases against libraries already deployed. This allows for a “set it and forget it” approach to zero-day threats.

> “In general, advanced persistent threat (apt) actors are more likely to have both the intent and capability to conduct the types of highly technical and prolonged software supply chain attack campaigns that may harm national security.”
>
> “Defending Against Software Supply Chain Attacks,” National Institute of Standards and Technology, April 2021

05

A Holistic Approach to the Software Supply Chain

Applications are the engine behind the ongoing digital transformation that is taking place in virtually all organizations today—a longstanding trend that was accelerated by the COVID-19 pandemic. One study found that a big majority (79%) of executives said that budgets for digital transformation had increased due to the pandemic.[11] And 8 in 10 consumers now say that 80% of their customer interactions are digital in nature than before the coronavirus.[12]

As software becomes an increasingly important driver of economic activity, it also becomes an increasingly attractive target for cyber criminals. Organizations whose developers are making the greatest impact on the business are making extensive use of open-source software, and it is incumbent on development and application security leaders to ensure that this innovation is protected.

Doing so requires comprehensive observability of the entire software supply chain, something that is only available with tools from Contrast Security. These tools help organizations speed development cycles while making the end product more secure.

> “
>
> [T]here is nothing incredibly difficult about managing vulnerabilities in the open-source supply chain. The problem is that it requires more man hours than most companies can provide, or better automation than most solutions deliver.”
>
> Kevin Townsend, “Library Dependencies and the Open Source Supply Chain Nightmare,” SecurityWeek, April 8, 2021

[11] John Koetsier, “97% Of Executives Say Covid-19 Sped Up Digital Transformation,” Forbes, September 10, 2020.

[12] Laura LaBerge, et al., “How COVID-19 has pushed companies over the technology tipping point—and transformed business forever,” McKinsey, October 5, 2020.

Contrast Security provides the industry's most modern and comprehensive Application Security Platform, removing security roadblocks inefficiencies and empowering enterprises to write and release secure application code faster. Embedding code analysis and attack prevention directly into software with instrumentation, the Contrast platform automatically detects vulnerabilities while developers write code, eliminates false positives, and provides context-specific how-to-fix guidance for easy and fast vulnerability remediation. Doing so enables application and development teams to collaborate more effectively and to innovate faster while accelerating digital transformation initiatives. This is why a growing number of the world's largest private and public sector organizations rely on Contrast to secure their applications in development and extend protection in production.

240 3rd Street
2nd Floor
Los Altos, CA 94022
Phone: 888.371.1333
Fax: 650.397.4133

Contrast
SECURITY

contrastsecurity.com