SOLUTION BRIEF

Contrast OSS Helps DevOps Manage and Triage Hidden Third-Party Library Risk

Executive Overview

The adage, “Security teams don’t know what they don’t know,” rings true when it comes to open-source software (OSS) vulnerabilities. While third-party components enable developers to deliver feature-rich software at speed, the wake of the SolarWinds hack and emerging vulnerability classes like dependency confusion demonstrate that the software supply chain is a popular attack vector for bad actors. In order to safeguard proprietary and customer data from costly breaches, organizations place significantly more scrutiny on where they source the commercial software they buy and, perhaps more predominantly, the thousands of open-source libraries utilized by their development team to help meet aggressive deadlines.

Open-source security risk goes beyond just the libraries developers actively use in their applications. Specifically, lack of insight into all of the different dependent libraries that get pulled into the application during continuous integration/continuous deployment (CI/CD) processes creates enormous visibility gaps in the application layer. Security and development teams often have no insight if libraries are being used when the application is run, and legacy software composition analysis (SCA) tools only provide a point-in-time assessment that cannot keep up with the volume of code changes. These factors create major backlogs, as DevOps teams cannot efficiently prioritize vulnerabilities that need to be immediately addressed.

As part of the Contrast Application Security Platform, Contrast OSS helps organizations prioritize critical vulnerabilities by tracking the libraries that actually get used by applications during runtime operation. It also provides development and security teams with comprehensive visibility of all OSS components to better understand the depth of risk that library dependencies can produce.

Software today is often built from as much as 90% open-source code—including hundreds of discrete libraries in a single application.[1] Developers routinely use open-source libraries (such as Apache) to introduce functionality to their applications at speed. What many developers do not know, however, is that for the top-level library to deliver on its functionality, it must call on directly dependent libraries. These libraries, in turn, may be linked to transitive dependent libraries—creating dependencies of dependencies.

Because these dependent libraries may include vulnerabilities, this structure creates layers of unaccounted risk. Application security teams typically have no insight into which of these libraries actually gets used when the application is running. This lack of visibility creates confusion when it comes to prioritizing vulnerability remediation—especially since any vulnerable libraries that get used when the application is running should garner the most immediate attention for remediation.

> “*Many people download open-source libraries, assuming they are safe— only to discover they’re infested with malware.*[2]

Third-Party Library Dependencies Create Visibility Problems

Security teams need to know about potentially vulnerable libraries in order to mitigate risk before shipping applications to production. Development teams need insight into which libraries (whether they be top-level or dependent) carry vulnerabilities so that they can better focus their remediation efforts. The ultimate goal is mitigation of software risk earlier in the SDLC. This ensures that developers spend less time fixing security defects and more time shipping code. It costs six times more to fix a bug found during implementation than to fix one identified during design; 15 times more if it is identified in testing; and 100 times more during regular maintenance once the code is in production.[3]

Contrast OSS Enables DevOps to Focus on Critical Library Vulnerabilities

Solving the key problems of open source means prioritizing vulnerable components that are actually used by the application for remediation effort by Security and Development teams. At the same time, it also translates into deprioritizing work on those components that are not in use. Accurately determining usage of third-party libraries requires directly observing and measuring the behavior of the application runtime. By assessing the application while it is running, teams can observe which top-level libraries and which dependencies are accessed in order to properly prioritize remediation efforts.

CONTINUOUS SCA DURING RUNTIME THROUGH INSTRUMENTATION

As an instrumentation tool embedded within the application itself, Contrast OSS can automatically perform software composition analysis (SCA) during the application runtime. It discovers all active open-source components and reports an exact bill of materials to Contrast TeamServer, which provides centralized management and reporting for policy-based control.

There is no need to run separate assessments with different tools. Because of its continuous assessment of a running application, Contrast OSS is more accurate than SCA tools relying solely on inspecting a "point-in-time" project configuration file.

REAL-TIME INSIGHT ON LIBRARY USAGE THROUGHOUT THE SDLC

Additionally, Contrast OSS reports on the extent to which each library is used. This occurs by detailing the total number of available library classes and the total called at runtime. With this information, security teams can prioritize remediation of libraries, ensuring the most critical libraries (those that introduce the greatest risk to the application) are prioritized first.

Contrast OSS also offers a means for developers to assess their applications at the earliest stages of the SDLC. Leveraging instrumentation, Contrast OSS detects if a particular library poses a risk to the running application. If a library is called at runtime, then it should be prioritized for remediation.

Contrast OSS also helps prevent potential dependency confusion vulnerabilities which can lead to malicious code being mistakenly pulled into the application from public repositories resulting in remote code execution or the exfiltration of sensitive data. Contrast helps benchmark dependency risk by automatically flagging libraries with suspicious versioning.[4]

DEVELOPER ENABLEMENT WITH CONTRAST CLI

In order to remediate, the team needs to know how the library got there. This is where Contrast's command-line interface (CLI) tool comes into play. The Contrast CLI allows developers to populate the dependency tree—a hierarchy of open-source library risks. If a vulnerable library is a top-level dependency, that top-level library needs to be updated. If the dependency is transitive, then the CLI's dependency tree helps contextualize which library is calling that transitive dependency.

The Contrast CLI also has features that provide instant feedback to developers, informing them if their internal libraries are at risk due to dependency confusion. The Contrast CLI flags any internal libraries in the node.js package as a dependency confusion risk; this includes flagging any libraries not in scope for a project, and showcasing the release history to indicate if an open-source project is still being maintained before merging.

> *The National Institute of Standards and Technology (NIST) sponsors the National Vulnerability Database (NVD)—a public repository for information on software vulnerabilities, including those in open-source software.*[5]

Expediting Workflows, Conserving Resources

Unlike outdated toolsets that assess open-source risks at a fixed moment in time, Contrast OSS continuously evaluates open-source libraries within actual running applications. This internal runtime visibility detects which open-source libraries contain vulnerabilities, which ones are called in runtime, and if they expose the organization to unnecessary IP riskdue to open-source licensing.

As part of Contrast's instrumentation-based Application Security Platform, Contrast OSS delivers a more accurate understanding of critical third-party libraries to help expedite CI/CD pipelines by prioritizing the vulnerabilities that actually matter to the running application. Focusing the remediation efforts of security and development teams saves time and money while simultaneously improving the security of applications.

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. OWASP notes that using old versions of open-source components with known vulnerabilities has been one of the most critical web application security risks in recent years.[6]

[1] Frank Nagle and Jenny Hoffman, "The Hidden Vulnerabilities of Open Source Software," Harvard Business School, February 24, 2020.

[2] Gilad David Maayan, "How to Make Your CSO Happy with Your Open Source Components," CPO Magazine, August 28, 2019.

[3] Mukesh Soni, "Defect Prevention: Reducing Costs and Enhancing Quality," iSixSigma, accessed February 23, 2021.

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

[5] "National Vulnerability Database," NIST, accessed February 23, 2021.

[6] Kirk Jackson, "Introduction to the OWASP Top Ten," OWASP, February 9, 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