Contrast SECURITY

Understanding the Risks of Open-Source Software

Executive Overview

Adoption of third-party open-source software (OSS) has increased significantly over the last few years. OSS refers to application components (e.g., frameworks and libraries) within the public domain that developers can use, modify, and share to help augment proprietary code developed in-house and to accelerate time to market. As a result, OSS has gained wide adoption and is practically everywhere—embraced by major corporations, including Walmart, JPMorgan Chase, and even Microsoft.[1] At the same time, OSS can present risks with licensing limitations as well as security trade-offs such as vulnerabilities and targeted attacks against open-source code. Further, managing OSS in DevOps can present workflow challenges.

A Competitive Differentiator for Accelerating Development

As software comprises an increasing portion of the value in many products and services, rapid software development has become a competitive differentiator. If a company can deliver new features to its customers faster than a competitor, it improves its chances to gain more market share and capture value. The demand for new and innovative software continues to be brisk—64% of organizations report an application development backlog (19% have more than 10 applications queued).[3]

In consideration of this pressure to alleviate development bottlenecks, writing custom code for all features and functionality of an application is very often not a practical option. To accelerate development while minimizing costs, organizations have embraced two main strategies that were rarely used 10 years ago: rapid development processes (e.g., Agile and DevOps) and the aggressive use of OSS.

> “Open-source vulnerabilities typically stem from poorly written code that leave gaps, which attackers can use to carry out malicious activities—such as extracting sensitive data or damaging a system.” [2]

Contrast SECURITY

> Proprietary software involves a central design by an organization that standardizes its process for new additions and fixes. Open source is a bit more chaotic, with contributors adding new features and improving the software all the time.[4]

Use of open-source code by developers grew at 40% from 2018 to 2019 and will continue, though tapering down to 14% by 2023.[5] OSS helps developers lower costs and reduce time to market by reusing existing components as building blocks for their applications. While many organizations originally avoided OSS due to concerns about provenance and licensing, today it is widely used in all markets, including government, financial services, and technology providers.

Where it was once rare to find open-source in commercial software, modern, state-of-the-art software may be built from as much as 90% open-source code, including hundreds of discrete libraries in a single application.[6] Open-source code is used by companies in all industries and of all sizes. In addition to well-known open-source operating systems (e.g., Linux, FreeBSD, OpenSolaris), enterprise users also leverage open-source productivity software, tools for administrators and developers, as well as source libraries used to build their own software.

> Much of the software that powers the world's largest companies, protects our personal data, or encrypts national security information is open source. It is often developed and maintained collaboratively by an army of thousands—from unpaid volunteers to employees at competing tech companies.[7]

Open Source Presents Benefits—and Risks

While OSS can energize application development cycles and reduce expenses, unmanaged use of open source introduces two significant risks: intellectual property (IP) risk from restrictive and reciprocal licenses, and security risk from components with code vulnerabilities.

LICENSING RISKS

OSS is free to users, but that doesn't mean it can be used without complying with other obligations. Open source can be issued under one of hundreds of different licenses, or under no license at all. Some licenses require developers to link to the code and others require that proper attribution be provided. The licenses range from extremely permissive versions such as MIT and Apache (which allow users to copy, modify, license, and sublicense the code freely) to restrictive licenses like that of GPL (which requires any changes to the open source or derivative works to be licensed as open source).

Using undeclared or restrictive licenses in a way that allows proprietary code to be defined as "derivative works" can therefore put an organization's intellectual property at risk. This complexity around licensing terms may also extend to other aspects of software commercialization (e.g., patent rights) that may require legal expertise to safely navigate.[8]

> Organizations unaware of their obligations under the open-source license (or not abiding by those obligations) can cause an oss user to lose intellectual property or experience a monetary loss.[9]

SECURITY RISKS

It is difficult to fully understand the security of OSS for two reasons:

- By design, OSS is distributed in nature. Thus, there is no central authority to ensure quality and maintenance.
- As OSS can be freely copied and modified, it is unclear what types of OSS are most widely in use.[10]

Essentially, OSS is no more or less secure than custom code. But as with any software, it can include errors that result in security issues. Since OSS is publicly available, security researchers can manually review the code to identify these vulnerabilities. The result is that each year thousands of new vulnerabilities are discovered and disclosed publicly, often with exploits used to prove the vulnerability exists.[11] According to OWASP, using old versions of open-source components with known vulnerabilities (Common Vulnerabilities and Exposures [CVE]) has been one of the most critical web application security risks in recent years.[12] Indeed, the number of disclosed OSS vulnerabilities grew by 50% year over year —from just over 4,000 in 2018 to over 6,000 in 2019.[13]

The heartbleed security bug in the OpenSSL cryptography library impacted nearly 20% of secure web servers on the internet (approximately a half a million instances) and allowed numerous data breaches—including the theft of 4.5 Million medical records from a large hospital chain.[14]

WHY ATTACKERS LIKE VULNERABILITIES IN OPEN SOURCE

While vulnerabilities in open source are not necessarily more dangerous than other vulnerabilities, they provide cyberattackers with an attractive attack vector.

Hackers understand that organizations often are unaware of the open source used in their environments, not to mention the presence of vulnerabilities in those components. Rather than working for months trying to hack an organization's custom code, criminals use publicly available exploits against a broad range of organizations to identify systems with vulnerable open-source components and compromise them.

Attacks on vulnerable open-source code can be just as effective as other approaches—and with far less effort. A prime example is the 2017 Equifax breach, which stemmed from a vulnerability in the widely used Apache Struts open-source development framework for creating enterprise Java applications—at a cost of at least $1.38 billion to date.[15]

Challenges of Managing Open Source

Managing the use of OSS components in a DevOps organization can be difficult. While setting open-source usage policies is a good starting point, experience shows that these policies can be easily bypassed—whether by accident or through negligence.

So, why is that the case? The simplest reason is that development teams continue to be tasked with and rewarded for delivering a specific set of capabilities by a specific date. OSS helps meet those goals—often becoming the first option for developers. Since OSS is freely available from repositories and project homepages, it is difficult to prevent it from entering an organization and application codebase without the same types of security and compliance scrutiny to which commercial software would be subjected in a standard procurement process.

Manual efforts to monitor the introduction of new OSS struggle to keep up with the pace of code churn in a DevOps environment, where dozens of new releases can be seen each day. Rapid changes to an application can result in a rapidly changing risk profile. However, stopping builds and slowing down pipelines to review changes manually would be an impractical "gating" practice—going against the natural development flow of DevOps and frustrating teams.

Risk can be introduced even when proper controls are in place to track open-source components. Older code versions with unpatched vulnerabilities may remain in repositories and workspaces to be pulled into projects instead of newer versions by mistake. Another common problem is dependent libraries, which may require other transitive components in order to work properly.

> The problem with OSS is even greater for smaller companies that prefer to use OSS to save time and resources but lack the necessary security measures to ensure the components they implement are secure enough to be implemented into the code.[16]

Clearing a Path Toward OSS Risk Mitigation

The many advantages of using open-source components in applications come with a cost—risk exposures in both licensing and cybersecurity. The "free" nature of OSS can unexpectedly carry cumbersome usage limits that impact IP rights. And as a favorite target of cyber criminals, open-source code vulnerabilities can become a moving target requiring constant vigilance to prevent bad actors from taking advantage.

In this case, manual management of OSS components in DevOps brings additional challenges that can bottleneck pipelines and impact delivery. To effectively realize OSS benefits, DevOps leaders must account for these critical inhibitors.

> OSS has become so popular, it makes up more than half of the analyzed enterprise codebase, many of which combine a great number of OSS components. Many people download open-source libraries, assuming they're safe, only to discover they're infested with malware.[17]

1 "The WIRED Guide to Open Source Software," WIRED, April 24, 2019.
2 "The Dangers of Open-Source Vulnerabilities, and What You Can Do About It," Security Today, August 19, 2019.
3 "The State of Application Development," OutSystems, May 2019.
4 "The Dangers of Open-Source Vulnerabilities, and What You Can Do About It," Security Today, August 19, 2019.
5 "The Application Security Market Will Exceed $7 Billion By 2023," Forrester, October 4, 2018.
6 "The Hidden Vulnerabilities of Open Source Software," Harvard Business School, February 24, 2020.
7 "How open-source software took over the world," CNBC, December 14, 2019.
8 "The Risks of Open Source Software," FindLaw, November 9, 2017.
9 "The Risks and Potential Impacts Associated with Open Source," DevOps.com, January 27, 2020.
10 "The Hidden Vulnerabilities of Open Source Software," Harvard Business School, February 24, 2020.
11 "National Vulnerability Database," NIST, accessed April 10, 2020.
12 "Introduction to the OWASP Top Ten," OWASP, February 9, 2020.
13 "Number of open source vulnerabilities surged in 2019," Help Net Security, March 13, 2020.
14 "The Hidden Vulnerabilities of Open Source Software," Harvard Business School, February 24, 2020.
15 "Equifax data breach FAQ: What happened, who was affected, what was the impact?," CSO, February 12, 2020.
16 "5 Best Practices for Managing Open-Source Components," DevOps.com, September 11, 2019.
17 "How to Make Your CSO Happy with Your Open Source Components," CPO Magazine, August 28, 2019.

Contrast SECURITY

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