Managing open source vulnerabilities

Canonical

ABSTRACT

This article explores open-source vulnerabilities, the challenges faced by organizations and best practices to address them.

Dorian Naveh

Senior Director, Global Alliances

Managing open source vulnerabilities

Embracing NIST as a security framework to provide effective cybersecurity

Executive summary

The usage of open source software has become an organizational mandate. Today, over 85% of enterprises[1] rely on open-source software due to its modularity, flexibility, and creative capabilities. Developer productivity and IT operations efficiency are cited as advantages of its use and has now become central to mission-critical technology stacks across multiple industries.

A result of open source becoming more pervasive in the enterprise is the realization that enterprise-class services, such as business continuance, have now become a key requirement. However, protecting open source software from sophisticated cyber attacks as well as securing and maintaining open source software can be complex and requires proven methodologies to address. Moreover, resource limitations, skill deficiencies, and complexities associated with open source pose real challenges for customers.

This paper will provide insights into open source vulnerabilities highlighting the challenges faced by organizations and offer best practices to overcome them for effective cybersecurity. The document will offer tangible steps by implementing a holistic approach to open source security via purpose-built security and compliance tools, and by choosing the NIST cybersecurity framework as a comprehensive security model.

Open source adoption…and vulnerabilities

The proliferation of open source software is clearly observed in the 2023 State of open source[2] report where more than 80% of the respondents indicated that over the past year there had been a significant increase in the use of open source throughout their organizations. The statistic represents a 5% increase from the previous year's study.

A chief benefit of open source is its emphasis on high security as a consequence of transparency by design and community support. IT leaders are aware of this fact and have come to trust the security of open source. In fact, according to industry reports, a majority of IT leaders see enterprise open source at least as secure, if not more, than proprietary software.

Despite the growing trust in open source, the distributed nature of open source and the fact that it comes from multiple and fragmented ecosystems do make open source maintenance and security a challenge. As a company's open source infrastructure increases in complexity, the secure integration, configuration, patching and upgrading of different tools and services gets exponentially harder as more applications and dependencies are added over time.

As expected, software maintenance and security feature high on the list of reported support challenges for companies that use open source software. This is expected as fully protecting your digital infrastructure and your business from security threats is extremely challenging. This problem is further exacerbated when considering enterprise business continuance.

There have been some real world examples in recent times that underscore the significance of robust open source vulnerability management practices. The Apache Log4J2 vulnerability and more recent OpenSSL vulnerabilities shed light on the importance of staying on top of updates and patches.[3]

The Log4J patch was available within days of the vulnerability becoming known in December 2021. However, a sobering 5% of all projects are found to still contain the vulnerability (comprising 11% of projects with a Java codebase) in the 2023 OSSRA report.[4]

Why open source security is so challenging

What makes open source software and infrastructure so hard to secure? The sections that follow explore some key factors behind this challenge.

Scarce resources

The most important barrier to enterprises adopting open source software has nothing to do with open source technology itself. It is the lack of internal skills available to test, operate, integrate and maintain open source.

Security operations teams are often understaffed and under-resourced, with hiring demand greatly exceeding the supply. Moreover, frequent training is required to keep up-to-date on the latest critical security aspects. In fact, it is reported that 41% of companies have zero skills to maintain their open source deployments.[5]

The technology stack in an organization is often composed of a wide range of technologies and languages, which makes it even harder to find employees with expertise that covers all its essential elements.

Software stack complexity

The security of a software stack is not the sum of the security of its individual elements. For example, two servers might be securely configured, but once connected, vulnerabilities might still be introduced. A single application may easily rely on dozens of underlying technologies, which quickly scales up the challenge.

The many different interconnected parts of a company's public, private or hybrid cloud landscape create an exponential increase in attack surface area. This poses a huge security challenge, compounded by new technologies being introduced as part of the constant drive for innovation.

Multiple dependencies

Knowing all the dependencies of each software component in your stack— and their dependencies in turn— is a challenge in itself. Companies are vulnerable to security breaches in components that may be hidden deep within their software's dependencies.

A surprising 45% of CISOs[6] report not having a clear view of their application stack and everything within them. This lack of optics may hide critical vulnerabilities that might be contained within and negatively impact compliance requirements.

The 2023 OSSRA report[7] underlines the prevalence of the problem when it found that of the almost 1,500 codebases it reviewed, 91% of projects contained outdated open source components and 88% of projects had at least one vulnerability, of which 48% were high-risk vulnerabilities, like Log4J.

There can be valid reasons why software is not updated in specific cases, but if the component is buried within multiple layers of dependencies, security teams are simply not aware that upgrades are required or Common Vulnerabilities and Exposures (CVEs) need to be patched.

Another consequence of having outdated software buried within a stack is that once a CVE is discovered, you often need to migrate to a newer version or are forced to backport the fix to be able to thoroughly mitigate the vulnerability in the specific packages you are using.

Varying security failures

When we consider vulnerability management in open source or any software, the default is to think of the technical vulnerabilities or CVEs in the packages that need detecting and patching.

But there are in fact three categories of potential failure to avoid or mitigate as part of an open source vulnerability management strategy:

- Insecure configuration and setup
- CVEs (technical errors/vulnerabilities)
- Human error

Successful open source security practices help security operations teams:

- Reduce the room for human error
- Prevent future complications through robust configurations
- Respond quickly and adequately to any CVEs that will almost inevitably surface

Open source security through NIST's cybersecurity framework

The best practice for open source security is a holistic approach. Following an established, comprehensive security framework helps to ensure that all security and compliance aspects are addressed.

The NIST CSF (US National Institute of Standards and Technology Cybersecurity Framework) is an industry standard for cybersecurity. The flexible framework is widely adopted by organizations of all sizes and industries as a valuable tool for enhancing cybersecurity resilience and aligning with industry best practices.

NIST CSF helps assess and improve an organization's cybersecurity practices as it encourages enterprises and institutions to develop a risk-based approach to cybersecurity. It asks the important question for organizations to consider. Namely, what are the security considerations relevant to an organization?

The NIST framework establishes a new way of thinking within an organization. It fosters collaboration between different departments and stakeholders to create a crucial security-minded network across an organization by elevating the importance of building a plan to address cybersecurity unilaterally.

The NIST CSF is built upon five core functions: **Identify**, **Protect**, **Detect**, **Respond**, and **Recover**. All five need to be focused upon and worked on continuously in order to make security operations a core element of an organization's operational excellence.

Identify

The "identify" function focuses on understanding and managing cybersecurity risks by identifying critical assets, assessing vulnerabilities and establishing risk management processes.

It is doing the due diligence beforehand necessary to map everything out so there is an understanding within the organization of both the risks and capabilities when it comes to the cybersecurity of critical assets. It creates a starting point for audits and helps identify areas that need improvement.

Protect

The "protect" function is about protecting the identified critical assets and managing potential vulnerabilities. Up front investment in this function is key to avoid future cybersecurity threats.

More than half the work of open source security is to ensure open source infrastructure is set up and configured properly to comply with established security baselines. This includes having the right tooling setup for vulnerability management, patching and upgrading.

Detect

Continuous monitoring is essential to be able to promptly detect and identify cybersecurity incidents. The "detect" function of the NIST cybersecurity framework involves implementing intrusion detection systems, security event monitoring and gathering threat intelligence to ensure timely detection.

To safeguard systems effectively, it is imperative to have state-of-the-art protection against vulnerability exploitation and malware. The utilized software for threat detection should enable seamless integration with management tools that patch and upgrade to significantly increase overall system security.

Respond

The "respond" function of the NIST framework occurs when a cybersecurity incident has taken place. It focuses on developing and implementing response plans to mitigate the impact and severity of a cybersecurity incident.

Response plans should include incident response procedures, communication channels and coordination with relevant stakeholders. It focuses upon developing an alert system so that the

correct people are notified who can contain or even eradicate the threat by taking appropriate action. An example of this could be to suspend user accounts or block firewalls.

Recover

Lastly, the “recover” function involves restoring normal operations and services after a cybersecurity incident, including data recovery, system restoration and processing the lessons learned.

Speed is of the essence to stop a security threat. As we’ve seen, companies struggle to remediate the vulnerabilities themselves quickly. Equally important, and even more difficult, is to enable a quick recovery after an incident and then rebuild the environment, re-establishing normal operations.

It is simply not sufficient to go back to the way things were before the incident. But rather, it is imperative that modifications be made to eliminate the potential for the incident to recur after the environment is installed and configured. This seems intuitive, but data shows that no less than 38% of companies that fell victim to a ransomware incident were hit again shortly after[7], because they failed to detect and eliminate the root cause that allowed their systems to be compromised.

Best practices for effective vulnerability management

Adopting a comprehensive cybersecurity framework like the NIST CSF requires many practical decisions to be made. In this section, we will cover important considerations and best practices to be considered as well as key features and capabilities to look for as you implement them.

Reduce human error through automation

Do not underestimate the impact of human error. Consider the human element involved in recent data breaches. Studies show that there were human elements involved in 74% of all data breaches last year, either via error, privilege misuse, use of stolen credentials or social engineering (2023 Data Breach Investigations Report by Verizon).

Automation is key to preventing human error. It can help increase security through automating configuring, patching and hardening processes, and it also reduces the number of tedious and repetitive tasks that are susceptible to error.

One such error-prone process is rebuilding infrastructure after a breach. Operational knowledge is usually concentrated in a handful of key individuals and the large number of manual steps involved means that the process is lengthy and leads to inconsistent results.

Using automation tools (like Canonical’s Juju) can help. Juju uses software operators (called Charms) that ‘encode’ your operational knowledge to enable you to redeploy complex environments and help manage day two operations like backups, scaling and recovery. Encoding operational knowledge in software also means that fewer administrators need to have access to the environments, effectively reducing the attack surface area and increasing security.

Secure the full stack, not isolated blocks

Software and security distributors are often guilty of talking about the security features of their products as if they existed in isolation or as if they were deployed in clean, greenfield environments. The reality looks very different:

All new infrastructure needs to coexist and integrate with many other systems, some of which may be outdated. Here again, the key is implementing a holistic approach.

Everything is connected and the defense you set up needs to offer in depth protection throughout your stack. Combining two technologies that are individually secure does not guarantee that the resulting system will be secure too. Just as importantly, if one package has a vulnerability, it does not mean that the combined system can be compromised.

Cybersecurity increases when the infrastructure architecture is designed to isolate elements in your technology stack through segmentation and confinement. Strict confinement ensures that the application is isolated and cannot access or modify critical system resources without explicit permission.

Often, companies' security operations tend to focus only on the open source applications at the top of the stack. But, the best practice is always to include the effects of vertical integration into your security considerations when testing and implementing new software. Seek out solutions that are vertically integrated offerings that cover everything from the operating system you deploy on bare metal, to container images and all the way to application automation.

Prevention through secure configuration

The most effective software security strategies center on constructing resilient systems that have a vulnerability management policy in place and need minimal human intervention.

System configurations are essentially a trade-off between usability, performance and security. Industry standards like the CIS benchmarks or DISA-STIG provide hundreds of configuration recommendations to increase the security posture of software deployments and lock systems down. However, the sheer number of configuration steps makes manually hardening and auditing a system a tedious and error-prone process.

Therefore, to run regulated and high-security workloads and allow easy audits, it is advisable to use trusted automation tools that can conform to the chosen cybersecurity and compliance frameworks.

A good example is the Ubuntu Security Guide, which streamlines the configuration process and satisfies requirements for hardening and compliance profiles, such as the FIPS 140-2 and Common Criteria certifications.

Systems carrying dedicated workloads can often be hardened further to reduce their attack surface. Use the Ubuntu Hardening guidelines to set up the most secure infrastructure with as few trade-offs as possible.

In reality, implementing a consistent hardening and security patching strategy is one of the most difficult things for IT teams to get right. Solutions should make security patching and hardening within reach and easy to implement for teams of all sizes.

When using open source, a secure source and a solid community behind the project are essential for open source security. It is important to be aware of its provenance, dependencies and upstream connections, especially in cases when there's no vendor backing the project.

Patching made easy

Once security vulnerabilities are identified, the effort required to patch them depends on the tooling used. This can be accomplished manually by tracking each security vulnerability and the corresponding patch notice, and then the security operator applies the appropriate software patches.

But security professionals looking to strike a balance between security, usability and availability must leverage automation. The gold standard is security patching that can be automated at scale and audited on the fly with on-demand reports.

Look for turnkey security patching solutions that work in even the most restrictive environments through unattended upgrades. Implemented together, they can help reduce the average CVE exposure time from 98 days to just one for the most critical vulnerabilities.

Security incidents might not be entirely avoidable, but a consistent hardening and security patching strategy will go a long way in deterring ordinary, unsophisticated threat actors from easily breaking into your systems.

Threat detection integration

Not all threats are preventable. That is why it is important to make sure you can quickly contain the threat and restore your operations.

The speed of your response is determined to a large degree by how closely connected the detection and response measures are in your systems, as this enables you to tackle the emergency as fast and efficiently as possible.

It is important to work with companies that deal with threat detection and response through partnerships with leading security vendors, and can streamline recovery after an incident, with software operators that automate common actions across the most popular open source applications. Look for affiliations with vendors of vulnerability management platforms like Tenable Nessus, malware detection systems like Microsoft Defender and infrastructure security tools like Aqua Security.

Conclusion

The pervasiveness of open source within the enterprise have increased the need to pay close attention to security and vulnerability management in order to stave off cybersecurity threats. This can be taxing on organizations struggling with scarce resources, software stack complexity, fragmentation, and talent shortages. Embracing mature and comprehensive security frameworks,

like NIST, has become an organizational imperative. In addition, to address critical vulnerabilities, organizations should acquire software from vetted, secure sources and rely on automation tooling for maximum efficiencies.

References

1. Embracing the Enterprise Open Source Mandate, Canonical
2. 2023 State of open source Report, open source Initiative & OpenLogic (Perforce).
3. Linux security patches whitepaper, Canonical.
4. 2023 open source Security and Risk Analysis Report, Synopsis, Inc.
5. open source survey 2022, open source Initiative.
6. Ten takeaways from the 2023 State of open source survey, Voices of open source.
7. 2023 open source Security and Risk Analysis Report, Synopsis, Inc.
8. 2023 ransomware insights, Barracuda Networks, Inc.