Canonical

A guide to open source vulnerability management

Spend less time tracking dependencies while improving your security posture with a comprehensive approach

Contents

Executive summary

In this whitepaper we provide insights into open source vulnerability management, highlighting the challenges faced by organisations and offering best practices to overcome them for effective cybersecurity.

Open source usage has increased significantly, permeating most organisations' technology stacks. Despite open source being perceived as secure, maintaining and securing open source software can be complex. Scarce resources and skill shortages, software stack complexity and a multitude of dependencies pose challenges.

We emphasise the importance of a holistic approach to open source security and highlight the NIST cybersecurity framework as a comprehensive security model. This guide discusses the five core functions of the framework: Identify, Protect, Detect, Respond and Recover.

Each best practice is accompanied by practical examples of how Canonical, its tools and expertise can support their implementation.

We conclude by highlighting how Ubuntu Pro simplifies open source vulnerability management by offering comprehensive security and compliance features, automated patching, and expert support.

Introduction

Essentially all technology stacks these days contain open source and more of it each year. In the 2023 State of Open Source[1] report 80% of respondents said that over the last year there had been an increase in the use of open source in their company, with 41% even reporting a significant increase.

Has your organiation increased the use of open source software over the last year?

Yes, significantly	41.28%
Yes	38.76%
Remain the same	18.46%
Reduced the use of open source	1.49%

Source: 2023 State of Open Source Open Source Initiative & OpenLogic (Perforce).

A chief benefit of open source is its high security as a consequence of its transparency by design and community support. IT leaders know this, they trust the security of open source and 89% of them see enterprise open source as more secure or as secure as proprietary software according to industry reports.

Despite this high level of trust, 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.

Unsurprisingly, maintenance and security feature high on the list of reported support challenges for companies that use open source software.

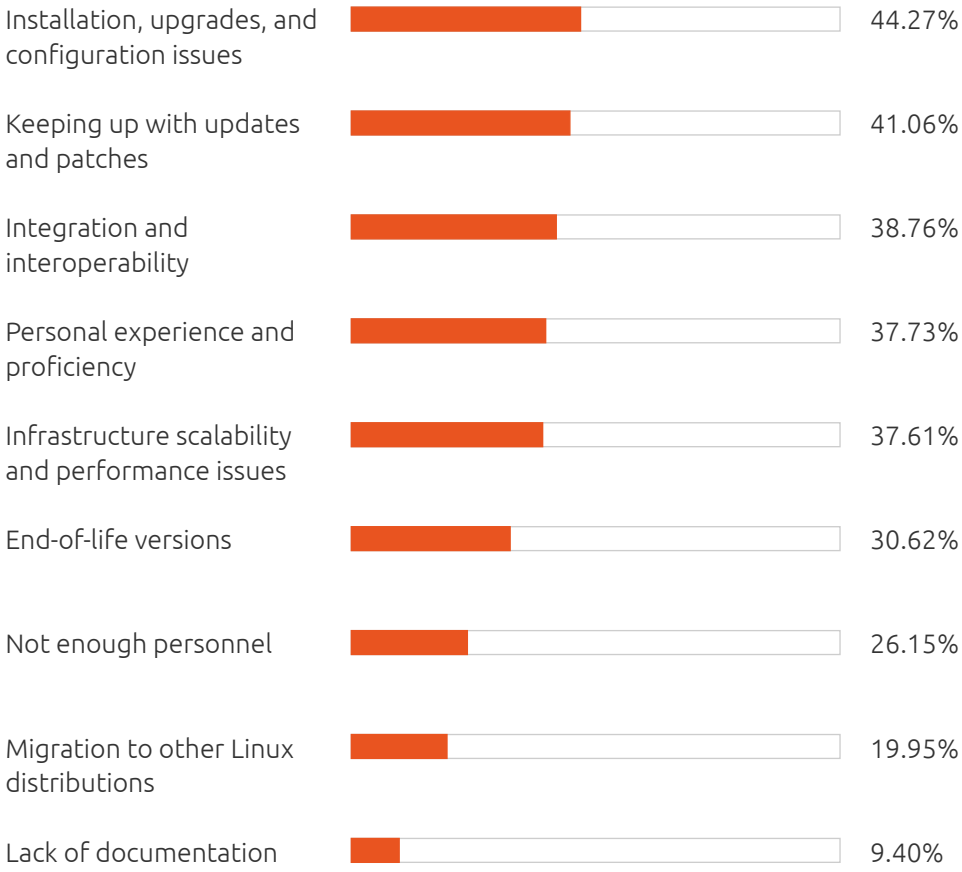
And with good reason. In reality, it's practically impossible to fully protect your digital infrastructure and your business from security threats. Security is always going to be an ongoing endeavour, there is no end to it if it is taken seriously.

There have been a few cautionary tales 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.[2]

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.[3]

It's hard to get open source security right consistently and most of us are aware of it. Organisations report their main support challenges when it comes to their open source infrastructure as follows:

What are the main support challenges with your open source software infrastructure?



Source: 2023 State of Open Source Open Source Initiative & OpenLogic (Perforce).

Why open source security is so challenging

In the cause of a problem lies the essence of its solution. What makes open source software and infrastructure so hard to secure?

In the sections that follow, we 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 to test, operate, integrate and maintain open source.

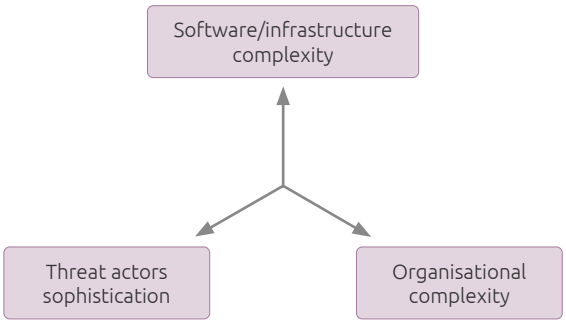
Security operations teams are often understaffed and under-resourced, with hiring demand currently greatly exceeding the supply and frequent trainings required to keep knowledge up-to-date. In fact, 41% of companies have zero skills to maintain open source deployments.[4]

The technology stack in an organisation often comprises 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. Two servers might be securely configured, but once they are connected, vulnerabilities might still be introduced. And one application easily relies on dozens of underlying technologies, quickly scaling 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 challenge for security operations, compounded by new technologies being introduced as part of the constant drive for innovation.



A kaleidoscope of 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 in their software's dependencies.

A contributing factor is that developers often pull unnecessary dependencies that they don't need or use for the feature they want to develop, increasing the attack surface area.

As a result, a surprising 45% of CISOs[5] report not having a clear view of their application stack and everything it comprises, which automatically blurs the view on the vulnerabilities these might contain and negatively impacts compliance.

The 2023 OSSRA report[6] 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 in multiple layers of dependencies, security teams often are simply not aware that upgrades are required or Common Vulnerabilities and Exposures (CVEs) need to be patched.

Another consequence of having outdated software in your 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.

The varied origins of 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 your 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

Holistic open source security: the NIST cybersecurity framework

The only secure approach to open source security is a holistic approach. Following an established, comprehensive security framework helps to cover all security and compliance aspects.

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 organisations 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 organisation's cybersecurity practices as it encourages enterprises and institutions to develop a risk-based approach to cybersecurity. It asks the question, what are the security considerations relevant to your organisation?

A key element is that the process of establishing a cybersecurity strategy fosters collaboration between different departments and stakeholders to create a crucial security-minded network across an organisation.

NIST CSF is built upon five core functions: **Identify**, **Protect**, **Detect**, **Respond**, and **Recover**. All five need to be addressed - and worked on continuously - to make security operations a successful, steady effort.

Identify

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

In essence it is doing due diligence beforehand. You map everything out so there is an understanding within the organisation 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 second function, “protect”, is about protecting the identified critical assets and managing potential vulnerabilities. As so often in life, prevention is much better than a cure.

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 utilised software for threat detection should enable seamless integration with management tools that patch and upgrade to significantly increase overall system security.

Respond

Inevitably, a cybersecurity incident will take place and it will be time for the function “respond” of the NIST framework. It focuses on developing and implementing response plans to mitigate the impact of cybersecurity incidents.

This includes incident response procedures, communication channels and coordination with relevant stakeholders. Or to put it differently, it is about alerts being sent so that people are notified and contain or even eradicate the threat by taking appropriate action, for example suspending user accounts or blocking firewall rules.

It is trickier than it might seem to find a balance of when the alarm bells should go off and how loudly. Many security operators get notification fatigue because too much is flagged as potentially dangerous, but on the other hand, the very high risk of not being alerted appropriately in case of real trouble is obvious.

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 the security threat. As we've seen, companies struggle to remediate the vulnerabilities themselves quickly. Equally important, and even more difficult, is a fast recovery after the incident, rebuilding the environment and reestablishing normal operations.

It is not sufficient to simply replicate what was there before the incident, then it is imperative that modifications make it impossible 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 NIST CSF still requires plenty of practical decisions to be made. In this section we cover important considerations and best practices, as well as key features and capabilities in Canonical's portfolio that can help you implement them.

Adopting a comprehensive cybersecurity framework like NIST CSF still requires plenty of practical decisions to be made. In this section we cover important considerations and best practices, as well as key features and capabilities in Canonical's portfolio that can help you implement them.

Reduce human error through automation

To start off with a general rule, do not underestimate the impact of human error. Take data breaches as an example. There was a human element 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 Juju can help. Juju uses Charms (software operators) 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

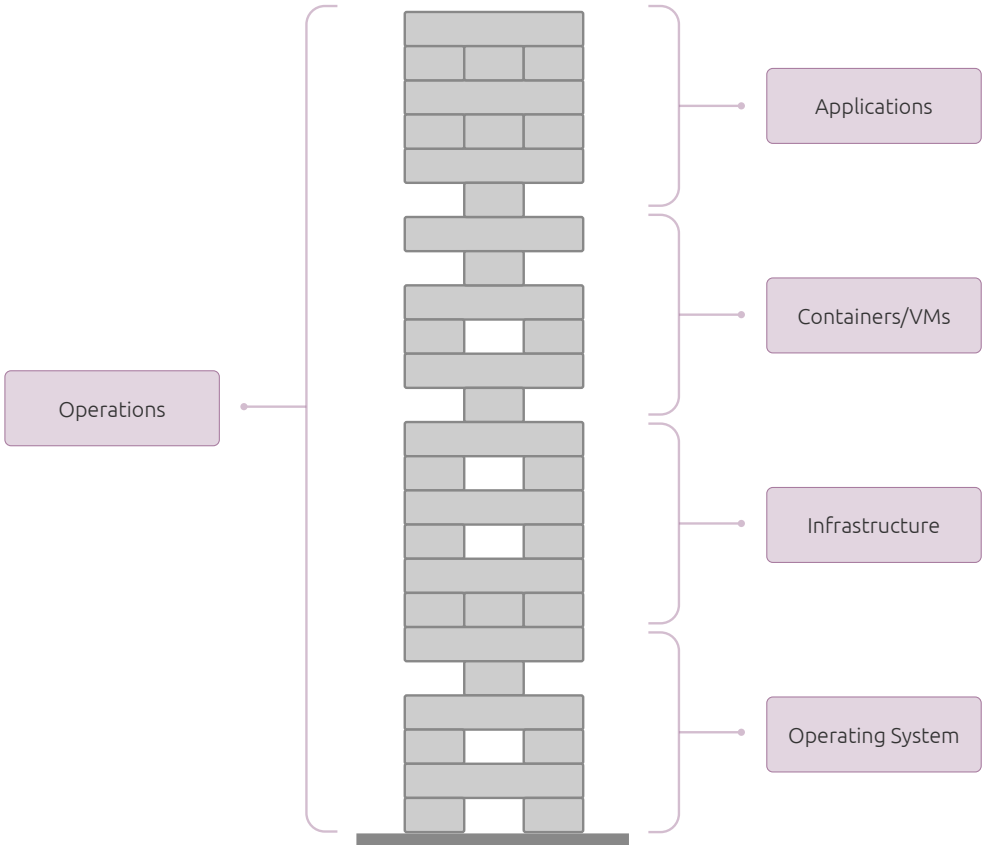
Software and security distributors are often guilty of talking about the security features of their products as if they existed in isolation or as if they were deployed in clean, greenfield environments. The reality looks very different:

all new infrastructure needs to coexist and integrate with a lot of other systems, some of which may be outdated. It is again all about a holistic approach.

Everything is connected and the defence you set up needs to offer protection in depth 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 automatically 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. Canonical makes it easy to confine applications with snaps, applications that are containerised with all their dependencies. This bundling reduces the risk of compatibility issues or unintended changes to the underlying system, with the great number of available snaps ranging from popular everyday apps to security and infrastructure-related ones like Microk8s, MAAS, Juju and Livepatch.

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.



Look at the entire stack as a Jenga tower. The risk of ignoring the bottom layers becomes quickly apparent. If you remove a brick from the top, a few of the blocks might fall. But if you try to remove one from the bottom the chances of the entire tower toppling over are exponentially higher. The answer lies in confinement at the infrastructure, containers and operating system layer to limit the likelihood of issues spreading to several layers of the stack.

At Canonical this principle is always top of mind. When testing for open source security and reliability we make sure that our products not only perform well on their own, but they work even better when deployed alongside each other. As a result, we have a vertically integrated offering that covers everything from the operating system you deploy on bare metal, to container images and all the way to application automation with Juju.

Prevention through secure configuration

The most effective software security strategies centre 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 Linux 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. At Canonical, we believe that security patching and hardening should be within reach and easy to implement for teams of all sizes. That is why we provide Ubuntu Pro access for personal and small-scale commercial use for free for up to five machines (including the hardening features mentioned above; more about our Ubuntu Pro offering below).

The focus on prevention through configuration should go beyond just the operating system. For instance, Canonical Kubernetes and Microk8s implement the Center For Internet Security (CIS) hardening profile by default. We are also working on minimal and hardened container images with the lowest possible surface attack area.

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.

The easiest way to make sure is to work with a vendor that collaborates closely with the upstream. Canonical works with the upstream under embargo, this gives us insight into patches and vulnerabilities ahead of others. It is likely that developers in your organisation are already using the Ubuntu repositories. Ubuntu Pro lets them pull over 30,000 packages from Ubuntu repositories, all secured and tested by Canonical.

Patching made easy

Once security vulnerabilities are identified, there is a big difference in the effort required to patch them depending on the tooling used. It is possible to do it manually. To track 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.

Canonical provides turnkey security patching solutions that work in even the most restrictive environments through unattended upgrades, Snaps, Livepatch and Landscape. Together, they can help reduce the average CVE exposure time from 98 days to just one for the most critical vulnerabilities.

These tools can be configured to update your systems in a staggered manner to prevent any downtime. Livepatch eliminates the need for unplanned maintenance windows for high and critical severity kernel vulnerabilities by patching the Linux kernel while the system runs. Landscape also scales well when security patches have to be applied in an automated manner across many machines.

Another tool that significantly decreases the effort required to patch CVEs is Livepatch, which allows you to apply important CVE fixes outside of the scheduled maintenance window and defer reboots.

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. Your open source security will be ahead of most organisations already.

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.

Canonical helps you deal with threat detection and response through partnerships with leading security vendors. Canonical solutions can also streamline recovery after an incident, with software operators that automate common actions across the most popular open source applications. Over the years, we have become affiliated with vendors of vulnerability management platforms like Tenable Nessus, malware detection systems like Microsoft Defender and infrastructure security tools like Aqua Security.

Our long history of collaboration means they understand the inner workings of our systems (and hence are good at distinguishing normal from unexpected behaviour) and they are aware of all the available patches and vulnerabilities.

How Ubuntu Pro simplifies open source vulnerability management

Ubuntu Pro is an extra layer of services on top of every Ubuntu LTS (Desktop and Server) that ensures the open source you use is maintained, secured and tested, in the cloud and on-premise. The comprehensive subscription for security and compliance includes hardening and patching automation tooling.

In a steady cadence, the Ubuntu releases have followed one another like clockwork since 2004. Just as methodically, we have built our open source security reputation by actively monitoring for vulnerabilities in open source software packages and applying fixes. Each fix is always tested to ensure it does not cause disruptions or introduce new issues into the code elsewhere.

Any Ubuntu LTS already gets five years of security maintenance for the base operating system (and its Main repository). With Ubuntu Pro, we commit to CVE patching any package in not just the Main repository but also the Universe repositories for ten years. Together, the Universe repositories currently contain over 30,000 packages which comprise the most popular open source applications and toolchains.

Critical CVEs in the kernel are automatically patched with Livepatch, without causing any downtime. The patches also include backporting when necessary, taking away the complexity of understanding different codebases and their dependencies yourself.

Since we know every environment is complex, Ubuntu Pro also offers a support tier so you can contact experts on call when things go wrong. We help you bridge the troubleshooting and bug-fixing skill gap in your organisation. This is ideal when something breaks in your environment and you need to respond quickly.

Conclusion

Open source security and vulnerability management can be time-consuming and difficult. Scarce resources, software stack complexity, fragmentation and talent shortages compound the issues many companies face. Embracing a security framework like NIST is an organisational imperative. For diligent vulnerability identification and patching, organisations can also pull software from vetted, secure sources and rely on automation tooling to save time.

Organisations using Ubuntu can offload many of these tedious tasks and get comprehensive security and compliance with Ubuntu Pro.

Ubuntu Pro in a nutshell

1. **Faster time to fix by** reducing the average CVE exposure time from 98 days to just one for the most critical vulnerabilities.

2. **Ten years of stability** for infrastructure, operating system and applications, contained in Ubuntu's Main and Universe repositories.

3. **Ensures uptime and quick remediation** with patching tools like Landscape and Livepatch

4. **Streamlines compliance** certification and audits with frameworks like DISA-STIG, FedRAMP, HIPAA and more

5. **Automation for hardening at scale** with Landscape and Ubuntu Security Guide.

Next steps

- Contact us for a security assessment
- Learn more about Ubuntu Pro
- Get an Ubuntu Pro subscription

References

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

Canonical