Rocket® software

Securing Open-Source Software on the Mainframe

Building a Secure and Resilient Technology Stack with Open-Source Software in a Zero-Trust Environment

Contents

Back to contents

Executive summary

As technology evolves, both companies and federal agencies face growing pressures to build secure, scalable, and cost-efficient technology stacks. While Open-source Software (OSS) offers exceptional speed to market, flexibility, and innovation, security concerns have raised questions about its role in sensitive, zero-trust architectures. This white paper examines how OSS can be a reliable cornerstone for modern technology stacks, addresses common security concerns, and outlines strategies for confident integration of OSS in highly sensitive environments.

Back to contents

The value of open-source software

Open-source software has become a critical driver of innovation and efficiency across industries.
Key benefits include:

01

Cost-effectiveness:

OSS eliminates licensing fees, freeing budget for other priorities such as hiring talent and infrastructure investment.

02

Flexibility and customizability:

Developers can modify OSS to fit unique organizational needs.

03

Community-driven innovation:

Global communities of contributors ensure OSS evolves rapidly and adapts to emerging threats.

04

Transparency:

Source code is open to inspection, allowing organizations to vet software for vulnerabilities.

Despite these advantages, the open nature of OSS often raises concerns in sensitive environments.

Addressing security concerns in OSS

A common misconception is that OSS is inherently less secure than proprietary software. In fact, the transparency of OSS can lead to stronger security when managed correctly. Here's why:

01

Community vigilance:

Large, active OSS communities often identify and patch vulnerabilities faster than proprietary vendors.

02

Third-party audits:

Many OSS projects undergo rigorous external security audits, adding an additional layer of oversight.

03

Zero-day resilience:

Frequent updates and patches minimize the window of exposure to new vulnerabilities.

04

Transparency and trust:

Unlike closed-source software, OSS allows organizations to inspect every line of code for potential backdoors or weaknesses.

While addressing these security concerns is critical, the principles of zero-trust architecture provide an ideal framework for securely integrating OSS, ensuring that no component is trusted implicitly, and every interaction is rigorously verified.

Back to contents

Zero-trust architectures and open-source software

Zero-trust security models assume threats can originate from inside and outside the network, requiring strict access verification at every point. Integrating OSS into a zero-trust framework requires strategic measures to ensure compatibility and security.

Strategies for securing OSS in a zero-trust environment:

01 Supply chain security

- Use tools like Software Bill of Materials (SBOMs) to trace dependencies and verify the integrity of OSS components
- Adopt cryptographic signing of OSS packages to prevent tampering during distribution

02 Rigorous vetting and testing

- Conduct static and dynamic code analysis to identify vulnerabilities in OSS
- Use penetration testing and fuzzing tools to simulate real-world attacks on OSS components. All software/tools regardless of where they came from should be checked against NIST's Vulnerabilities Database

03 Compliance with standards

- Ensure OSS adheres to standards like NIST's Cybersecurity Framework and the Federal Information Security Management Act (FISMA)
- Choose OSS projects that actively implement and support these standards

04 Community and governance assessment

- Evaluate the maturity of OSS projects, including the activity of maintainers and the project's track record for addressing security issues
- Prioritize OSS backed by reputable foundations, such as the Linux Foundation, which often oversee projects with robust governance models

05 Implementation of Zero-Trust Principles

- Micro-segmentation: Limit the scope of OSS access within the network
- Identity Verification: Use multi-factor authentication and strict identity verification for OSS-related tools and systems
- Continuous Monitoring: Leverage observability tools to detect anomalies and enforce runtime security policies

Back to contents

Case study: Open source in federal government

Example 1: The Department of Defence (DoD)

The DoD has embraced OSS through its Code.mil initiative, which emphasizes transparent collaboration to develop secure software solutions. By leveraging OSS, the DoD benefits from rapid innovation while maintaining stringent security protocols.

Code.mil is a broad initiative aimed at building open collaboration between the global developer community and DoD projects through open-source software (OSS). By embracing open-source principles, Code.mil has enabled the DoD to tap into a vast pool of technical talent, fostering innovation and enhancing the efficiency of software development processes critical to national security. This approach promotes innovation, reduces duplication of effort, and enhances security through community-driven development.

Relevance to zero trust and security:

Incorporating OSS through initiatives like Code.mil can align with zero-trust security models by:

- **Ensuring code transparency:** Open access to source code allows for thorough security assessments, ensuring that software components are free from vulnerabilities and backdoors.
- **Facilitating rapid patching:** The collaborative nature of OSS communities enables swift identification and remediation of security issues, reducing exposure to potential threats.
- **Enhancing supply chain security:** By actively participating in OSS projects, organizations can better understand and manage their software supply chains, ensuring that all components meet stringent security standards.

Through Code.mil, the DoD exemplifies how open-source initiatives can be integrated into secure, zero-trust environments, leveraging community collaboration while maintaining rigorous security protocols.

Example 2: The Cybersecurity and Infrastructure Security Agency (CISA)

CISA promotes the use of OSS in its guidelines for securing software supply chains, demonstrating confidence in the ability of OSS to meet rigorous federal security requirements.

Key initiatives and guidelines supporting OSS

1. **Software supply chain security**

   CISA's guidance emphasizes securing the software supply chain, a critical aspect of using OSS. Through its **Software Bill of Materials (SBOM) initiative**, CISA encourages organizations to document all OSS dependencies, ensuring transparency and traceability. SBOMs allow federal agencies to understand the origin, integrity, and potential vulnerabilities of OSS components before incorporating them into sensitive systems.

2. **Binding operational directives (BODs)**

   CISA issues BODs requiring federal agencies to address vulnerabilities in software, including OSS. For example, agencies are mandated to patch vulnerabilities listed in CISA's **Known Exploited Vulnerabilities Catalog**, ensuring that all software — including OSS — is updated and secure.

3. **Zero trust integration**

   CISA's **Zero Trust Maturity Model** highlights the compatibility of OSS with zero-trust principles. By leveraging OSS tools that enforce micro-segmentation, identity verification, and continuous monitoring, federal agencies can build resilient systems aligned with the zero-trust framework.

4. **Community engagement**

   CISA actively collaborates with the open-source community to improve software security. This includes partnerships with organizations like the Open-Source Security Foundation (OpenSSF) to develop secure coding practices, enhance vulnerability management, and establish governance models for OSS projects.

Meeting rigorous federal security needs

1. **Transparency and audits**

   OSS allows federal agencies to inspect source code for security vulnerabilities, backdoors, and compliance with government standards. CISA encourages thorough auditing of OSS components to ensure they meet stringent security requirements.

2. **Rapid patch management**

   The collaborative nature of OSS communities aligns with CISA's directive for timely vulnerability remediation. OSS contributors often release patches faster than proprietary software vendors, reducing exposure to known exploits.

3. **Compliance with federal standards**

   Many OSS projects are designed to comply with federal security frameworks, such as:

   - **NIST Cybersecurity Framework (CSF):** OSS tools like OpenSCAP help automate compliance with NIST standards.
   - **Federal Information Security Management Act (FISMA):** OSS can be integrated into FISMA-compliant systems by adopting secure development practices and supply chain security measures.

4. **Cost-effectiveness and innovation**

   CISA acknowledges the cost-saving benefits of OSS while emphasizing its role in driving innovation. The reduced cost of OSS allows federal agencies to allocate resources toward implementing additional layers of security and adopting cutting-edge technologies.

Examples of CISA-endorsed OSS tools

CISA has highlighted and promoted the use of several OSS tools to meet federal security requirements, such as:

- **Kubernetes:** For container orchestration with robust security configurations.
- **OpenSCAP:** For automating compliance and security audits.
- **OSQuery:** For endpoint monitoring and security.

Strengthening Federal Confidence in OSS

CISA's comprehensive approach to securing OSS aligns with its broader mission to protect federal networks and critical infrastructure. By promoting rigorous vetting, collaboration with OSS communities, and adherence to federal standards, CISA ensures that OSS can be securely integrated into high-stakes environments.

Conclusion

Through its guidelines and initiatives, CISA demonstrates that OSS is not only compatible with rigorous federal security needs but can also enhance the resilience and agility of government technology stacks. With transparency, robust supply chain security, and proactive vulnerability management, OSS emerges as a powerful tool in building secure and innovative solutions for federal agencies.

Back to contents

Confidence in open-source security: Overcoming myths

Addressing common myths about open-source software is essential to dispel misconceptions that might hinder its adoption, especially in environments where security and compliance are paramount.

1. **OSS is risky because it's open**

   Reality: Transparency allows for broader scrutiny, enabling rapid identification and mitigation of vulnerabilities.

2. **OSS lacks support**

   Reality: Reputable OSS projects often have vibrant ecosystems, with vendors and third parties providing enterprise-grade support.

3. **OSS is incompatible with compliance**

   Reality: Many OSS tools are designed to meet or exceed compliance standards. For example, OpenSCAP provides automated compliance checking for government agencies.

Recommendations for Adoption

1. **Create an OSS policy framework:** Define standards for evaluating and approving OSS for use in sensitive systems.
2. **Invest in secure development practices:** Train development teams in secure coding and OSS integration techniques.
3. **Collaborate with the OSS community:** Actively participate in OSS projects to influence their security roadmap.
4. **Adopt continuous improvement:** Regularly review and update OSS components to address evolving threats.

Conclusion

Open-source software, when carefully selected and managed, is not only compatible with but can enhance the security of zero-trust architectures. Federal agencies and companies alike can confidently leverage OSS by implementing robust governance, rigorous security measures, and proactive monitoring. By embracing the power of OSS, organizations can achieve a resilient, cost-effective, and future-proof technology stack.

Learn more about Rocket® Open-Source Solutions

Ask an expert

About Rocket Software

Rocket Software is the global technology leader in modernization and partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,000 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located around the world. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on **LinkedIn** and X (formerly Twitter) or visit **RocketSoftware.com** to learn more.

Modernization. Without Disruption.™

Visit RocketSoftware.com ›

Rocket software

MAR-12557_OpenSourceSecurity_Whitepaper_v7