



Alpha-Omega
2023 Grantee
Outcomes by
Investment



Key Achievements

Eclipse Temurin About to Reach SLSA Build: Level 3 Compliances

Eclipse Temurin, a key Eclipse Foundation project, is nearing a significant achievement in software supply chain security by attaining Level 3 compliance in the SLSA (Supply chain Levels for Software Artefacts) build track. Renowned for its high-performance, cross-platform, open-source licensed OpenJDK distributions that are Java SE TCK-tested, Eclipse Temurin's upcoming milestone underscores its commitment to trustworthiness and provenance in software development. Level 3 compliance, to be achieved with their next release in January, mandates building on a hardened platform with robust tamper protection, significantly reducing the risk of supply chain attacks. Furthermore, Eclipse Temurin emphasizes reproducible builds, ensuring that binaries are consistently generated from the same source code, a fundamental principle for reliable software supply chains.

OtterDog: Securing Repository Management

Eclipse OtterDog, our innovative tool, is transforming the management of GitHub organisations by enabling large-scale management of repository configurations and security policy enforcement through an infrastructure-as-code approach. It facilitates the large-scale management of repository configurations and policy enforcement, allowing configurations to be hosted in separate repositories. Changes are proposed via pull requests and require approval by designated teams. Since its deployment, OtterDog has been adopted by over a third of the Eclipse Foundation's GitHub organisations, leading to heightened engagement and the creation of over 300 community-driven pull requests. These range from minor repository adjustments to implementing branch protection rules, substantially mitigating supply chain attack risks. Additionally, OtterDog enables active monitoring of critical settings like branch protection and secret scanning, promoting the widespread implementation of security policies.



Key Achievements

The Rust for Linux project is seeing its first use cases getting upstreamed along with growing maintainer interest. Rustls is growing as a performant and memory safe TLS library.

Lack of memory safety in widely-used open source software causes great harm to Internet users every time a bug is exploited. Prossimo seeks to change this paradigm by bringing memory safety to critical parts of the Internet like TLS and the Linux kernel. This year, the Rust for Linux effort has expanded and accelerated to include more developers and greater interest from kernel maintainers. The Rustls project reached major milestones in 2023, most notably the release of pluggable cryptographic backends.



Key Achievements

JavaScript technologies are the front door to a global digital economy through every business that uses it for their websites. jQuery, for example, is a fast, small, and feature-rich JavaScript library that makes building company websites easy. A recent IDC study commissioned by the OpenJS Foundation showed more than three-quarters of a billion websites are out of date, with most capturing personal and financial information.

IDC surveyed more than 500 people in 23 industries across North America, UK and Europe, representing small, medium, and large organizations. Over one-third of respondents confirm having experienced a security incident in the last 24 months.

In fact, most businesses surveyed reported that jQuery was not the cause of their security incident. However, jQuery is the canary in the coal mine. While an outdated jQuery itself might not be the cause of those security risks, it's likely a sign that processes to keep open source software patched and safe are missing. Further, the public and private sectors need to work together to get businesses to move off outdated and unsupported open source technologies.

The work is culminating in early 2024 with the release of a Healthy Web Checkup Tool and a broad campaign to partner with key stakeholders and raise awareness in businesses across the globe. By adopting regular healthy habits, similar to medical checkups, the consumer web will be strengthened and continue to be useful in the years ahead.

Announcement: [OpenJS Foundation Warns Consumer Privacy and Security at Risk in Three-Quarters of a Billion Websites](#)



Key Achievements

Node.js is a widely popular, community-led project at the OpenJS Foundation used by companies and organizations like NASA and Netflix. Yet the maintainers are overwhelmed by end users and companies who depend on Node.js and yet contribute very little back. Simply put, many community-led JavaScript projects like Node.js lack the time, people and expertise for security. Project maintainers face additional pressures to respond to vulnerabilities on timelines that are unrealistic for these volunteers.

The Node.js maintainers and OpenJS Foundation received funding from Project Alpha-Omega, an affiliated project of the Open Source Security Foundation (OpenSSF). By providing direct resources to the Node.js project by funding a full

time security engineer, the project reduced the overall risk of the project to triage and manage more security updates, create more automated workflows, and conduct community outreach that has served as a catalyst to build a culture of security around open source. Prior to a funded resource, the project was reactively responding to security issues. Now it has proactive policies in place and is making improvements to a comprehensive set of security measures.

Most recent monthly update on the OpenJS Foundation blog: [Node.js Security Progress Report – Security Release and Node.js 21](#)



Key Achievements

In a span of four months, from August to November 2023, the team has triaged 1,079 open source projects and submitted 168 bug reports (security, reliability and logical bugs). The team has published a publicly available [spreadsheet](#) where the data can be studied.

The following key performance indicators are tracked.

- Number of projects scanned and cleared. In four months, the team has scanned and triaged 1,079 projects. No bugs were reported in 938 projects.
- Number of security and reliability bugs reported. 79 security and reliability bugs have been reported to date. The reported bugs include cross-site scripting, command injection, cross-site request forgery, deserialization issues, weak cryptography issues, data races, null pointer dereferences, etc.
- Number of security and reliability bugs merged. 25 of the 79 security bugs (31%) have already been merged by the maintainers.
- Number of total bugs merged. Among the total bugs submitted, about 45% of the reports have already been accepted by the maintainers. About 5% of the bug reports were challenged; this highlights the accuracy of the results and the triaging process.

The remainder of the bug reports are going through the process of being accepted/merged by the maintainers.

The following contributions are also made by the OpenRefactory team.

- The OpenRefactory team is using a custom-created dashboard (the Triage Portal) to triage the bugs that have been reported by the tools. The dashboard implements a coordinated vulnerability disclosure policy that has been prescribed by the principals at the vulnerability disclosure working group.
- The team members have explored many bug categories in depth to understand how developers use certain language options and how they can lead to bugs. We have collected those thoughts and have created a series of blog posts.
- The OpenRefactory team hosted two high school interns and two undergraduate interns and exposed them to the chores of detecting, triaging, reporting and fixing bugs.



Key Achievements

OSTIF completed more projects this year than ever before by turning funding into direct security improvements through custom security audits. Our work has global impact, helping open source maintainers on projects of all sizes. This is done through OSTIF's ability to connect, direct, and manage security engagements from funding to publication and beyond. The Alpha-Omega project funded a security of OpenSSL in 2023, and OSTIF answered the call, resulting in multiple bug fixes and improvements to the project. Our organization plans to continue not only increasing our rate of output and capacity, but to offer ongoing support to open source projects facing an ever increasing demand for security.

OSTIF has a long history of successfully helping open source projects fulfill a

timeless best practice, independent third party security audits; where security experts help open source projects with their needs and set them up for continued success in improving security posture. In 2023, OSTIF hit a milestone of 50 published security audits for open source projects, with 100 critical/high (according to CVSS score) bugs found and fixed. Furthermore, at an approximate cost of \$6,000 USD per critical/high vulnerability found and fixed, security audits are one of the most cost effective and impactful (high ROI) ways to “turn money into security”. OSTIF’s process strongly encourages transparency, a focus on fixes and improvements, and improved tooling that goes beyond the point in time of the audit.


In 2023, OSTIF worked on multiple collaborations managing a suite of custom audit programs. These programs with funders like Amazon Web Services, the Eclipse Foundation, Google, the Drupal Association, and the Cloud Native Computing Foundation provide not just security work but assistance and specialized labor to open source maintainers. Engagements like these provide impactful short and long term benefits to open source projects, mainly security related but not exclusively. Our forte is being able to source teams that can scope, audit, document, and provide or create tooling that is meaningful and beneficial to all parties involved.

In conclusion, I hope that you see us as a strong and viable option for more funding in 2024. OSTIF has a strong track record for success, and has demonstrated strong leadership in helping open source projects with their security needs. We have the people and expertise to curate successful security outcomes, and have capacity to do more security work for open source projects. Track record: <https://github.com/ostif-org/OSTIF/blob/main/Completed-Engagements.md>



Key Achievements

The Python Software Foundation [hired Seth Larson in June 2023 as the Security](#)



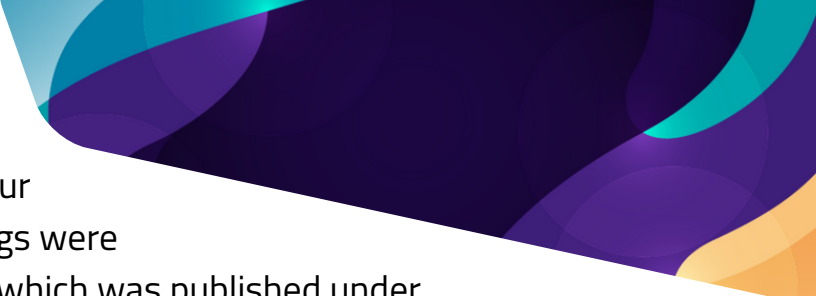
Developer-in-Residence to implement improvements to the supply chain security of the Python ecosystem. Seth has completed systematic improvements to the Python ecosystems' handling of vulnerabilities, including the PSF being authorized as a CVE Numbering Authority. Ongoing improvements are being made to the Python runtime's release process including build environment hardening, build reproducibility, and a proposal to add Software Bill-of-Materials and Vulnerability Exchange documents to releases.

When enhancing security of an open source ecosystem, the primary concern is for the community of people contributing to projects and usually volunteering their time to do so. Keeping these folks top of mind is critical to improving security without compromising the sustainability of the ecosystem. This thinking leads to designing well-documented systems that minimize manual effort and involving maintainers throughout the development process to avoid ending up with an unsustainable solution.

Seth started his focus on the Python Security Response Team (PSRT) which is tasked with triaging and remediating vulnerabilities in the core Python language as well as pip, Python's package manager. Seth was added to the PSRT and quickly became the coordinator for vulnerability disclosures and remediations meaning volunteers only needed to spend time developing remediations instead of on coordination work.

Seth took on the work to authorize the Python Software Foundation as a CVE Numbering Authority (CNA) in order to manage its own CVEs and advisories. This effort included revitalizing the security advisory mailing list and creating an advisory database with new and historical vulnerabilities affecting the Python runtime using the Open Source Vulnerability format. This work meant that Python would always publish advisories containing remediation information and insights from core developers and that vulnerability scanners would be able to consistently detect vulnerabilities in the Python runtime using a standardized format.

The PSF becoming a CNA required learning how the CNA program applied to open



source foundations and projects. During our learning journey all of the work and findings were documented and aggregated into a guide which was published under the OpenSSF Vulnerability Disclosures Working Group. The goal for this guide is to give other open source projects the confidence to decide whether to become CNAs themselves by knowing the time and resource investments upfront without needing to do extensive research themselves.

The Python runtime release process is complex with many moving parts and people involved. Any successful attack against the Python release process would impact large numbers of community members as has happened to other open source projects. Towards mitigating these risks, Seth began by documenting the entire release process for the Python runtime and then compared the process against known sources of supply chain risk and added mitigations for those risks. Mitigations that have been completed so far include making source artifacts completely reproducible byte-for-byte to help detect injected code and auditing the existing Sigstore signatures to fix discrepancies so they can be used by automated tooling to ensure artifacts aren't tampered with post-release.

Proof-of-concept mitigations have been created for moving builds to hardened SLSA-compliant build platforms like GitHub Actions and adding Software Bill-of-Materials to track dependencies being used during the build process to signal to consumers of Python when subcomponents are affected by vulnerabilities. We're hoping to implement these proposed improvements in early 2024 in collaboration with Python core developers.

Much of the work being done in this role is replicable, and documenting the work being done in a public and consumable way means that other software ecosystems can use it as a reference for enhancing their own ecosystems' security. Seth publishes a weekly report to his blog about where the work is happening and linking to discussions and news. This regular publication has resulted in coverage from Python newsletters and podcasts as well as general technology publications like The New Stack and The Register.



Key Achievements


As part of the Rust Foundation Security Initiative in 2023, the Rust Foundation hired several security-focused engineers who now routinely work with members of Rust Project security teams and working groups to surface and address top Rust security needs. Our Security Initiative leaders began analyzing existing Rust ecosystem security threats, delivered several important threat models and audits, built and shared two new open source Rust security projects (Painter and Typomania), made many improvements to the security of crates.io, and much more.

Since the first stable release of the Rust programming language in 2015, the Rust ecosystem has grown tremendously, thanks to the tireless work of its maintainers, contributors, and advocates. Every day, Rust's many advantages are becoming more evident to developers and organizations alike.

But like any programming language, the meteoric rise of Rust also introduces complexities and risks. When the user base of any programming language grows, it becomes more attractive to malicious actors. As any programming language ecosystem expands with more libraries, packages, and frameworks, the surface area for attacks increases. Rust is no different.

As the steward of the Rust programming language, the Rust Foundation has a responsibility to provide a range of resources to the growing Rust community. This responsibility means we must empower contributors to participate in the Rust Project in a secure and scalable manner, eliminate security burdens for Rust maintainers, and educate the public about security within the Rust ecosystem.

In September 2022, the Rust Foundation [announced](#) its commitment to fulfilling



these responsibilities through the Security Initiative, which we were able to create thanks to a generous initial investment of \$460k from Alpha-Omega.

In 2023, Alpha-Omega's support of the Security Initiative enabled us to hire a team of full-time security engineering professionals. Over the past year, the newly expanded Rust Foundation team oversaw many impressive security advancements within the Rust language ecosystem, including...

- The completion and release of Rust Infrastructure and Crates Ecosystem threat models
- The Foundation's first technical open source projects, Painter - an open source tool that creates a complete call graph across the entire crates ecosystem - and Typomania - A tool to check for typosquatting in package registries.
- A variety of crates.io security improvements, including the creation of a crates.io admin console, scoped API tokens, and technical debt reduction.
- New tools and best practices to identify and address malicious crates
- Reduced technical debt within the Rust Project, producing/contributing to security-focused documentation, and elevating security priorities for discussion within the Rust Project.
- New opportunities for Security Initiative-Rust Project collaboration and leadership.