OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Secure Open Source Software Vision Brief 2023

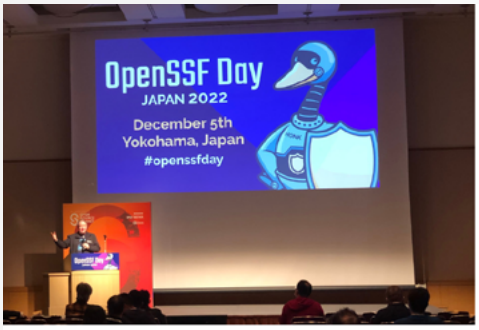
The Open Source Security Foundation (OpenSSF), an initiative of the Linux Foundation, was formed in 2020 as a cross-industry forum to improve open source software (OSS) security collaboratively. In December 2021, the Log4Shell vulnerability in the OSS component log4j was publicly revealed (see the CSRB report on Log4j). While many vulnerabilities have been found in closed source software, Log4Shell made many organizations realize how dependent they have also become on OSS. This led to a global discussion about how to improve OSS security.

The White House issued a call to action and asked organizations to investigate how to improve the security around OSS packages and in particular supply chain security threats. The OpenSSF drafted a Mobilization Plan in early 2022 to identify potential starting points. Turning these ideas into action has faced headwinds given socio-economic realities facing the open source community, which often includes maintainers and contributors working outside their day-to-day to create OSS for the greater public good.

Throughout 2022-2023, despite broader challenges in the technology ecosystem, OpenSSF has achieved significant momentum toward improving OSS security (e.g., building off the activities outlined in our Annual Report December 2022). Our accomplishments in 2022 through the summer of 2023 include:

1. **Software Security Education**. As of August 2023, 20,019 developers enrolled in our courses on the fundamentals of developing secure software (LFD121 and LFD104x). The diversity, equity, and inclusion (DEI) group is supporting an "office hours" program (via videoconference) to mentor newcomers to OSS security.

2. **Security Guides**. Various guides were released or updated for developers, consumers, and the security community to help improve security:
   - » Concise Guide for Developing More Secure Software
   - » Concise Guide for Evaluating Open Source Software
   - » Guide to implementing a coordinated vulnerability disclosure process for open source projects
   - » Guidance for Security Researchers to Coordinate Vulnerability Disclosures with Open Source Software Projects
   - » npm Best Practices Guide



3. **OSS Security Evaluation**. Simplified obtaining security information about OSS so, consumers and maintainers can more efficiently assess OSS security:

» **OpenSSF Scorecard**. Automatically assesses OSS projects against various software security criteria. A score is produced that helps OSS consumers estimate the security of the OSS and helps OSS maintainers by giving them a goal to improve their score. OpenSSF is continually improving the Scorecard, and we now run a weekly Scorecard scan of over one million OSS projects. Allstar is a complementary effort that helps streamline using the Scorecard within a developing organization or project. We've recently added support for GitLab (in addition to GitHub) thanks to contributions from Lockheed Martin.

» **OpenSSF Best Practices Badge.** Security and sustainment criteria that OSS projects can use to more deeply evaluate their efforts and also help OSS consumers understand their status. For example, requiring at least one developer to know how to design secure software and counter common vulnerabilities, and as significant new functionality is added, the developer must add tests to an automated test suite. We have over 6,000 participating projects.

» **Supply-chain Levels for Software Artifacts (SLSA)**. SLSA is a framework to prevent tampering, improve integrity, and secure packages and infrastructure. SLSA version 1.0 was released in April 2023, focusing on protecting build processes. SLSA was subsequently adopted by npm for package integrity.

» **Security reviews collection**. We have collected a set of known security evaluations of OSS, enabling others to find and review this information quickly.

4. **Improved OSS infrastructure & Tooling**. Improving infrastructure and tooling has a broad impact on improving security.

» **Sigstore**. Digital signatures enable receivers to verify that the software package they received was intended by the sender and who the sender was. Digital signatures counter many attacks that involve tricking users into installing malicious packages provided by an adversary. Unfortunately, past digital signature services have either been impractical for OSS or too hard to use. We've released a free service called "sigstore" that provides digital signing and verification for artifacts. Sigstore General Availability (GA) began in October 2022. Sigstore is used for signing releases of CPython, Kubernetes Artifacts, and as part of npm package provenance. We also have a free course on how to use sigstore. Over 32 million entries have been recorded for signatures within sigstore's public signature transparency log, spanning over 17,000 unique OSS projects including Kubernetes, CPython, LLVM, KNative, Istio, and ArgoCD.

» **Secured Software Repositories**. Many OSS packages are acquired through repositories, so we're working with repositories to improve their security. We first conducted a Package Manager Security Landscape Survey to contrast their security capabilities. We then developed Build Provenance for All Package Registries guidance to encourage improvements across all repositories. Some repositories implemented new best practices with support from other repositories. For example, the Python Package Index (PyPI) implemented a new 2FA mandate based on the experience of JavaScript's npm. Some projects are adopting OpenSSF capabilities to improve security like npm which has added an easy mechanism to generate signed provenance attestations (building on Sigstore and SLSA) and publish these results in npm to counter tampering with a package's build process.

» **Better Tooling**. OpenSSF funded spdx/tools-python to improve SBOM handling. We also released Fuzz Introspector to improve fuzz testing and fuzz testing tools to enable detecting vulnerabilities before attackers find them.

5. **Vulnerability finding and reporting**.

» **Alpha-Omega**. A significant effort, with $12.5M in corporate sponsorship, to partner with OSS maintainers to systematically find and fix undiscovered vulnerabilities and improve their overall processes. Current partners include the Python Software Foundation — including funding a Python security developer-in-residence, the OpenJS Foundation and jQuery, the Eclipse Foundation, Node.js, and the Rust Foundation. All of these partners manage important, widely-used OSS, so the security improvements we implement together will substantially improve security for all. The 2022 Alpha-Omega report summarizes recent accomplishments.

» **Security Audits**. We've supported in-depth security audits of some widely-used OSS, often via Alpha-Omega, including Eclipse Equinox P2 (a widely used provisioning platform), sigstore, Jackson-Core and Jackson-Databind, slf4j (a logging framework), and Symfony (a widely used PHP framework).

» **Open Source Vulnerability (OSV) Schema**. OSV is a machine-readable format that precisely maps vulnerabilities to open source package versions or commit hashes. OSV enables rapid automatic identification of vulnerable components so projects and organizations can update those components. More ecosystems are using OSV today, including the relatively recent additions of AlmaLinux, Rocky Linux, and the Haskell programming language (a total of 18 ecosystems today).

6. **Research**. OpenSSF has overseen research such as the Census II of Free and Open Source Software — Application Libraries to identify the most widely-used OSS application libraries and a survey on Addressing Cybersecurity Challenges in Open Source Software. We have also identified a set of critical OSS and are continuing to refine this set.

7. **Community Building & Outreach**. We have held formal OpenSSF Days in North America, Europe, and Japan, as well as local meetups in the global OSS community to encourage the use of OpenSSF resources. Numerous OpenSSF initiatives have been featured in industry conferences (e.g., Scorecard featured at DEFCON 31, 2023) highlighting the benefits of many of our projects and how open source projects can integrate our materials. OpenSSF members are working with the US Cybersecurity and Infrastructure Security Agency (CISA) and the Joint Cyber Defense Collaborative (JCDC) to develop the executive leadership fact sheet ("Improving Security of Open Source Software in Operational Technology and Industrial Control Systems"). This sheet promotes an understanding of OSS and its implementation in operational technology (OT) and industrial control systems (ICS) environments, including best practices for securing OSS in OT.

OpenSSF Plans and Potential Partnership

We believe there are many opportunities to collaborate, both on work we intend to do and on possible new work. The US National Cybersecurity Implementation Plan of July 2023 (NCIP) notes explicitly the need to "scale public-private partnerships" (NCIP 1.2). It includes items such as "promote open-source software security and the adoption of memory safe programming languages" (NCIP 4.1.2). It also notes the need to "use federal grants and other incentives to build in security" (NCIP 3.4.2). Collaboration could help the US government to meet or exceed its completion dates, avoid duplication of efforts (eliminating waste and strengthening results by pooling resources), and enable both the US Government and industry to be seen as partners working together to solve critical issues.

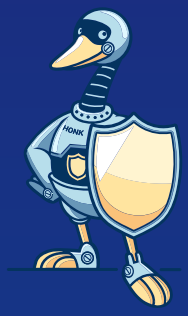
The OpenSSF intends to do the following (here, we note alignments with the NCIP):

1. **OSS Security Education**. We are developing more materials for hands-on and in-depth learning, building on materials listed above and the security knowledge framework. We also intend to release a course for managers on developing secure software. We have other education proposals that still need to be funded but are open for review and comment. Our education work aligns with "publish a US National Cyber Workforce and Education Strategy (NCWES) and track its implementation" (NCIP 4.6.1) and the NCWES itself.
2. **Security Guides**. We're developing more guides, e.g., Compiler Options Hardening Guide for C and C++ and Source Code Management (SCM) Platform Configuration Best Practices. This aligns with "increase agency use of frameworks and international standards…" which notes the need to use guidance (NCIP 1.1.3) and "scale public-private partnerships to drive development and adoption of secure-by-design and secure-by default technology," which notes the need to partner with the "open-source software community" (NCIP 1.2).

3. **Improved OSS Security Evaluation**. These should help "drive development and adoption of secure-by-design and secure-by-default technology" (NCIP 1.2).
   a. **Supply Chain Integrity**. We intend to evolve our existing frameworks SLSA and S2C2F to cover a greater range of functional concerns, enabling open source consumers to more thoroughly assess upstream threats and better manage supply chain risks across their dependency portfolio.
   b. **Scorecard**. We plan to improve the detection of security tools and processes.
   c. **Dashboard**. We intend to develop a "dashboard" (building on our existing prototype and tools) to provide a combined view of security-related information about OSS for maintainers, potential users of OSS, and organizations.
4. **Improved OSS Infrastructure and Tooling**.
   a. **Improved integrated tooling**. We're working on assembling a secure workbench of capabilities to automatically list, scan, remediate, and secure the OSS components flowing through the software supply chain that come together as software is written, built, deployed, consumed, and maintained. This includes a desire to automatically generate software bill of materials (SBOMs) for OSS. This aligns with the "advance software bill of materials (SBOM) and mitigates the risk of unsupported software" (NCIP 3.3.2).
   b. **Increase memory safety**. Our Memory Safety group will encourage critical OSS projects to move to memory-safe-by-default languages. Where it's not possible/ practical, we'll encourage projects to reduce memory safety vulnerabilities. This aligns with "promote open-source software security and the adoption of memory-safe programming languages" (NCIP 4.1.2) and "accelerate the maturity, adoption, and security of memory-safe programming languages" (NCIP 4.2.1).
   c. **Secured Software Repositories**. We have developed a list of areas to work to improve security for Homebrew (MacOS/*nix package manager) and are working on a draft for RubyGems, based in part on previous work for Python. We intend to develop more areas and then work to request and search for funding.
4. **Vulnerability finding and reporting**. These align with "coordinated vulnerability disclosure" (NCIP 3.3.3).
   a. **Alpha-Omega**. We intend to scan many more OSS components and develop automated fixes for them at scale. This aligns with "promote open-source software security…" (NCIP 4.1.2).
   b. **Open Source Software Security Incident Response Team (OSS-SIRT)**. OSS-SIRT is a proposed process and coordinated cross-industry expert group that will be available to help OSS maintainers remediate high-impact security vulnerabilities and related security emergencies. This also aligns with "support private sector and state, local, Tribal, and territorial (SLTT) efforts to mitigate ransomware risk" (NCIP 2.5.4).
   c. **OpenWall mailing lists**. We plan to improve the OpenWall mailing list infrastructure, widely used for OSS vulnerability coordination. This also aligns with "support private sector and state, local, Tribal, and territorial (SLTT) efforts to mitigate ransomware risk" (NCIP 2.5.4).
   d. **OpenVEX.** OpenVEX is a specification for vulnerability exchange (VEX) data. It implements the Cybersecurity and Infrastructure Security Agency (CISA) Minimum Elements for VEX. We expect to update its specification and its vexctl tooling. We are working on guidance for security scanner vendors to effectively ingest VEX documents leveraging OpenVEX and any other VEX statements.
   e. **Vulnerability Disclosures AutoFix**. This special interest group (SIG) within the Vulnerability Disclosures Working Group is developing a

framework and methodology to automate reporting discovered vulnerabilities to upstream open source projects. Alpha-Omega will leverage this framework and it is available to any security researcher to help provide best practices and methods to engage with open source projects effectively.

6. **Artificial Intelligence/Machine Learning (AI/ML) Security.** OpenSSF will collaborate closely with DARPA's Artificial Intelligence Cyber Challenge (AIxCC). AIxCC is a competition in AI and cybersecurity to defend software, and OpenSSF will guide teams in creating AI systems capable of addressing vital cybersecurity issues. More broadly, OpenSSF is developing an approach to collaborate with the LF AI & Data foundation to establish guidance on AI/ML, so AI/ML will improve (instead of reduce) OSS security. The NCIP doesn't mention AI/ML, but the US White House "Blueprint for an AI Bill of Rights" requirement for demonstrating "safety and effectiveness" implies the need for security.
7. **Research**. We plan to update Census II ("Census III") and are considering other research. This aligns with "prioritize funding for cybersecurity research" (NCIP 3.4.2).
8. **Community building and outreach**. OpenSSF will continue this work. This aligns with "strengthen international partners' cyber capacity" (NCIP 5.2.1).



OpenSSF

OPEN SOURCE SECURITY FOUNDATION

openssf.org