May 9th, 2022

The Honorable Eddie Bernice Johnson, Chairwoman
The Honorable Frank Lucas, Ranking Member
Committee on Science, Space, and Technology
2321 Rayburn House Office Building
Washington, DC 20515-6301

Dear Chairwoman Johnson, Congressman Lucas, and distinguished members of the Committee on Science, Space and Technology,

Thank you for your invitation to address you today, and the opportunity to share with you the work being done within the Open Source Security Foundation and the broader open source software community to raise the level of security and trustworthiness of open source software.

1. What are the consequences of insecure open-source software and what is industry as a whole, and the Open Source Security Foundation in particular, doing to tackle such Vulnerabilities?

Open source software ("OSS") has become an integral part of the technology landscape, as inseparable from the digital machinery of modern society as bridges and highways are from the physical equivalent. According to one report, typically 70% to 90% of a modern application "stack" consists of pre-existing OSS, from the operating system to the cloud container to the cryptography and networking functions, sometimes up to the very application running your enterprise or website. Thanks to copyright licenses that encourage no-charge re-use, remixing, and redistribution, OSS encourages even the most dogged of competitors to work together to address common challenges, saving money by avoiding duplication of effort, moving faster to innovate upon new ideas and adopt emerging standards.

However, this ubiquity and flexibility can come at a price. While OSS generally has an excellent reputation for security, the developer communities behind those works can vary significantly in their application of development practices and techniques that can reduce the risk of a defect in the code, or in responding quickly and safely when one is discovered by others. Often, developers trying to decide what OSS to use have difficulty determining which ones are more likely to be secure than others based on objective criteria. Enterprises often don't have a well-managed inventory of the software assets they use, with enough granular detail, to know when or if they're vulnerable to known defects, and when or how to upgrade. Even those enterprises who may be willing to invest in increasing the security of the OSS they use often don't know where to make those investments, nor their urgency relative to other priorities.

There are commercial solutions to some of these problems. There are vendors like Gitlab or Red Hat who sell support services for specific open source software, or even entire aggregate distributions of OSS. There are other vendors, like Snyk and Sonatype, who sell tools to help enterprises track their use of OSS and flash an alert when there is a new critical vulnerability in software running deep inside an enterprise's IT infrastructure.

However, fighting security issues at their upstream source - trying to catch them earlier in the development process, or even reduce the chances of their occurrence at all - remains a critical need. We are also seeing new kinds of attacks that focus less on vulnerabilities in code, and more on the supply chain itself - from rogue software that uses “typosquatting” on package names to insert itself unexpectedly into a developer’s dependency tree, to attacks on software build and distribution services, to developers turning their one-person projects into “protest-ware” with likely unintended consequences.

To address the urgent need for better security practices, tools, and techniques in the open source software ecosystem, a collection of organizations with deep investments into the OSS ecosystem came together in 2020 to form the Open Source Security Foundation, and chose to house that effort at the Linux Foundation. This public effort has grown to hundreds of active participants across dozens of different public initiatives housed under 7 working groups, with funding and partnership from over 75 different organizations, and reaching millions of OSS developers.

The OpenSSF’s seven working groups are:

1. **Best Practices for Open Source Developers**: This group works to provide open source developers with best practices recommendations, and easy ways to learn and apply them. Among other things, this group has developed courseware for teaching developers the fundamentals of secure software development, and implement the OpenSSF Best Practices Badge program.
2. **Securing Critical Projects**: This group exists to identify and help to allocate resources to secure the critical open source projects we all depend on. Among other things, this has led to a collaboration with Harvard Business School to develop a list of the most critical projects.
3. **Supply Chain Integrity**: This group is helping people understand and make decisions on the provenance of the code they maintain, produce and use. Among other things, this group has developed a specification and software called “SLSA”, for describing and tracking levels of confidence in a software supply chain.
4. **Securing Software Repositories**: This group provides a collaborative environment for aligning on the introduction of new tools and technologies to strengthen and secure software repositories, which are key points of leverage for security practices and the promotion to developers of more trustworthy software.
5. **Identifying Security Threats in Open Source Projects**: This group enables informed confidence in the security of OSS by collecting, curating, and communicating relevant metrics and metadata. For example, it is developing a database of all known security reviews of OSS.
6. **Security Tooling**: This group’s mission is to provide the best security tools for open source developers and make them universally accessible. Among other activities, this group has released code to better enable a security testing technique called “fuzzing” among open source projects.
7. **Vulnerability Disclosures**: This group is improving the overall security of the OSS ecosystem by helping advance vulnerability reporting and communication. For example, this group has produced a Guide to Coordinated Vulnerability Disclosure for OSS.

There are also a series of special projects under the OpenSSF worthy of special mention:

- **Project sigstore**: an easy-to-use toolkit and service for signing software artifacts, ensuring that the software you are holding is the same as what the developer intended, addressing a wide array of supply chain attacks.
- **The Alpha-Omega Project**: an effort to systematically search for new vulnerabilities in open source code, and work with critical open source projects to improve their vulnerability handling and other security practices.
- **The GNU Toolchain Initiative**: this effort supports the build ecosystems for perhaps the most critical set of developer libraries and compilers in the world, the GNU Toolchain, as a means to ensure its safety and integrity.

All the above efforts are public-facing and developed using the best practices of open source software communities. Funding from our corporate partners goes towards supporting the core staff and functions that enable this community, but all the substance comes from voluntary efforts. In some cases funds flow to assist with specific efforts - for example, recently the Alpha-Omega project decided to allocate funding towards the NodeJS community to augment its security team with a part-time paid employee and to fund fixes for security issues.

The Linux Foundation has also begun to adapt its “LFX” platform, a set of services designed to support the open source communities hosted by the Foundation, to incorporate security-related data such as vulnerability scans from Snyk and BluBracket, along with information from the OpenSSF Best Practices Badge program and the OpenSSF Security Scorecards initiative, to provide a unified view of the security risks in a particular collection of open source code, and what maintainers and contributors to those projects can do to improve those scores and reduce those risks. We expect to see more kinds of risk-related data coming into a unified view like this, helping developers and enterprises make better decisions about what open source components and frameworks to use, and how to reduce risk for those components they depend upon.

Guiding all of this is a deep conviction among the OpenSSF community that while there are many different ways in which security issues manifest themselves in the OSS ecosystem, every one of them is addressable, and that there are lots of opportunities for investment and collective action that will pay a return many times over in the form of lower risk of a future major vulnerability in a widely-used package, and lesser disruption if one is discovered.

Other efforts at the Linux Foundation include “Prossimo”, an effort focused on moving core Internet-related services to “memory-safe” languages like Rust, Go, or Java, which would eliminate an entire category of vulnerabilities that other languages allow too easily. Another is the SPDX standard for Software Bill of Materials (“SBOMs”), addressing the needs identified by White House Executive Order 14028 in a vendor-neutral and open way.

This is by no means a comprehensive list of all such efforts in the OSS ecosystem to improve security. Every OSS foundation either has a security team in operation today or is scrambling to identify volunteers and funding to establish one. There is a greater emphasis today than I’ve seen in my 30 years of using and contributing to OSS (since before it was called OSS) on the importance of such efforts. Clear metrics for progress are elusive since we lack clear metrics for evaluating software risk; in fact developing ways to measure and represent that risk is a key priority for OpenSSF. We will never see a time when open source software is free from security defects, but we are getting better at determining the tools and techniques required to more comprehensively address the risk of vulnerabilities in open source code. Scaling up those tools and techniques to address the tens of thousands of widely used OSS components and to get them more quickly updated remains a challenge.

2. How can the Federal government improve collaboration with industry to help secure open-source software?

I'll focus here on principles and methods for collaboration that will lead to more secure OSS, and then for question 3 on specific opportunities to collaborate on.

First, focus on resourcing long-term personal engagements with open source projects.

Over the last few years, we have seen a healthy degree of engagement by the Federal government with OSS projects and stakeholders on the topic of improving security. The push established by Executive Order 14028 for the adoption of SBOMs aligned nicely with the standardization and growing adoption of the SPDX standard by a number of OSS projects, but it was aided substantially by the involvement of personnel from NIST, CISA, and other agencies engaging directly with SPDX community members.

Often the real secret to a successful OSS effort is in the communities of different stakeholders that come together to create it - the software or specification is often just a useful byproduct. The Federal government, both through its massive use of open source code and the role that it traditionally performs in delivering and protecting critical infrastructure, should consider itself a stakeholder, and like other stakeholders prioritize engagement with upstream open source projects of all sizes. That engagement need not be so formal; most contributors to open source projects have no formal agreement covering that work aside from a grant of intellectual property in those contributions. But as they say, "history is made by those who show up." If the IT staff of a Federal agency (or of a contractor under a Federal contract) were authorized and directed to contribute to the security team of a critical open source project, or to addressing known or potential security issues in important code, or to participating in an OpenSSF working group or project, that would almost certainly lead to identifying and prioritizing work that would result in enhanced security in the Federal government's own use of open source code, and likely to upstream improvements that make OSS more secure for everyone else.

Second, engage in OSS development and security work as a form of global capacity building, and in doing so, in global stability and resilience. OSS development is inherently international and has been since its earliest days. Our adversaries and global competitors use the same OSS that we do, by and large. When our operating systems, cloud containers, networking stacks and applications are made to be more secure, there are fewer chances for rogue actors to cause disruption, and that can make it harder to de-escalate tensions or protect the safety of innocent parties. Government agencies in France, Taiwan, and more have begun to establish funded offices focused on the adoption, development, and promotion of OSS, in many ways echoing the Open Source

Program Offices being set up by companies like Home Depot and Walmart or intergovernmental agencies like the WHO. The State Department in recent years has funded the development of software like Tor to support the security needs of human rights workers and global activists. The Federal government could use its convening authority and statecraft to bring like-minded activities and investment together in a coordinated way more effectively than any of us in the private sector can.

Third, many of the ideas for improving the security of OSS involve establishing services - services for issuing keys to developers like Project sigstore does, or services for addressing the naming of software packages for SBOMs, or services for collecting security reviews, or providing a comprehensive view of the risk of open source packages. Wherever possible, the Federal government should avoid establishing such services themselves when suitable instances of such services are being built by the OSS community. Instead of owning or operating such services directly, the Federal Government should provide grants or other resources to operators of such services as any major stakeholder would. Along similar lines, should the Federal government fund activities like third party audits of an open source project, or fund fixes or improvements, it should ensure not only that such efforts don't duplicate work already being done, it should ensure that the results of that work are shared (with a minimum of delay) publicly and upstream so that everyone can benefit from that investment.

These three approaches to collaboration would have an outsized impact on any of the specific efforts that the Federal government could undertake.

3. Where should Congress or the Administration focus efforts to best support and secure the open-sourced software ecosystem as a whole?

The private sector and the Federal government have a common cause in seeing broad improvements in the security of OSS. I'm happy to share where I see the private sector starting to invest in enhanced OSS security, in the hopes that this may inspire similar actions from others.

1. **Education**. Very few software developers ever receive a structured education in security fundamentals, and often must learn the hard way about how their work can be attacked. The OpenSSF's Secure Software Fundamentals courses are well regarded and themselves licensed as open source software, which means educational institutions of all kinds could deliver the content. Enterprises could also start to require it of their own developers, especially those who touch or contribute to OSS. There must be other techniques for getting this content into more hands and certifications against it into more processes.
2. **Metrics and benchmarks**. There are plenty of efforts to determine what are suitably objective metrics for characterizing the risks of OSS packages. But running the cloud systems to perform that measurement across the top 100,000 or even 10,000 open source projects may cost more than what can be provided for free by a single company, or may be fragile if only provided by a single vendor. Collective efforts funded by major stakeholders are being planned-for now, and governments as a partner to that would not be turned away.
3. **Digital signatures.** There is a long history of U.S. Government standards for identity proofing, public key management, signature verification, and so on. These standards are very sophisticated, but in open

source circles, often simplicity and support are more important. This is pulling the open source ecosystem towards Project sigstore for the signing of software artifacts. We would encourage organizations of all sorts to look at sigstore and consider it for their OSS needs, even if it may not be suitable for all identity use cases.

4. **Research and development investments into memory-safe languages**. As detailed above, there are opportunities to eliminate whole categories of defects for critical infrastructure software by investing in alternatives written in memory-safe languages. This work is being done, but grants and investments can help accelerate that work.
5. **Fund third-party code reviews for top open source projects.** Most OSS projects, even the most critical ones, never receive the benefit of a formal review by a team of security experts trained to review code not only for small bugs that may lead to big compromises, but to look at architectural issues and even issues with the features offered by the software in the search for problems. Such audits vary tremendously in cost based on the complexity of the code, but an average for an average-sized code base would be $150K-250K. Covering the top 100 OSS projects with a review every other year, or even 200 every year, seems like a small price compared to the costs on US businesses to remedy or clean up after a breach caused by just one bug.
6. **Invest into better supply chain security support in key build systems, package managers, and distribution sites**. This is partly about seeing technologies like SBOMs, digital signatures, specifications like SLSA and others built into the most widely used dev tools so that they can be adopted and meaningfully used with a minimum of fuss. Any enterprise (including the Federal government) that has software certification processes based on the security attributes of software should consider how those tools could be enhanced with the above technologies, and automate many processes so that updates can be more frequent without sacrificing security.

These activities, if done at sufficient scale, could dramatically lower the risks of future disruptive events like we have seen. As a portfolio of different investments and activities they are mutually reinforcing, and none of them in isolation is likely to have much of a positive impact. Further econometrics research could help quantify the specific reduction of risk from each activity. But I believe that each represents a very cost-effective target for enhancing security in OSS no matter who is writing the check.

Thank you again for the opportunity to share these thoughts with you. I look forward to answering any questions you may have or providing you with further information.

Sincerely,

[illegible]

Brian Behlendorf
General Manager, Open Source Security Foundation
The Linux Foundation