Office of the National Cyber Director, Executive Office of the President, Cybersecurity and Infrastructure Security Agency, DHS, National Science Foundation, Defense Advanced Research Projects Agency, and Office of Management and Budget, Executive Office of the President

Comments of the Messaging Malware Mobile Anti-Abuse Working Group (M^3AAWG) on Request for Information on Open-Source Software Security: Areas of Long-Term Focus and Prioritization

Consultation reference: Request for Information on Open-Source Software Security: Areas of Long-Term Focus and Prioritization
ONCD-2023-0002
RIN 0301-AA01

Introduction

The Messaging Malware Mobile Anti-Abuse Working Group (M^3AAWG) appreciates the opportunity to submit comments in response to the above-referenced consultation. M^3AAWG is a technology-neutral global industry association. As a working body, we focus on operational issues of internet abuse including technology, industry collaboration, and public policy. With more than 200 institutional members worldwide, we bring together stakeholders in the online community in a confidential yet open forum, developing best practices and cooperative approaches for fighting online abuse.

Context

Eighty to ninety percent (80%–90%) of all current software solutions include at least some open-source elements within their builds.[1] While both closed-source and open-source come with their own specific risks, open-source implementations of security-relevant code come with – or have the potential to realize – various important benefits. For example, open-source code is auditable, with many eyes being able to check the code. While this is not always done, or not always sufficiently done, the possibility is important. Free or Open-Source Software (F/OSS) therefore has the potential to significantly increase digital trust, and can help inform the creation and maintenance of best practices as well as industry standards.

However, F/OSS also comes with pertinent risks and is a key area of concern for security professionals. In the RFI, M^3AAWG has directly applicable expertise, experience, and suggestions for two key areas identified as considerations:

[1] Fox,Fox, Brian. "The EU's Product Liability Directive could kill open source,." *TechRadar*, 10 July 2023, https://www.techradar.com/pro/the-eus-product-liability-directive-could-kill-open-source. Accessed 30 August 2023.

- *RFI Area:* Secure Open-Source Software Foundations
    - *RFI Sub-Area*: Strengthening the software supply chain
        - Detection and mitigation of vulnerable and malicious software development operations and behaviors
- *RFI Area*: Behavioral and Economic Incentives to Secure the Open-Source Software Ecosystem

M[3]AAWG has been working within our collaborative international community to monitor and address technical concerns with malicious actors, their abuse of technology, their methods, and their footprints, as well as working to stop their disruptions while enabling the legitimate use of internet technologies. Although malicious actors engaging in open-source attacks are not necessarily a new threat, the types of attacks and funding for long-term engagements have increased, and thus constitute a growing threat that the ecosystem needs help to address.

Threats and Risks

Based on our collaborative anti-abuse work, we would like to highlight the following threat and risk scenarios or areas. We note that these scenarios do not happen in a vacuum; they often overlap.

Malicious Actors Within the Open-Source Software Supply Chain

- The volume and frequency as well as the unsteady nature of contributions can hide malicious updates (as well as genuine mistakes). This risk is particularly pronounced when maintainers are overwhelmed or inexperienced.
- The maintainer's expertise in development or review for security or privacy-preserving code is not always given. Due to the nature of open-source work, it is possible for projects to gain traction and publicity fairly quickly, overwhelming resource-limited maintainers.
- The selling/sharing/reuse/compromise of developer credentials can lead maintainers or the community to arbitrarily trust an actor whose prior submissions were substantially supported, but whose new submissions may contain malicious insertion(s).

Economics and Incentives in Open-Source Ecosystems

- The majority of open-source software is developed by volunteers who do not receive remuneration, and who cannot be centrally overseen or governed.
- Therefore, many projects are not resourced sufficiently to provide reasonable assurance for code security, whether in terms of developer time, skill, or monetary resources.
- The volunteer status of maintainers and contributors and the potential for liability risk associated with contributions could stifle the innovative process around open-source software.
- Eric Raymond's famous 1999 slogan – "given enough eyeballs, all bugs are shallow" – may be less true today than it was back then. Incentives for contribution have dwindled, and developers adept in security or privacy-preserving efforts are stretched relatively thin. Not every project receives the attention and work necessary to make it secure and safe. Opening the code for review alone does not guarantee security. Being able to read the code is a necessary condition, but availability is not sufficient to guarantee reviews – especially reviews by qualified individuals.
- While some projects are well supported and resourced, many others are not. Due to the relatively swift development pace in many software fields, security does not always determine where resources are allocated.

- Due to its very nature, F/OSS is not always sustainable. Critical libraries used in important projects may be abandoned. Those who depend on the code must then consider questions, such as: Can the code be replaced by another project? Should it be maintained? And if so, how?
- Documentation also is a key issue in many open-source projects. Good quality documentation undoubtedly helps uptake, and also reduces accidental mistakes by coders that may result in eventual security issues. Yet with limited resources, documentation often is not a priority.

Complexity

- Reuse of code in the form of packages, dependencies, and other ways is extremely common, and overall preferable. The reuse of proven, reviewed, tested, and thus trusted implementations for security-relevant code is better than having large numbers of implementations written and designed by non-experts. Nevertheless, tracking and acting upon these dependencies and possible interactions is difficult in itself, and even more so in resource-constrained environments.

Recommendations

As a key recommendation, we urge the government and its agencies to consider how it can best support open-source projects and the community while also supporting security. In particular, the important role unpaid volunteer work plays in open-source development needs to be considered. Furthermore, we note the importance of licensing and the considerable impact of choosing a particular F/OSS license. At a minimum, the GPL family and BSD family and their inherent requirements and limitations should be considered when attempting policy solutions. Detailed recommendations appear below.

Recommendations for Hardening Security Against Malicious Actors Within the Open-Source Software Supply Chain

- Educate developers in the areas of supply-chain tooling and software inclusion risks. Educate, tool, and require the verification of included software. Best practices suggest that F/OSS builds offer checksums on the binary distributions and allow users to compile these themselves as well.[2]
- Educate developers about supply-chain risks and how to manage them in codebases.
- Increase effective prosecution of malicious actors.

Recommendations for Economics and Incentives in Open-Source Ecosystems

- Provide incentives and support for improved security in projects that falling in the area of the commons Governments interested in supporting those results can engage by:
    - Considering positive reinforcement (economic, publicity, recognition, etc.) instead of negative punishment to drive the desired results for free and open-source software.
    - Providing incentives and/or recognition for using tools like the Open-Source Security Foundation SLSA version 1.0 levels.[3]
- Limit legislation expanding product liability to open-source developers. If developers, and especially experts, are discouraged from contributing – or, even worse, are forced to leave the open-source

[2] Checksums are not always present, but in over 88% of cases users may not notice checksums, understand them, know how to go about verifying them, or simply don't bother to verify packages. Over 33% of those asked specifically to verify software do so improperly and fail to catch a mismatch (partial preimage attack). Scriber, Brian, "A Supply Chain of Weak Links: Open Source Versus Proprietary Software Threat Analysis," *Proceedings of SCTE CableTec Expo Technica*, p. 7.

[3] SLSA • SLSA specification, https://slsa.dev/spec/v1.0/

space due to expansive product liability legislation – software security and privacy-preserving elements are likely to be increasingly at risk and global innovation across sectors stifled.

- Research software monoculture threats and economic incentives that drive consolidation to a single set of tools like Log4J or OpenSSL which, when compromised, have broad impacts. Can or should these be met with diversity in tooling, or are the efficiencies and potential focus gained from having fewer key projects more effective in providing secure code? Is it possible for the government to support security testing and code reviews?
- Provide F/OSS authors with access to a full range of tools that can be used to probe and review code security, including proprietary software testing tools. Making those tools easier for interested authors to access and use could support OSS security.
- Organize interoperability events that push towards standards-based approaches to bridge the proprietary/open-source schism. Use of software, services, and devices which are standards-based and have met interoperability tests are often preferred for advances in these areas.
- Support community events focused on secure coding, security testing, and interoperability. A convening function might bring together various open-source authors to demonstrate their code and collaborate on challenging projects.

We appreciate the opportunity to submit these comments, and we welcome further opportunities to engage as needed to answer any questions during this process. Please address any inquiries to M^{3}AAWG Executive Director Amy Cadagin at comments@m3aawg.org.

Sincerely,
Amy Cadagin, Executive Director
Messaging Malware Mobile Anti-Abuse Working Group
comments@m3aawg.org
P.O. Box 9125
Brea, CA 92822

Works Cited

Fox, Brian. "The EU's Product Liability Directive could kill open source." *TechRadar*, 10 July 2023. https://www.techradar.com/pro/the-eus-product-liability-directive-could-kill-open-source Accessed 30 August 2023.

Scriber, Brian. "A Supply Chain of Weak Links: Open Source Versus Proprietary Software Threat Analysis." *Proceedings of SCTE CableTec Expo Technical*, vol. 40, no. Oct 2023, pp. 1-16. *Society for Cable Television Engineers*, scte.org.

"SLSA • SLSA specification." *SLSA*, https://slsa.dev/spec/v1.0/ Accessed 30 August 2023.