DevOps, Security, and Open Source Software

David A. Wheeler, Director, Open Source Supply Chain Security

The Linux Foundation

dwheeler@linuxfoundation.org

What is open source software (OSS)?

- OSS is software licensed to users with these freedoms:
  - to run the program for any purpose,
  - to study and modify the program, and
  - to freely redistribute copies of either the original or modified program (without royalties to original author, etc.)
- Full definition: Open Source Definition (Open Source Initiative)
- Common OSS licenses include MIT, Apache-2.0, BSD-3-Clause, LGPL, GPL
- Antonyms: Closed source, proprietary software
- OSS is a kind of commercial software (licensed to the general public)
- OSS licenses enable worldwide collaborative development of software

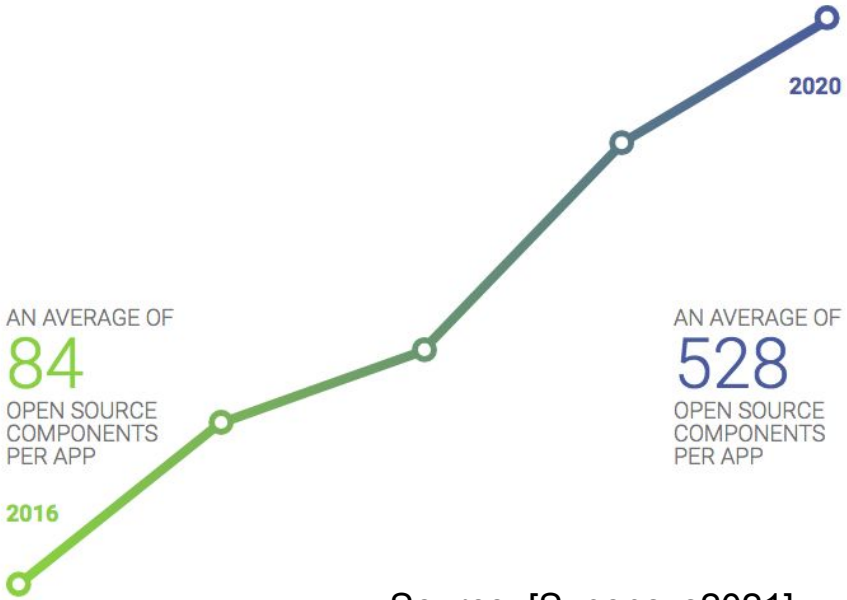
Open Source is critical part of the software supply chain

98%

Percent of general codebases and Android apps that contained OSS [Synopsys2021]

70-90%

Percent of codebase that was OSS on average [Synopsys2020] [Sonatype2020]



Source: [Synopsys2021]

Source:
[Synopsys2021] "2021 Open Source Security and Risk Analysis Report" by Synopsys https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html
[Synopsys2020] "2020 Open Source Security and Risk Analysis Report" by Synopsys https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2020-ossra-report.pdf
[Sonatype2020] "2020 Stateof the Software Supply Chain"by Sonatype https://www.sonatype.com/resources/white-paper-state-of-the-software-supply-chain-2020

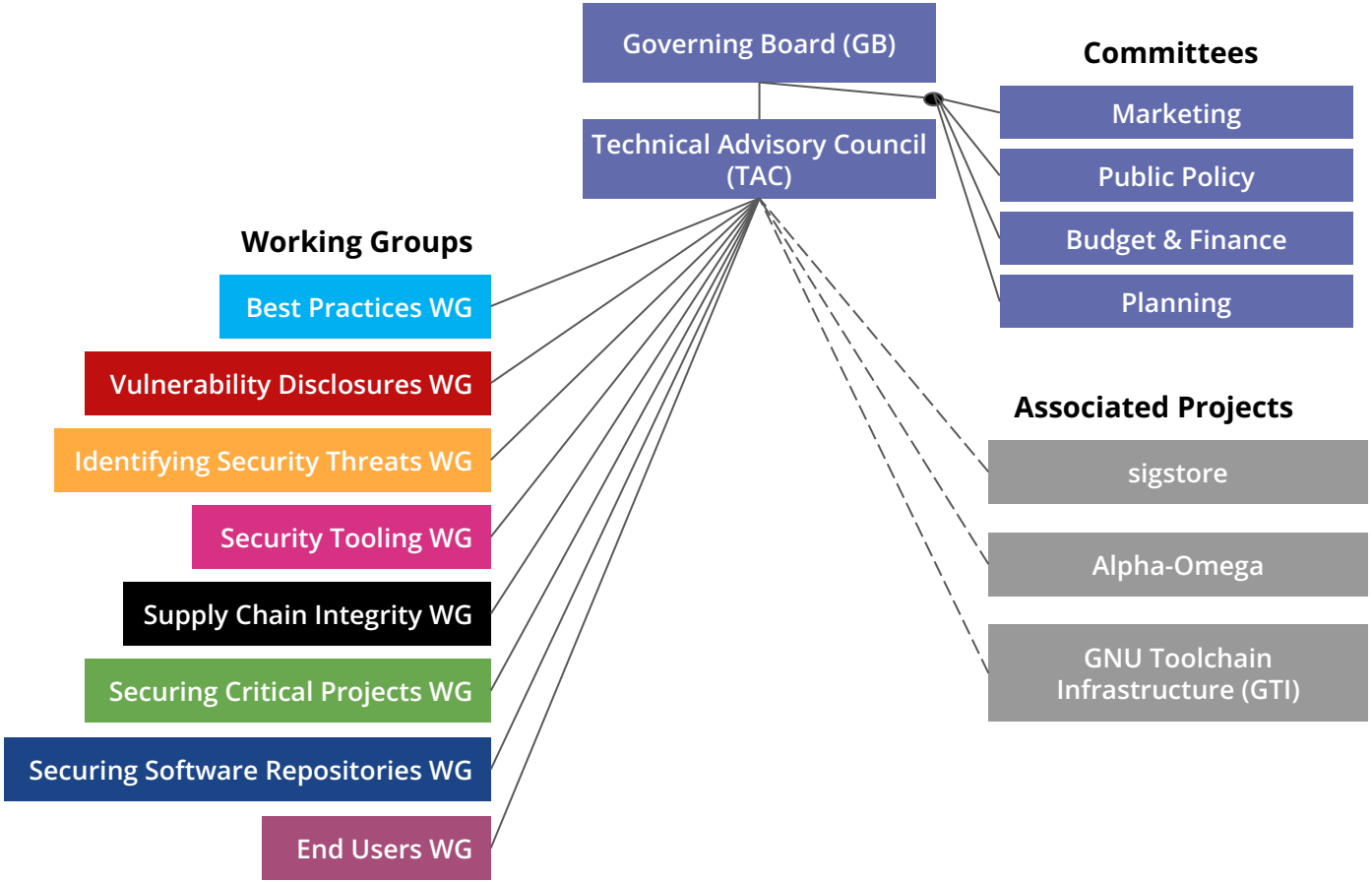
Initial Thoughts

- Attackers are attacking software worldwide, so you must prepare
- DevOps practitioners should be doing Dev**Sec**Ops
  - DevSecOps = Integrate security into DevOps (e.g., security tools in CI pipeline)
- Some OSS projects *do* apply DevSecOps
  - E.g., OpenSSF Best Practices Badge
  - % is difficult; many OSS projects are *components* & don't directly *deploy*
- Many OSS projects produce components that *enable* DevSecOps
  - Security guidance / tools / services / etc. - if they help, take advantage of them!
  - Enabling technologies, e.g., Kubernetes (k8s)
- For many OSS projects, build & release is their version of "ops"
  - Well-run OSS projects have CI pipelines that check security *before* release

OSS & OpenSSF

- Millions of OSS projects
- Many foundations which run OSS projects & relevant to DevSecOps
  - LF Foundations: Continuous Delivery Foundation, Cloud Native Computing Foundation, etc.
  - Other foundations: Apache Software Foundation, Python Software Foundation, etc.
- Can’t possibly talk about them all!
- My focus today: Open Source Security Foundation (OpenSSF)
  - “Collaboration and working both upstream and with existing communities to advance open source security for all”
  - Created in 2020 under the Linux Foundation
  - In Jan 2022 switched to member-funded model
  - Released “Open Source Software Security Mobilization Plan” May 2022

OpenSSF Structure



Sample OpenSSF Project/SIG Results

- *Secure Software Development Fundamentals*; **free** course https://openssf.org/training/courses/
- *OpenSSF Scorecards*; auto-measures OSS https://github.com/ossf/scorecard
- *OpenSSF Best Practices Badge* (for OSS projects); >5,000 participating, 3 levels (passing/silver/gold) https://bestpractices.coreinfrastructure.org
- *Alpha-Omega;* proactively find & fix vulnerabilities https://openssf.org/community/alpha-omega/
- *Vulnerability Disclosure Guide* https://github.com/ossf/oss-vulnerability-guide/
- Concise guides:
  - *Concise Guide for Developing More Secure Software*
  - *Concise Guide for Evaluating Open Source Software*

Concise Guide for Developing More Secure Software

1. Ensure all privileged developers use multi-factor authentication (MFA) tokens.
2. Learn about secure software development.
3. Use a combination of tools in your CI pipeline to detect vulnerabilities.
4. Evaluate software before selecting it as a direct dependency. [“Evaluating”]
5. Use package managers.
6. Implement automated tests.
7. Monitor known vulnerabilities in your software’s direct & indirect dependencies.
8. Keep dependencies reasonably up-to-date.
9. … (many more)

https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Developing-More-Secure-Software.md#readme

Concise Guide for Evaluating Open Source Software

1. Can you avoid adding it?
2. Are you evaluating the intended version?
3. Is it maintained?
4. Is there evidence that its developers work to make it secure? [“Developing”]
5. Is it easy to use securely?
6. Are there instructions on how to report vulnerabilities?
7. Does it have significant use?
8. What is the software’s license?
9. What is your evaluation of its code?

https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Evaluating-Open-Source-Software.md#readme

OpenSSF Mobilization Plan: 3 Goals, 10 Streams

Security Education

Risk Assessment

Digital Signatures

Memory Safety

Incident Response

Better Scanning

Code Audits

Data Sharing

SBOMs Everywhere

Improved Software Supply Chains

Continuous Delivery Foundation

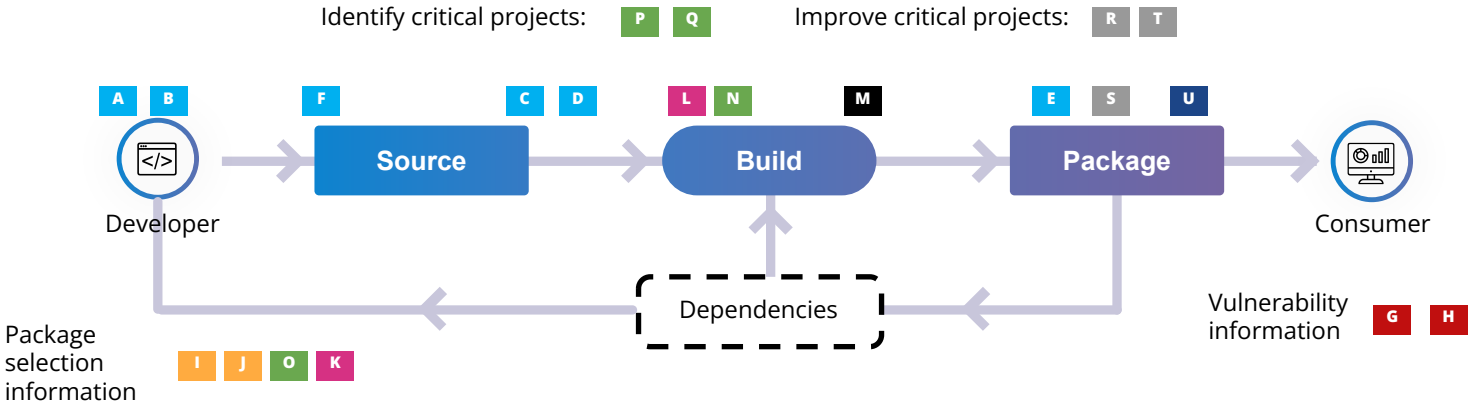
- SIG Software Supply Chain: https://github.com/cdfoundation/sig-software-supply-chain
- SIG Interoperability: https://github.com/cdfoundation/sig-interoperability
- CDEvents: https://cdevents.dev/
- Best Practices: https://bestpractices.cd.foundation/ and https://bestpractices.cd.foundation/learn/supplychain/
- CDF Reference Architecture: Preview site https://deploy-preview-23--prod-bp-cdf.netlify.app/architecture/

Get involved!

- Many other OpenSSF projects/SIGs, some in early stages
  - Sigstore
  - Supply chain Levels for Software Artifacts (SLSA) https://slsa.dev/
  - “SBOM Everywhere” tool work
  - Education work (deeper, K-12, manager, etc.)
  - Metrics Dashboard SIG - easily see status of an OSS project
  - OSS critical projects identification
  - In discussion: Microsoft’s Secure Supply Chain (SSC) work
- To get involved in OpenSSF see https://openssf.org
  - Biweekly meetings, mailing lists, Slack
- Many other OSS projects & foundations, e.g., Continuous Delivery
- Industry, academia, & government should work together
- The best way to influence an OSS project direction is to get involved!

Backup Slides

How OpenSSF Projects Work Together



Best Practices WG

- A. Secure Software Development Fundamentals courses (education)
- B. Security Knowledge Framework (SKF): Hands-on course (education), with OWASP
- C. OpenSSF Best Practices Badge
- D. Scorecards
- E. Great MFA distribution SIG
- F. Common Requirements Enumeration (CRE)

Vulnerability Disclosures WG

- G. Guide to coordinated vulnerability disclosure for OSS projects; Vulnerability Disclosures Whitepaper
- H. osv-schema

Identifying Security Threats WG

- I. security-reviews
- J. Project-Security-Metrics: Dashboard

Security Tooling WG

- K. ossf-cve-benchmark: measure tools
- L. Web Application Definition

Supply Chain Integrity WG

- M. Supply-chain Levels for Software Artifacts (SLSA)

Securing Critical Projects WG

- N. allstar
- O. package-feeds / package-analysis
- P. criticality_score
- Q. Harvard study

End Users WG

Special Initiative Funds

- R. Project Alpha-Omega
- S. sigstore
- T. GNU Toolchain Infrastructure (GTI) support

Securing Software Repositories WG

- U. Securing Software Repositories

Presentation Purpose

"examine the current state of DevSecOps in the open-source community, and will highlight opportunities for industry, government, and others to leverage existing projects, tools, and resources and collaborate with the community on DevSecOps-related efforts."

OR: “discuss the relevant OpenSSF projects and activities that NIST can leverage as we are developing a DevSecOps project to demonstrate existing recommended software development and supply chain practices in collaboration with the community.”

Your panel is at 13:20 ET with Google and Chainguard. Each panelist can provide a 15-minute presentation then there will be a 15-minute Q&A session.

https://www.nccoe.nist.gov/projects/software-supply-chain-and-devops-security-practices

This presentation released under the CC-BY-4.0 license