



Securing Open Source Software

End-to-end, At massive scale,
Together

#ossummit @WhyHiAnnabelle & @SecurityCRob



Why we brought you all here today....

- About us
- Defining the scope
- Threat Factors of the Open Source
- A Modest Proposal
- Q&A





Anne Bertucio (@WhyHiAnnabelle)

- Lead for Google's Open Source Programs Office (OSPO) Program Dev team
- We help Googlers release open source and contribute to open source
- Includes promoting good security practices (like vuln disclosure!)
- Member of OpenSSF Vuln Disclosure WG
- Previously: Kata Containers and OpenStack contributor, very mediocre bicycle racer



CRob, n, adj, and v (@SecurityCRob)

- Pronunciation: U.S. (K-robe)
- Over 25 years of Enterprise-class Architecture, Engineering, Operations, and Security experience
- Ambassador For Intel Product Assurance and Security
- Working Group lead for the OpenSSF Dev Best Practices & Vuln Coordination WGs, OpenSSF TAC and Public Policy Committee, FIRST PSIRT TPC WG, and others
- Co-Author FIRST PSIRT Services Framework & others
- Pirate-enthusiast & hat-owner

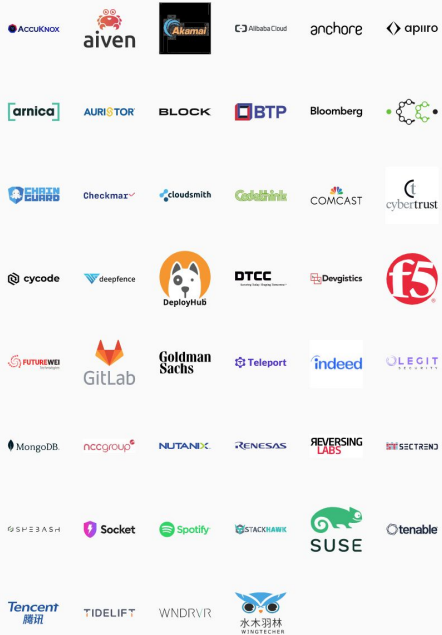
Open Source Security Foundation (OSSF)

OpenSSF is committed to collaboration and working both upstream and with existing communities to advance open source security for all.

Premier Members



General Members



Associate Members



openssf.org

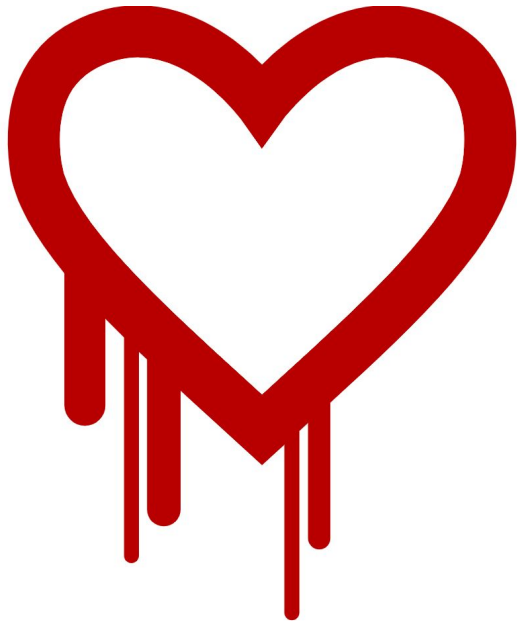


OpenSSF

OPEN SOURCE SECURITY FOUNDATION



End-to-End



<https://heartbleed.com/>

Vulnerability in **popular** open source library OpenSSL that could leak sensitive information otherwise thought protected by SSL/TLS encryption

This is a widely-used method to protect communications over TCP-IP-based networks (example - The Internet)

"At the time of disclosure, some 17% (around half a million) of the Internet's secure web servers certified by trusted authorities were believed to be vulnerable to the attack, allowing theft of the servers' private keys and users' session cookies and passwords" ⁽⁷⁾

Public disclosure - 7 April 2014

Exposure -

- 21 June 2014 - 309,197 public web servers remained vulnerable ⁽⁸⁾
- 6 July 2017, the number had dropped to 144,000 ⁽⁹⁾
- 11 July 2019, 91,063 devices were vulnerable ⁽¹⁰⁾

At the time, OpenSSL had **TWO** full-time developers to develop, maintain, test, and review 500,000 lines of code ⁽⁷⁾

(7) - [Heartbleed](#)
(8) - [300k vulnerable to Heartbleed two months later](#)
(9) - [Heartbleed's Heartburn: Why a 5 year Old Vulnerability Continues to Bite](#)
(10) - [Heartbleed Report](#)

Problem Overview - Securing the open source ecosystem

- It has been estimated that FOSS constitutes 80-90% of any given piece of modern software ⁽¹⁾
- One report found that 84% of these codebases had at least one vulnerability, with the average having 158 per codebase ⁽²⁾
- Other reports discover that average applications contain 118 libraries with roughly 1/3 being active; The average library age was 2.6 years old ⁽³⁾
- Over a 10 year period the volume of vulns has increased over 4 times [as measured with CVE] ⁽⁴⁾
- Most OSS vulns are discovered in indirect dependencies ⁽⁵⁾
- A typical vuln can go undetected for 218 weeks, and typically takes 4 weeks to get resolved once the project is alerted to it ⁽⁶⁾

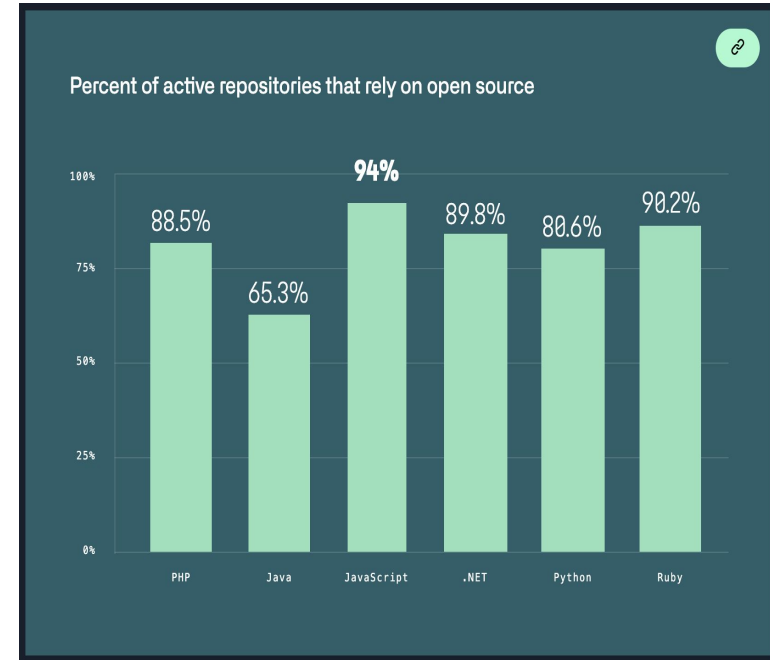


Image - [%Rely on OSS](#)

(1) - [State of the Software Supply Chain](#)
(2) - [2021 Open Source Security and Risk Analysis Report](#)
(3) - [2021 State of Open Source Security Report](#)

(4) - [2020 Red Hat Risk Report](#)
(5) - [2020 State of Open Source Security Report](#)
(6) - [2020 State of the Octoverse](#)

“According to a U.S. Department of Homeland Security advisory, the affected versions of SolarWinds Orion are versions are 2019.4 through 2020.2. 1 HF1. **More than 18,000 SolarWinds customers installed the malicious updates, with the malware spreading undetected**” - TechTarget - June 16, 2021

“Log4j 2.x is in the top 0.003% percentile in popularity by downloads out of a total population of 7.1 million.

In short - it's as popular as components get.” -

Sonotype - “Log4shell by the numbers- Why did CVE-2021-44228 set the Internet on Fire?” - December 14, 2021

“Unpatched open source software flaw blamed for massive Equifax breach. It's no surprise that Web application attacks are the leading cause of large breaches. **The *average* Web application or API has 26.7 serious vulnerabilities.** And organizations often have hundreds, thousands, or even tens of thousands of applications.”
eSecurity Planet - December 12, 2017



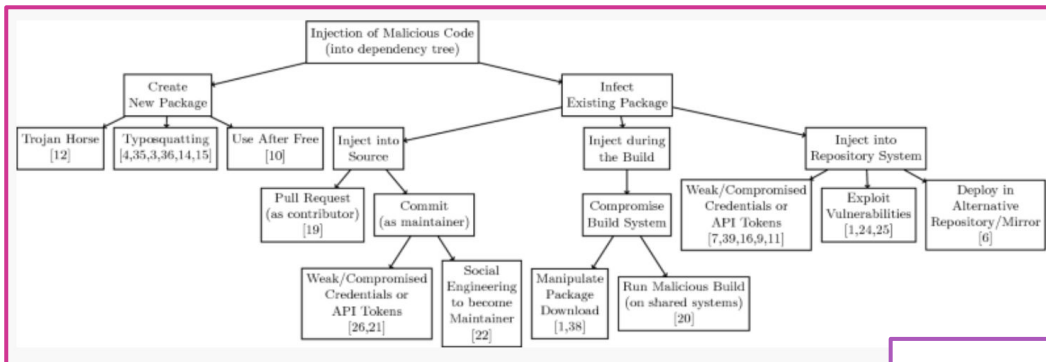
At Massive Scale

What makes OSS a unique target for adversaries?

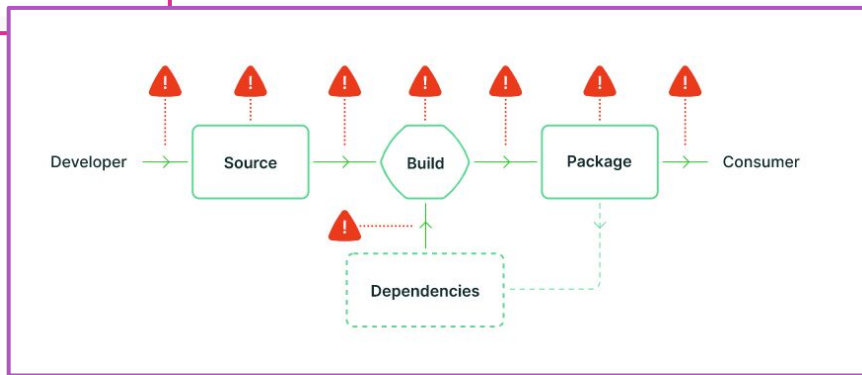
Many of the best things about open source development invite unique security challenges:

- Deobfuscated and **public-facing source code** lowers attacker barrier to entry
- Distributed **community-driven development** **with contributions from unknown third-parties**
- **Tragedy of the commons** regarding security analysis
- **Lack of consistently-deployed security standards, reviews and tooling**
- (Often) **decreased capacity for vulnerability remediation**
- Lack of resources for monitoring & typical **underpreparedness for incident response**
- **Different economic incentives & feedback loops** than: enterprise devs; threat actors
- In spite of this: many **high-value targets**, foundational to enterprises and the internet itself

OSS security is about more than just vulnerabilities in source code



Source: Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks (2020)



Source: slsa.dev

Top 50 packages (for each package manager)	Avg. dependent projects	Avg. direct contributors
Maven packages	167k	99
pip packages	78k	204
npm packages	3.5m	35
NuGet packages	94k	109
RubyGems packages	737k	146


Source: Github State of the Octoverse 2019

Open source projects have an average of **180 package dependencies**

The top 50 OSS projects with the most downstream dependencies had an average of **3.6 million projects dependent upon them**

Vulns in OSS have been central to major breaches and some of these vulns were **not found until decades** after their creation



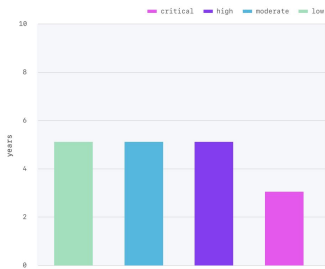


The creation of potentially exploitable vulnerabilities increasingly outpaces the rate at which we can search for and remediate them, and this problem is one that only gets worse with time.

The number of vulnerabilities “in the wild” outpaces the speed at which the security community can patch or even identify them.

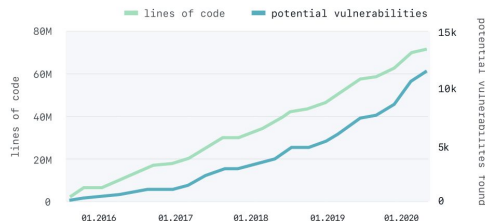
And each day, the world contains more lines of source code than it ever has before

Time to identify and fix a vulnerability, by severity



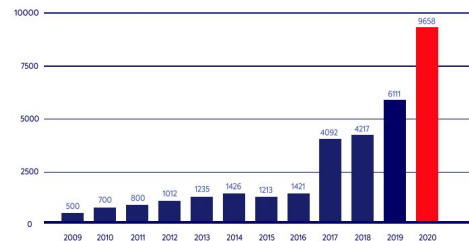
Source: Github State of the Octoverse 2020

Potential vulnerabilities found in source code scale with lines of code written



Source: Github State of the Octoverse 2020

Open Source Vulnerabilities per Year: 2009-2020



Source: Whitesource State of Open Source Security 2021

Every year, more lines of OSS are written than ever before, but vuln detection lags years behind

+

Vulnerabilities seem to scale with lines of code - but other metrics besides LOC show similar patterns

=

The number of reported vulnerabilities in open source codebases is growing each year



Together

User responsibility

- If you consume it, you need to monitor it and upgrade it
- Different from vendor software → get an individual alert, analysis specific to your use case
- Maintainers don't know who's running their projects, their environment, or what they're using it for
- Log4j → “Am I affected?” “Please fill out my company's incident response form”
 - Isn't the relationship between OSS maintainers and users

Moving beyond “responsible consumption”...

*“If you use software from a project, maybe you could **contribute** back to that project?”*

- A very wise person



COORDINATED APPROACH - WHY AND HOW



What is needed to make open source more secure?

- **Threat model** to understand the many places & times at which a project can be compromised.
- Data-driven **identification of the world's most critical open source projects**
- **Interventions to prevent vulnerabilities in the first place**, introduced at various parts of SDLC
- **Preventing inherited security debt** through tools that can help developers obtain and users assess the security of a project (such as the OSSF Best Practices badge and OSSF Scorecards)
- Continued research and open source **tool development** for scalable bug-hunting & remediation
- Investments in **technical security reviews** of critical open source projects,
- **Coordinated patching and incident response support** to respond to high-impact vulnerabilities in OSS
- Better **vulnerability disclosure** processes, response, and workflows.
- **Coordinated, impact-prioritized funding** for security improvements, audits, and research

OSSF Working Groups

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.

[GitHub](#) | [Slack](#) | [Email List](#)

Securing Critical Projects

This group exists to identify and help to allocate resources to secure the critical open source projects we all depend on.

[GitHub](#) | [Slack](#) | [Email List](#)

Supply Chain Integrity

This group is helping people understand and make decisions on the provenance of the code they maintain, produce and use.

[GitHub](#) | [Slack](#) | [Email List](#)

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.

[GitHub](#) | [Slack](#) | [Email List](#)

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.

[GitHub](#) | [Slack](#) | [Email List](#)

Security Tooling

This group's mission is to provide the best security tools for open source developers and make them universally accessible.

[GitHub](#) | [Slack](#) | [Email List](#)

Vulnerability Disclosures

This group is improving the overall security of the OSS ecosystem by helping advance vulnerability reporting and communication.

[GitHub](#) | [Slack](#) | [Email List](#)

Someone IS doing something

...and you can help!

10 Streams of Investment for Open Source Security



OpenSSF and The Linux Foundation propose 10 streams of investment to improve cybersecurity practices within open source development, code reviews, developer training, and software distribution.

[Read the Plan](#)



Security Education



Risk Assessment



Digital Signatures



Memory Safety



Incident Response



Better Scanning



Code Audits



Data Sharing



SBOMs Everywhere



Improved Software
Supply Chains

<https://openssf.org/oss-security-mobilization-plan/>

Thank you!

openssf.org/getinvolved
github.com/openssf



@whyhiannabelle



@SecurityCRob



OpenSSF

OPEN SOURCE SECURITY FOUNDATION



THE LINUX FOUNDATION



OPEN SOURCE SUMMIT

NORTH AMERICA