

SECURING OPEN SOURCE DEPENDENCIES

It's not just your code that you need to secure.



Rana Khalil
Application Security
Engineer Team Lead

June 15 -17, 2023



ABOUT ME



[LINKEDIN.COM/IN/RANAKHALIL1/](https://www.linkedin.com/in/ranakhalil1/)



[YOUTUBE.COM/RANAKHALIL101](https://www.youtube.com/RANAKHALIL101)



[ACADEMY.RANAKHALIL.COM](https://academy.ranakhalil.com)



[TWITTER.COM/RANA_K_KHALIL](https://twitter.com/RANA_KHALIL)



RANAKHALIL101.MEDIUM.COM



THE WHAT, WHY & HOW

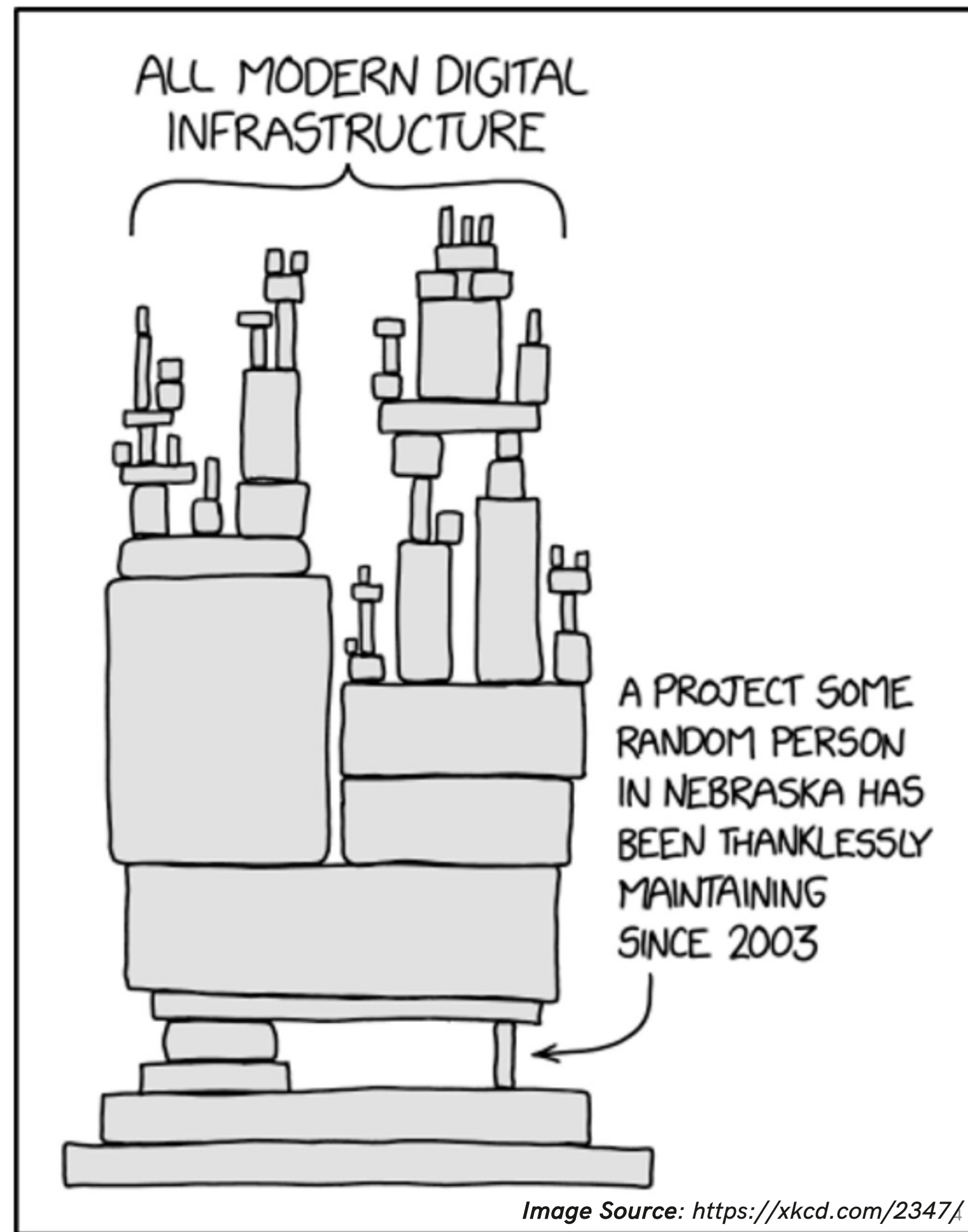


WHAT are open source dependencies?

WHY should dependencies be managed?

HOW can dependencies be secured?

WHAT ARE OPEN SOURCE DEPENDENCIES?



The Code Cocktail

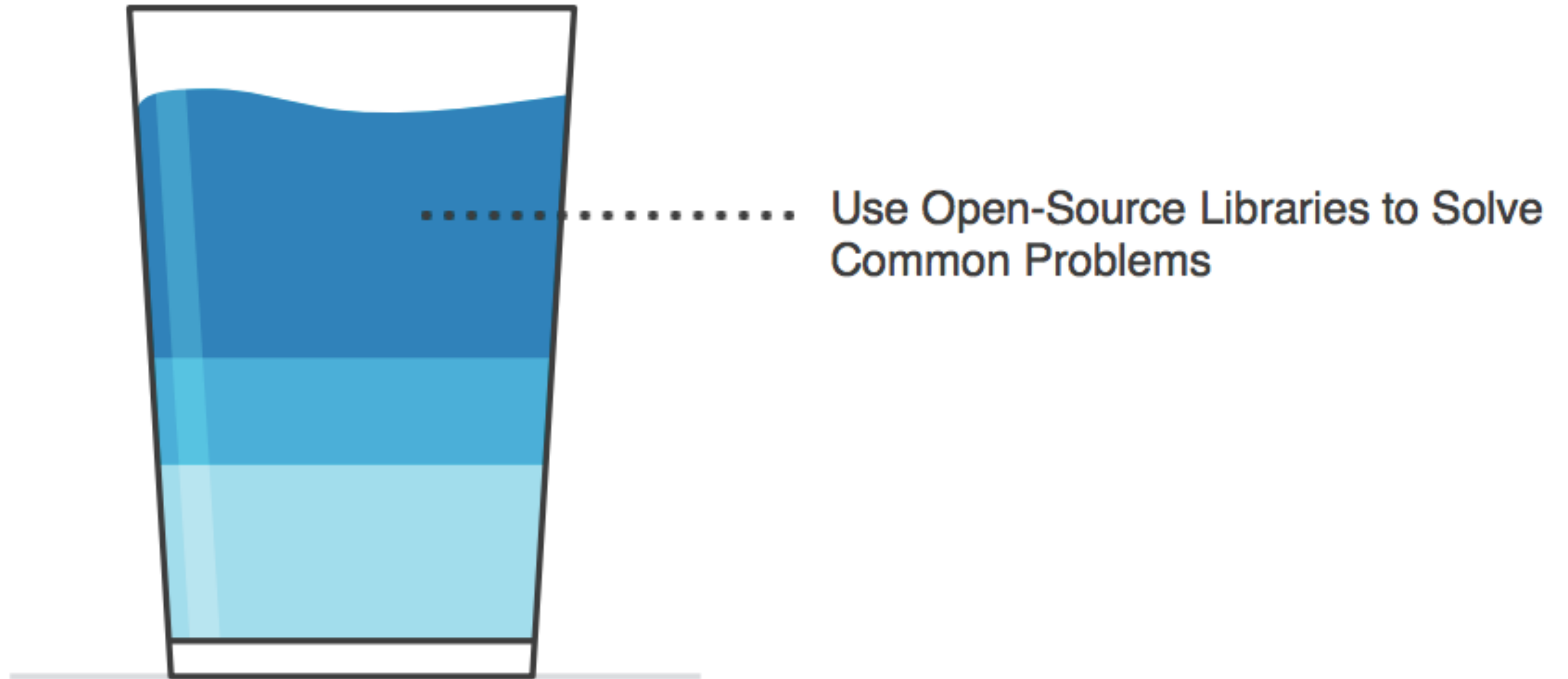


Image Source: HITBGSEC 2017 Keynote 1 - Finding Vulnerabilities and Malware in Open Source Code at Scale by Mark Curphey

The Code Cocktail

Open Source = ~ 90%

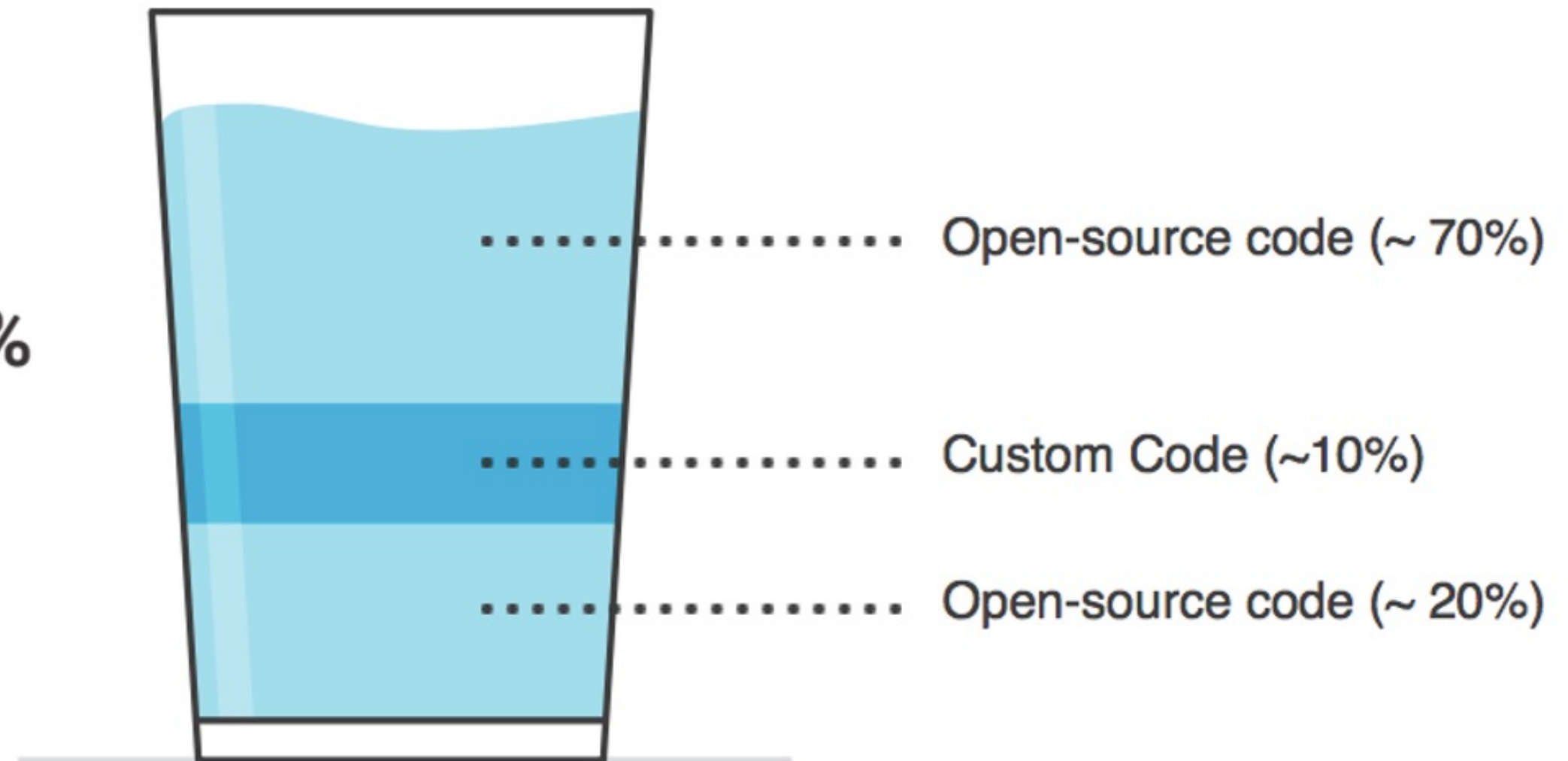


Image Source: HITBGSEC 2017 Keynote 1 - Finding Vulnerabilities and Malware in Open Source Code at Scale by Mark Curphey

Direct and Indirect Dependencies

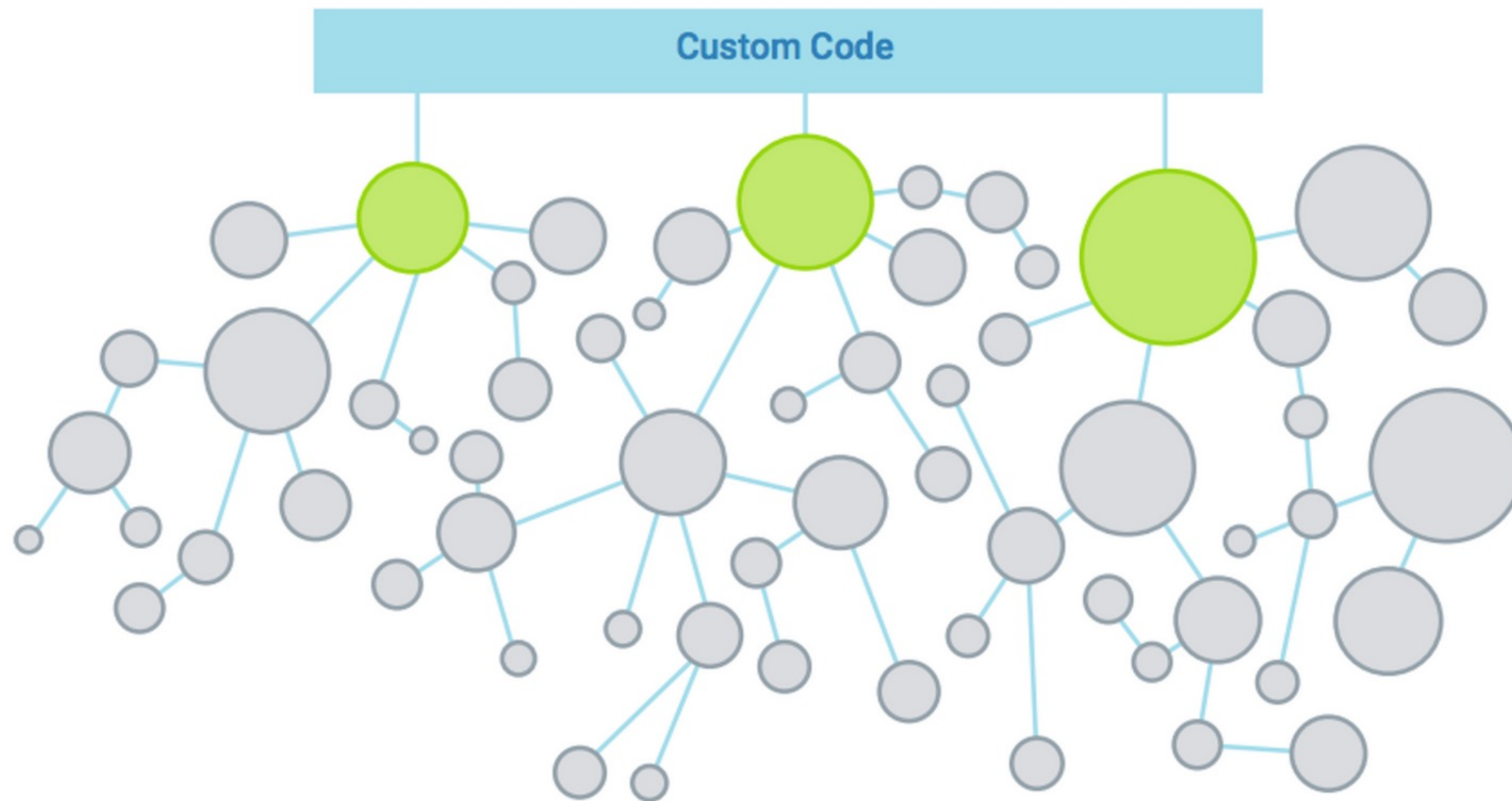
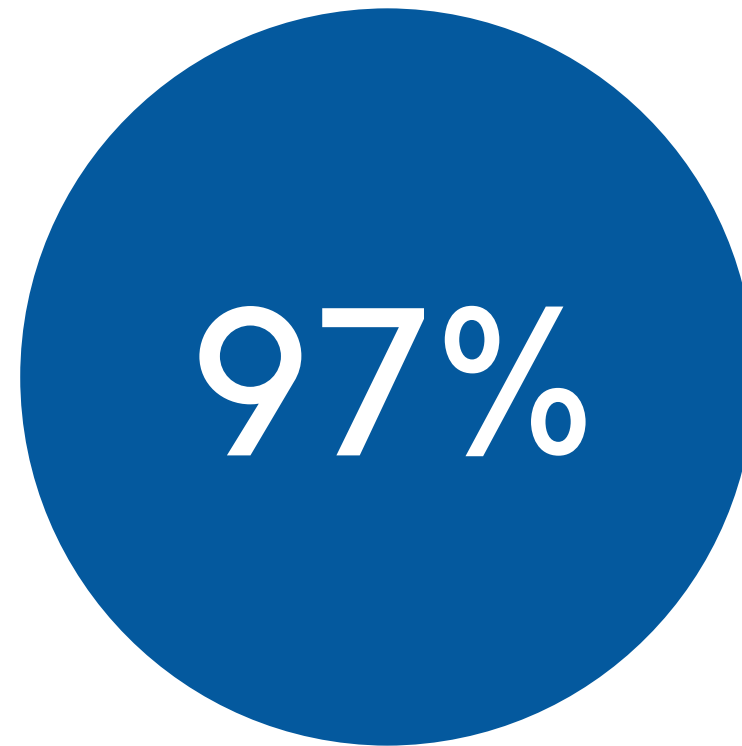


Image Source: HITBGSEC 2017 Keynote 1 - Finding Vulnerabilities and Malware in Open Source Code at Scale by Mark Curphey

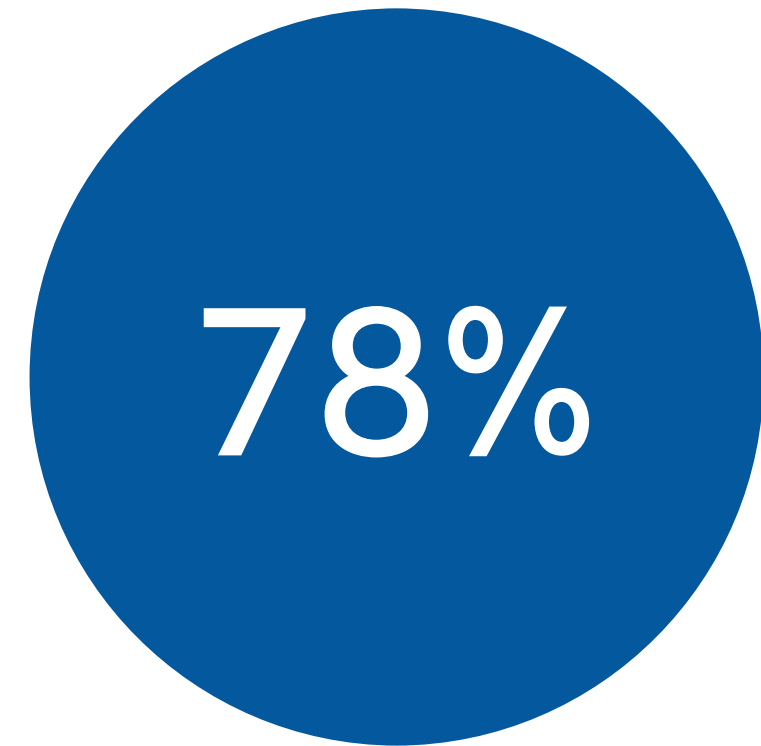
OSSRA 2022 Report



Number of codebases audited in 2021.



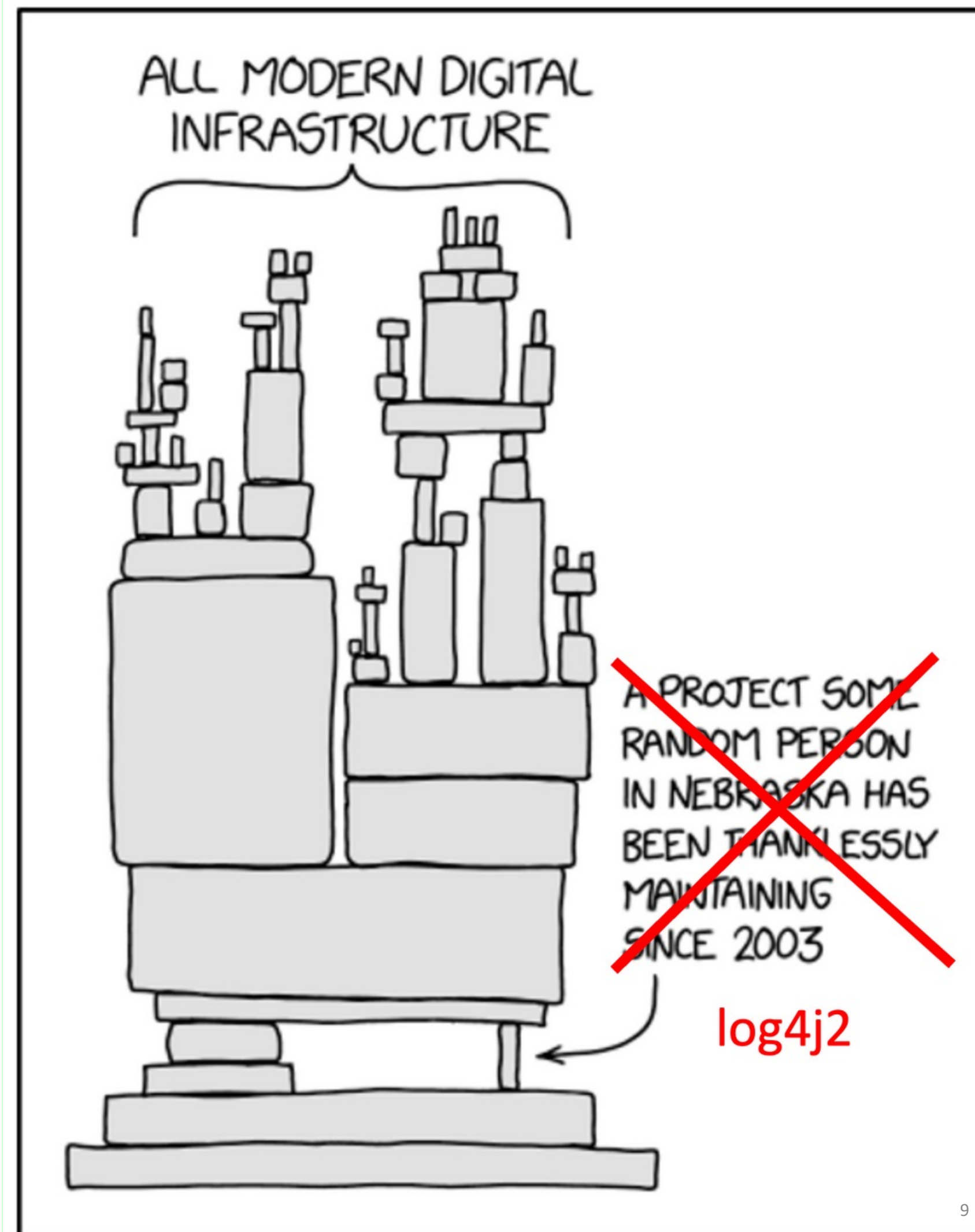
Percentage of codebases contained open source code.



Percentage of open source code in codebases.

Statistics Source: <https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-2022.pdf>

WHY SHOULD OPEN SOURCE DEPENDENCIES BE MANAGED?




Software

How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using

By Chris Williams, Editor in Chief 23 Mar 2016 at 01:24

168  SHARE ▼



Careful, careful ... Don't fumble this like the JS world (Credit: Claus Rebler)

```
module.exports = leftpad;

function leftpad (str, len, ch) {
  str = String(str);

  var i = -1;

  if (!ch && ch !== 0) ch = ' ';

  len = len - str.length;

  while (++i < len) {
    str = ch + str;
  }

  return str;
}
```

Article Source: https://www.theregister.com/2016/03/23/npm_left_pad_chaos/

**EQUIFAX BREACH EXPOSES
143-MILLION PEOPLE TO
IDENTITY THEFT,
CANADIANS AFFECTED**

**Equifax breach was 'entirely
preventable' had it used basic security
measures, says House report**

Equifax Officially Has No Excuse

A patch that would have prevented the devastating Equifax breach had been available for months.

**Report: Code Responsible for Equifax
Breach Downloaded 21 Million Times Last
Year**

**Most of the Fortune 100 still use flawed
software that led to the Equifax breach**

Why Experts are Calling Log4j the Worst Security Flaw of the Decade and Why You Should Care

🕒 January 18, 2022

[What's Log4j?](#)

[What does the future hold?](#)

[What's the Log4j vulnerability?](#)

[Best Practices](#)

[What's the impact?](#)

The Log4j vulnerability sent shockwaves throughout the entire cybercommunity when it was discovered in December 2021. Since then, organizations have been scrambling to uncover what systems use Log4j, if they're affected, and how to patch them before a [breach](#) occurs. Security researchers and IT teams around the world are racing to deploy solutions and map out the long-term implications of the situation.




But what exactly is Log4j? What's vulnerable? Why are experts calling it the [worst security flaw of the decade](#)? And why should you be worried about it? Keep reading to learn more about the Log4j crisis—what it is, what the future holds, and most importantly, how you can protect yourself.

Disclaimer: Please note that the information provided herein offers guidelines only. It is not exhaustive and does not constitute legal, insurance, or cybersecurity advice. For more guidance, please consult a lawyer, a licensed insurance representative, and/or a cybersecurity specialist.

Article Source: <https://prolink.insure/log4j-security-vulnerability/#:~:text=In%20December%202021%2C%20security%20researchers,%2C%20files%2C%20and%20other%20data>

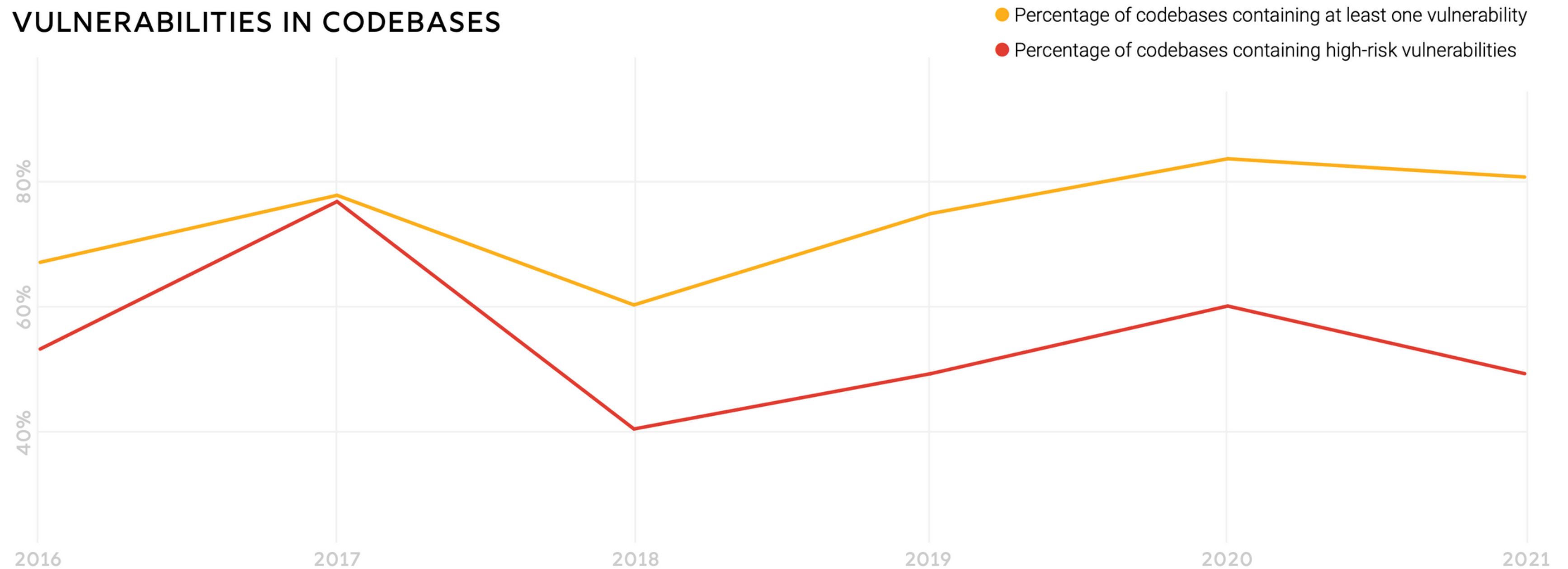
OWASP Top 10

Most Critical Security Risks to Web Applications

-  A01 - Broken Access Control
-  A02 – Cryptographic Failures
-  A03 - Injection
-  A04 – Insecure Design
-  A05 – Security Misconfiguration
-  A06 – Vulnerable and Outdated Components
-  A07 – Identification and Authentication Failures
-  A08 – Software and Data Integrity Failures
-  A09 – Security Logging and Monitoring Failures
-  A10 – Server Side Request Forgery (SSRF)

OSSRA 2022 Report

VULNERABILITIES IN CODEBASES

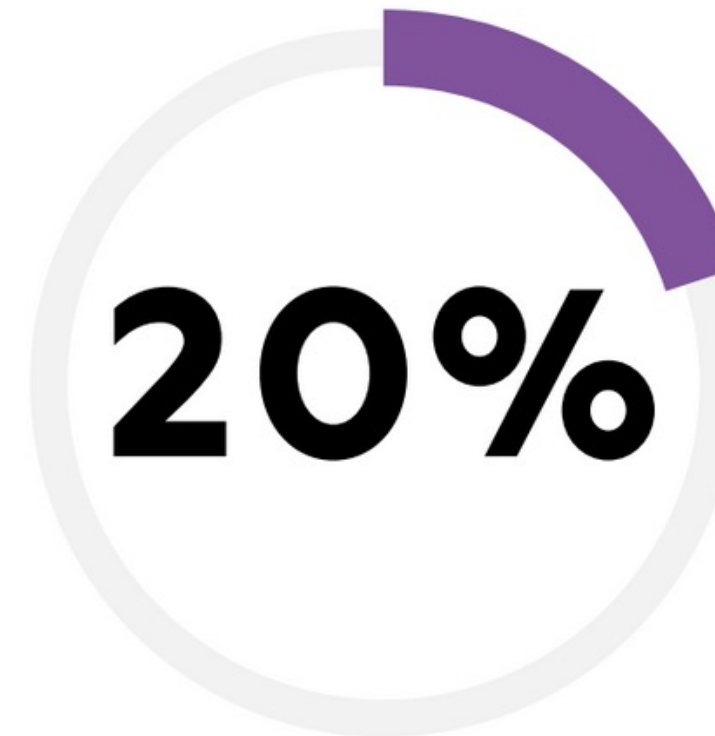


Statistics Source: <https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-2022.pdf>

OSSRA 2022 Report



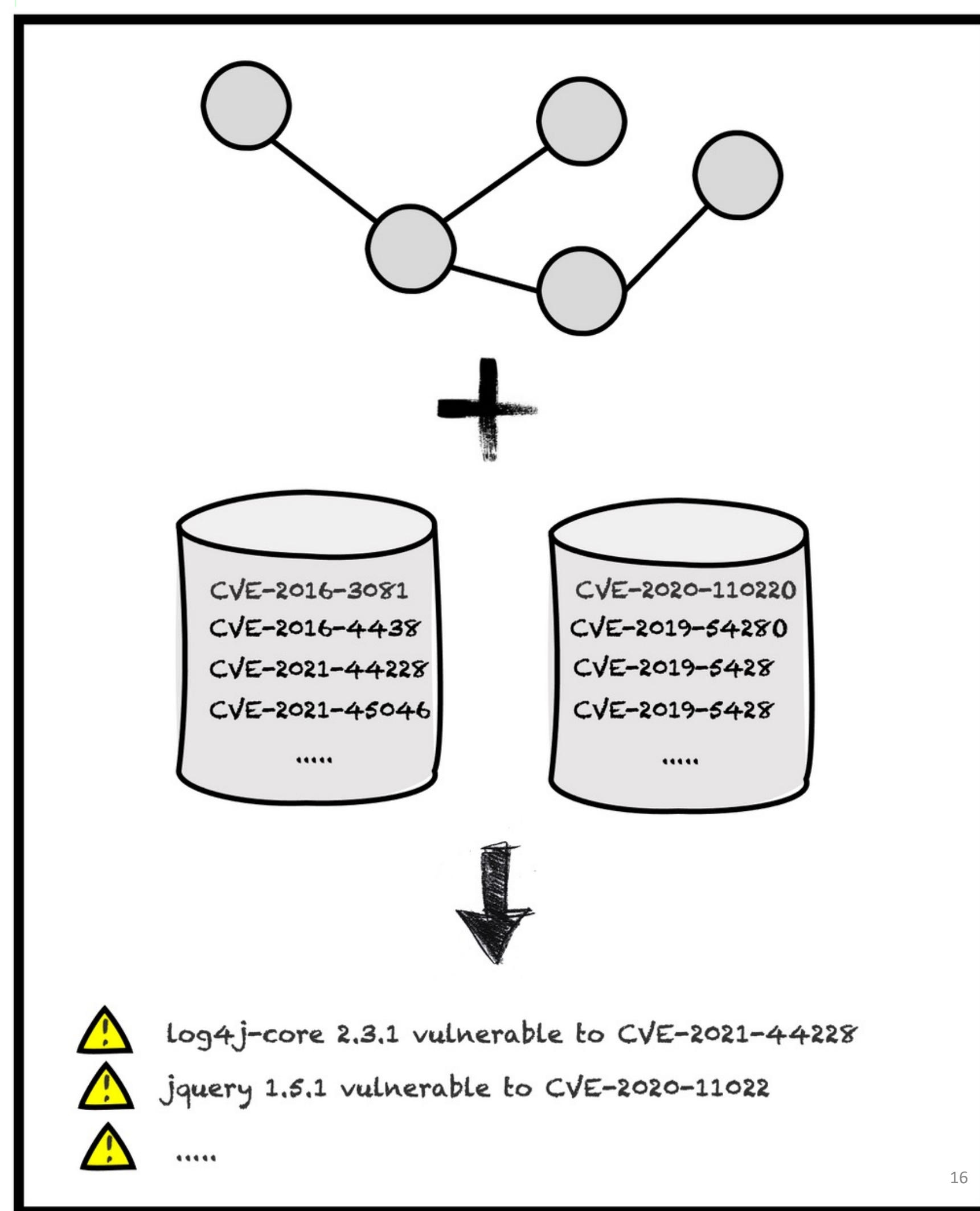
OF AUDITED
CODEBASES
CONTAINED
LICENSE
CONFLICTS



CONTAINED OPEN
SOURCE WITH
NO LICENSE OR A
CUSTOM LICENSE

Statistics Source: <https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/rep-ossra-2022.pdf>

HOW CAN OPEN SOURCE DEPENDENCIES BE SECURED?



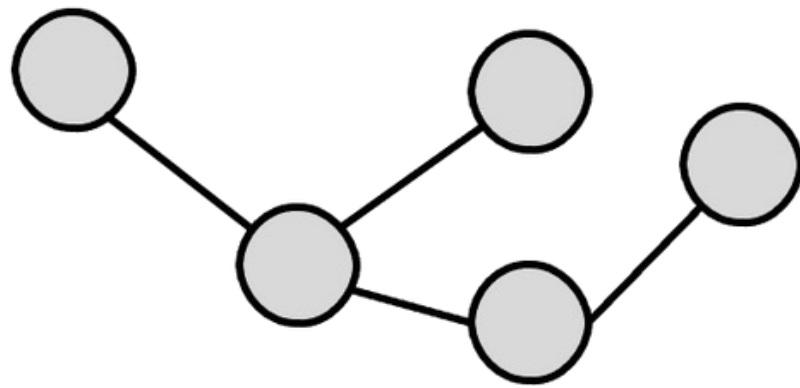
SBOM

- A Software Bill of Materials (SBOM) is a list of all the open source and third-party components (direct and indirect) that an application uses. Includes:
 - Package Supplier
 - Package Name
 - Package Version
 - Relationship
 - Creator
 - etc.
- Became a requirement in the U.S. in 2021 after the release of the [U.S. Presidential Executive Order on Improving the Nation's Cybersecurity](#).

Total Carbohydrate 21g		8%
Dietary Fiber 0g		0%
Total Sugars 13g		
Includes 13g Added Sugars		25%
Protein <1g		
Vitamin D 0mcg	0%	• Calcium 0mg 0%
Iron 0.3mg	0%	• Potassium 40mg 0%
*The % Daily Value tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.		
INGREDIENTS: CORN SYRUP; ENRICHED WHEAT FLOUR (FLOUR; NIACIN; FERROUS SULFATE; THIAMIN MONONITRATE; RIBOFLAVIN; FOLIC ACID); SUGAR; CONTAINS 2% OR LESS OF: PALM OIL; SALT; GLYCERYL MONOSTEARATE; ARTIFICIAL FLAVOR; CORNSTARCH; GLYCERIN; POTASSIUM CARBONATE; ARTIFICIAL COLOR (RED 40); SODIUM CARBONATE. ©		

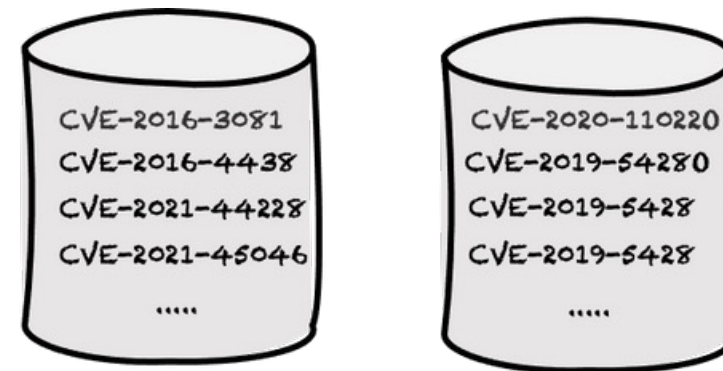
Software Composition Analysis

Software Composition Analysis (SCA) is an automated process of analyzing and managing open source dependencies. SCA tools are used to generate an SBOM, identify known vulnerabilities in open source dependencies and ensure license compliance.



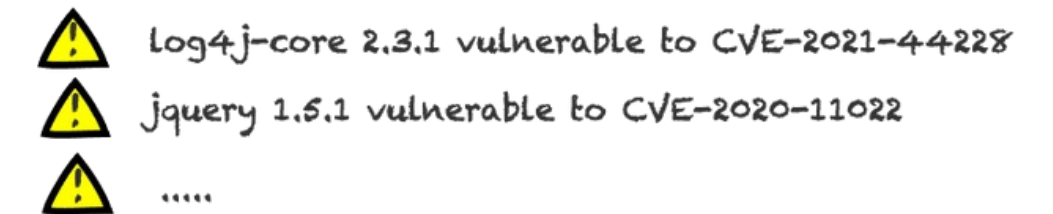
STEP 1

Determine direct and indirect dependencies (SBOM).



STEP 2

Check vulnerability data sources.



STEP 3

Generate report on vulnerable dependencies.

SCA Tools



npm-audit

Retire.js

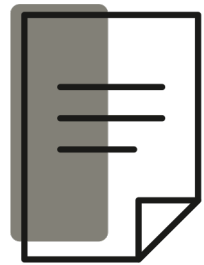
VERACODE

JFROG XRAY

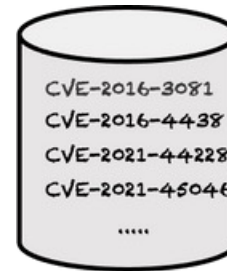


FOSSA

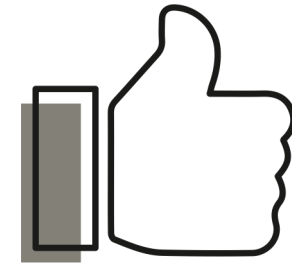
SCA Tool Features



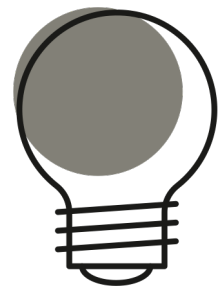
Language
Support



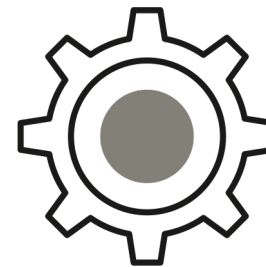
Vulnerability
Data Sources



Accuracy



Remediation
Capability



Configuration
& Integration



Reporting

Example - NPM Audit



To Summarize...

WHAT?

Open source dependencies are open source software that an application uses. They come in two forms: direct dependencies and indirect (transitive) dependencies.

WHY?

Open source dependencies should be managed for the purpose of risk management, privacy, security and license compliance.

HOW?

Open source dependencies can be managed using Software Composition Analysis (SCA) tools.

THANK YOU!



[LINKEDIN.COM/IN/RANAKHALIL1/](https://www.linkedin.com/in/RANAKHALIL1/)



[YOUTUBE.COM/RANAKHALIL101](https://www.youtube.com/RANAKHALIL101)



[ACADEMY.RANAKHALIL.COM](https://academy.ranakhalil.com)



[TWITTER.COM/RANA_KHALIL](https://twitter.com/RANA_KHALIL)



RANAKHALIL101.MEDIUM.COM