OWASP
Open Web Application
Security Project

Software Composition Analysis

By Richard Stephanus

24th October 2019

Agenda

1. Introduction

   Overview and Motivation

2. How to solve this problem?

   OWASP Dependency Check. Nexus. Artifactory.

3. How to get better?

Introduction
A real life example

Target: analyzed application was a middleware for REST services.

Some technical aspects:

- Key technologies: Java
- ~ 28.000 lines of code
- ~ 55 3rd party libraries
- ~ 38 APIs (REST). 82 functions
- Partly cloud based (MS Azure)
- ~ 120 Mio. Requests (clicks) per year

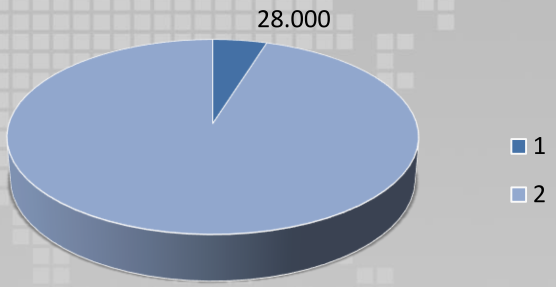
- Project key facts:
- Project duration: 8 months
- Agile approach with 16 Sprints (every 2 week)
- Up to 26 team members (18 developers)

Introduction
Some statistics …

Programmers wrote
~ 28.000 Line of code (loc)
Sounds much?

Anticipate a 3rd party component consists only of 10.000 lines of code (log4j = 180K loc*)

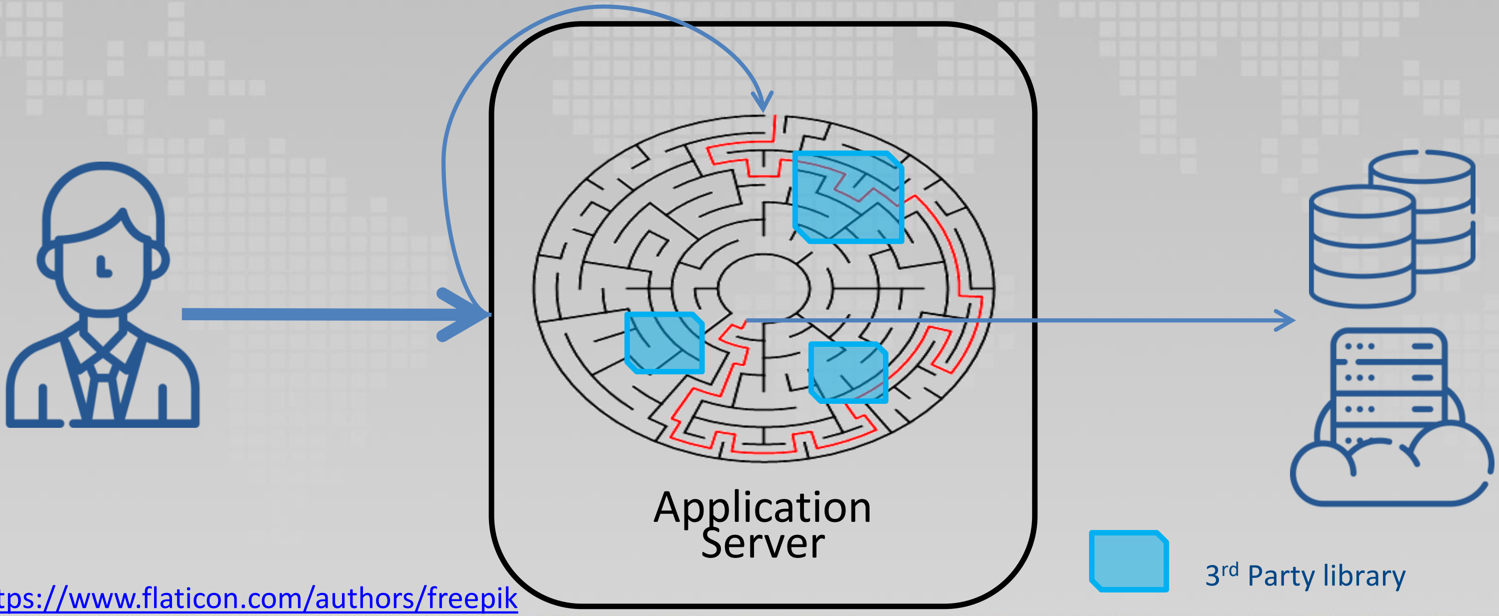
Whole application consists of
28K + 55 x 10.000 → 578K loc



* https://www.openhub.net/p?query=log4j

Introduction

What is this code doing?



🟦 3rd Party library

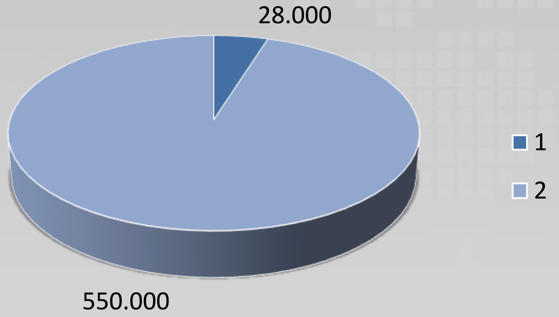
Icons from https://www.flaticon.com/authors/freepik

Introduction
Vulnerabilities in LOCs

Quality of source code changes depending on:

- Programming Language
  Example: compare Assembler with JavaScript?
- Type of Application
  Example: Mobile App and command line tool?
- Experience of developers
  ➔ Very different numbers
  ➔ 6 defects / 100.000 LOC *

Package	LOC
MySQL	2.862.087
PHP	3.882.984
Apache Tomcat	1.136.822
Linux	25.646.844
Mozilla Firefox	14.045.424
Google Chrome	15.441.702
Log4j	180.173
Spring	1.239.948
Hibernate ORM	720.095

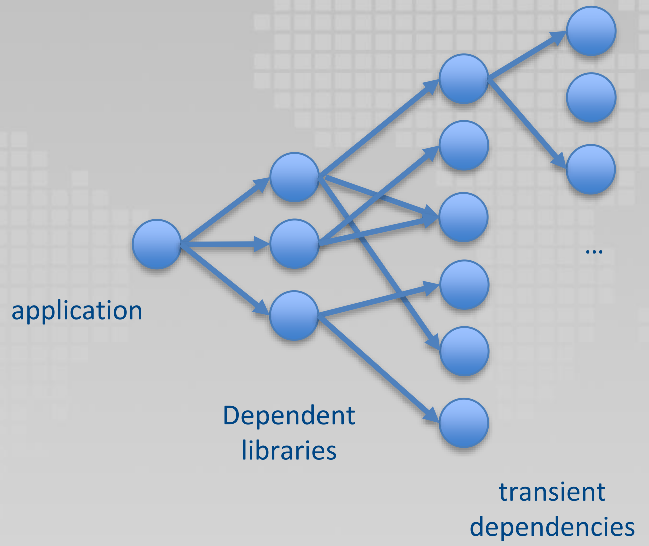


* The Economics of Software Quality By Capers Jones, Olivier Bonsignour

To say it clearly!

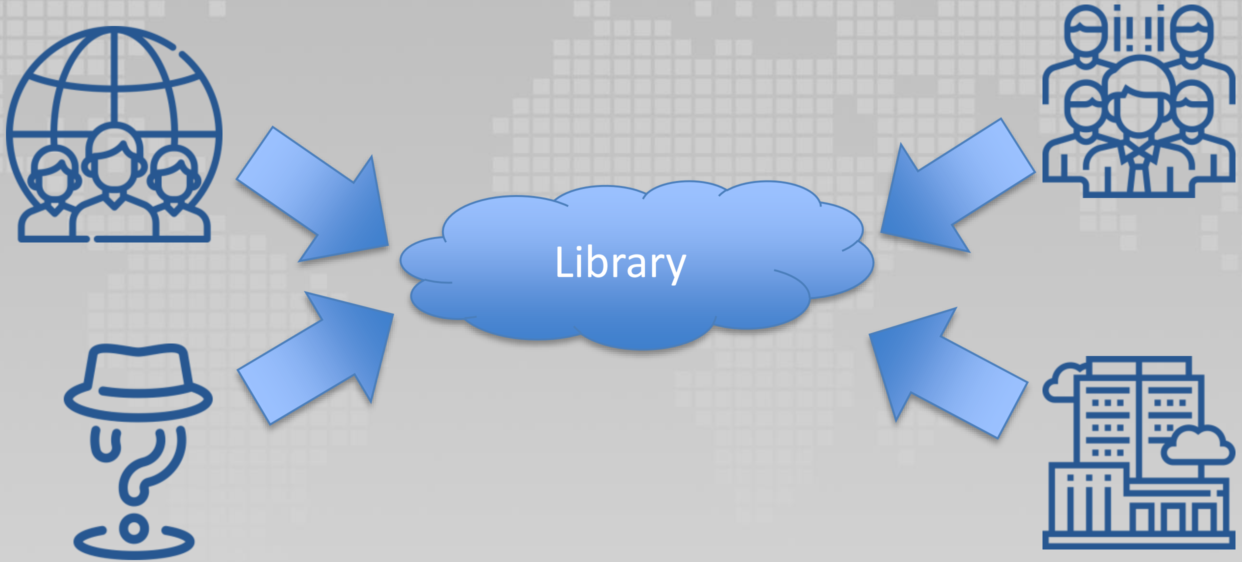
This is no problem of open source!

Introduction
Which libraries am I using?



Inspired by: https://blog.sonatype.com/2011/10/evaluate-open-source-components-before-use-open-source-development-tip-5/

Who is building OSS libraries?



Library

Icons from https://www.flaticon.com/authors/freepik

Why should you trust your libs?

- Our open source team approved them
- They‘re compiled!
- We control our software?
- Open source? Many eyes?
- We pentest?
- We patch?
- Static analysis?

Source: https://www.owasp.org/images/7/70/ASDC12-The_Unfortunate_Reality_of_Insecure_Libraries.pdf

OWASP Top 10

- 2010 → Security Misconfiguration → Position 6
- 2013 → Split of Security Misconfiguration
  → Position 5 (Security Misconfiguration)
  → Position 9 (Using Components with Known Vulnerabilities)
- 2017 → Using Components with Known Vulnerabilities → Position 9

- Examples
  → **Apache CXF (CVE-2012-3451) → Bypass authentication**
  → **Commons collections, ver. 3.2.1 (CVE-2015-6420) → Execute arbitrary code**

How do you get aware of problems with libraries?

How to solve this problem?

How to solve this problem?

Security Testing!

- OWASP Dependency Check
  Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities. Currently Java and .NET are supported; additional experimental support has been added for Ruby, Node.js, Python, and limited support for C/C++ build systems (autoconf and cmake).

- Sonatype Nexus Health Check
  A fully integrated health check for all components within a repository

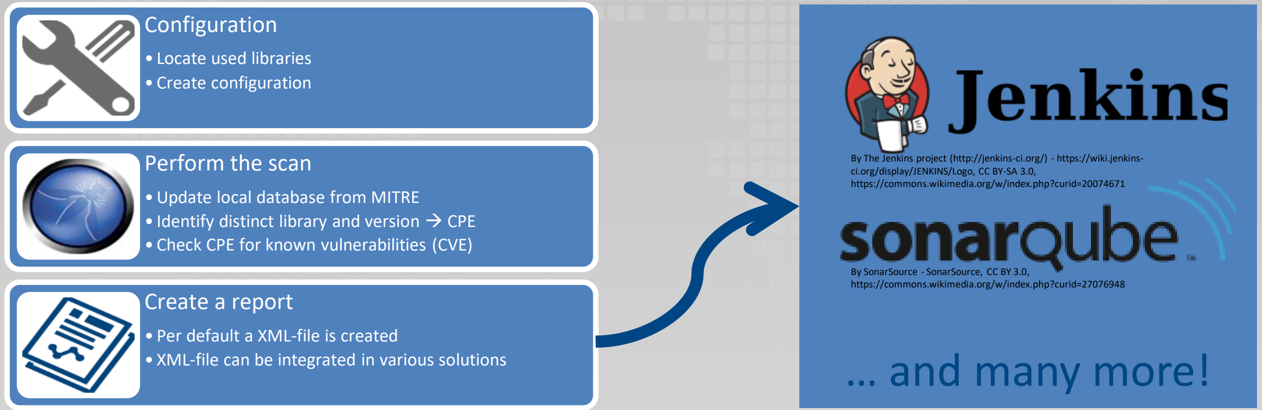
- Artifactory
  Integration with external service (Blackduck) possible.

- …

Commercial solutions

How to solve this problem? OWASP Dependency Check

Commandline tool to check libraries for security issues



Configuration

- Locate used libraries
- Create configuration

Perform the scan

- Update local database from MITRE
- Identify distinct library and version → CPE
- Check CPE for known vulnerabilities (CVE)

Create a report

- Per default a XML-file is created
- XML-file can be integrated in various solutions

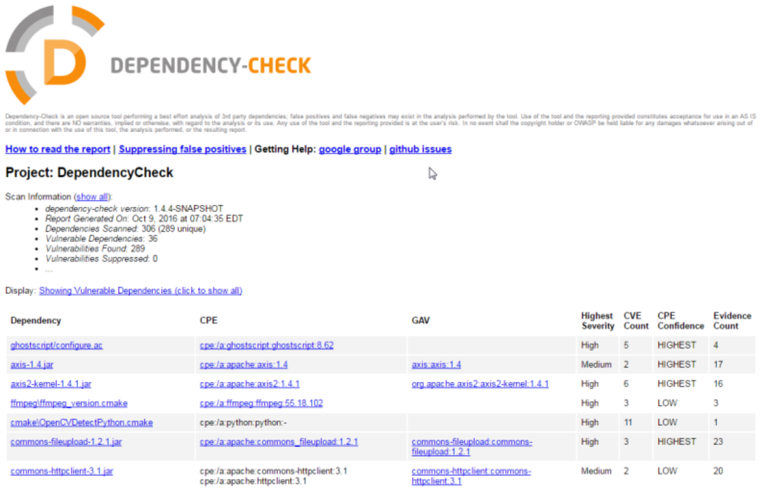
By The Jenkins project (http://jenkins-ci.org/) - https://wiki.jenkins-ci.org/display/JENKINS/Logo, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=20074671

By SonarSource - SonarSource, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=27076948

… and many more!

How to solve this problem?

OWASP Dependency Check



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: google group | github issues

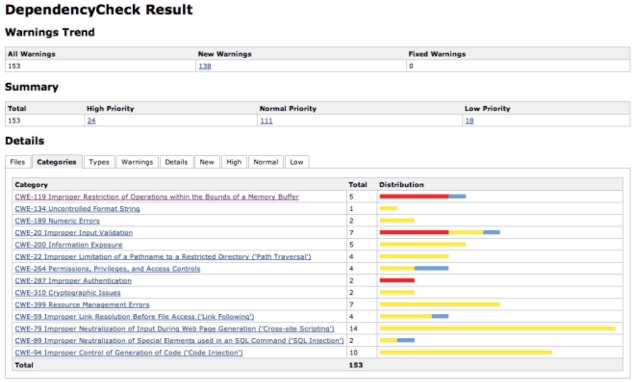
Project: DependencyCheck

Scan Information (show all):

- *dependency-check version*: 1.4.4-SNAPSHOT
- *Report Generated On*: Oct 9, 2016 at 07:04:35 EDT
- *Dependencies Scanned*: 306 (289 unique)
- *Vulnerable Dependencies*: 36
- *Vulnerabilities Found*: 289
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
ghostscript/configure.ac	cpe:/a:ghostscript:ghostscript:8.62		High	5	HIGHEST	4
axis-1.4.jar	cpe:/a:apache:axis:1.4	axis:axis:1.4	Medium	2	HIGHEST	17
axis2-kernel-1.4.1.jar	cpe:/a:apache:axis2:1.4.1	org.apache.axis2:axis2-kernel:1.4.1	High	6	HIGHEST	16
ffmpeg\ffmpeg_version.cmake	cpe:/a:ffmpeg:ffmpeg:55.18.102		High	3	LOW	3
cmake\OpenCVDetectPython.cmake	cpe:/a:python:python:-		High	11	LOW	1
commons-fileupload-1.2.1.jar	cpe:/a:apache:commons_fileupload:1.2.1	commons-fileupload:commons-fileupload:1.2.1	High	3	HIGHEST	23
commons-httpclient-3.1.jar	cpe:/a:apache:commons-httpclient:3.1 cpe:/a:apache:httpclient:3.1	commons-httpclient:commons-httpclient:3.1	Medium	2	LOW	20



DependencyCheck Result

Warnings Trend

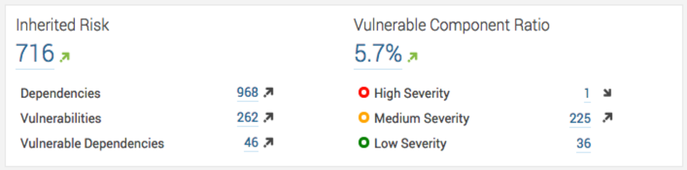
All Warnings	New Warnings	Fixed Warnings
153	138	0

Summary

Total	High Priority	Normal Priority	Low Priority
153	24	111	18

Details

Category	Total	Distribution
CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer	5	
CWE-134 Uncontrolled Format String	1	
CWE-189 Numeric Errors	2	
CWE-20 Improper Input Validation	7	
CWE-200 Information Exposure	5	
CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	4	
CWE-264 Permissions, Privileges, and Access Controls	4	
CWE-287 Improper Authentication	2	
CWE-310 Cryptographic Issues	2	
CWE-399 Resource Management Errors	7	
CWE-59 Improper Link Resolution Before File Access ('Link Following')	4	
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	14	
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	2	
CWE-94 Improper Control of Generation of Code ('Code Injection')	10	
Total	**153**	



Inherited Risk

716 ↗

Dependencies	968	↗
Vulnerabilities	262	↗
Vulnerable Dependencies	46	↗

Vulnerable Component Ratio

5.7% ↗

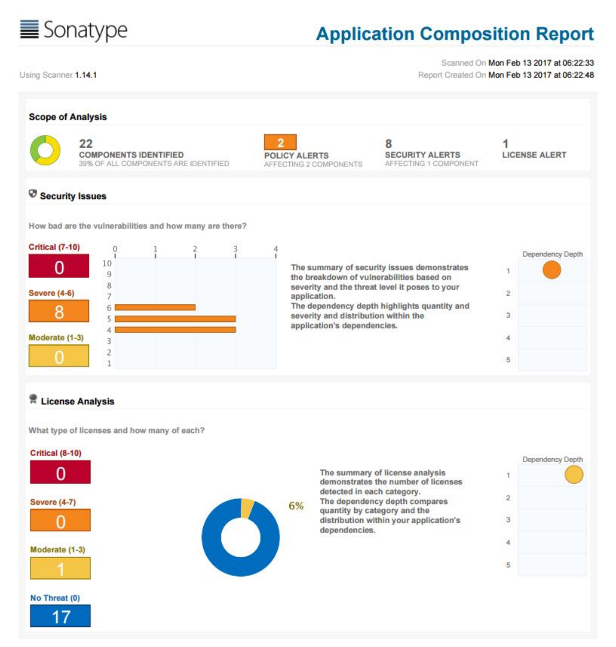
○ High Severity	1	↘
○ Medium Severity	225	↗
○ Low Severity	36	

How to solve this problem?

Sonatype Nexus

Technical aspects:

- Only available in Nexus PRO (commercial)
- Additional information: license analysis
- Nice reporting



Sonatype

Application Composition Report

Scanned On Mon Feb 13 2017 at 06:22:33
Report Created On Mon Feb 13 2017 at 06:22:48

Using Scanner 1.14.1

Scope of Analysis

22 COMPONENTS IDENTIFIED
39% OF ALL COMPONENTS ARE IDENTIFIED

2 POLICY ALERTS
AFFECTING 2 COMPONENTS

8 SECURITY ALERTS
AFFECTING 1 COMPONENT

1 LICENSE ALERT

Security Issues

How bad are the vulnerabilities and how many are there?

Critical (7-10)
0

Severe (4-6)
8

Moderate (1-3)
0

The summary of security issues demonstrates the breakdown of vulnerabilities based on severity and the threat level it poses to your application.
The dependency depth highlights quantity and severity and distribution within the application's dependencies.

License Analysis

What type of licenses and how many of each?

Critical (8-10)
0

Severe (4-7)
0

Moderate (1-3)
1

No Threat (0)
17

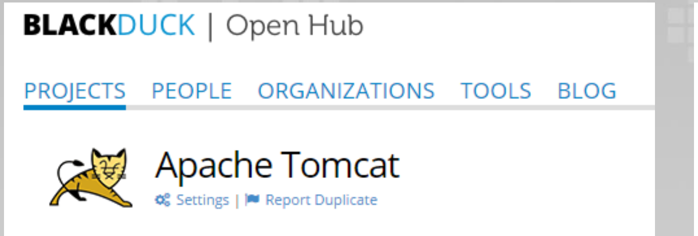
The summary of license analysis demonstrates the number of licenses detected in each category.
The dependency depth compares quantity by category and the distribution within your application's dependencies.

How to get better?

Choose wisely!

How to get better?

Which dependency is more secure?

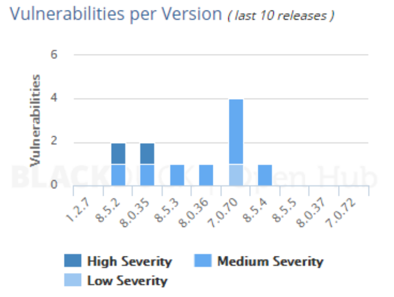


BLACKDUCK | Open Hub

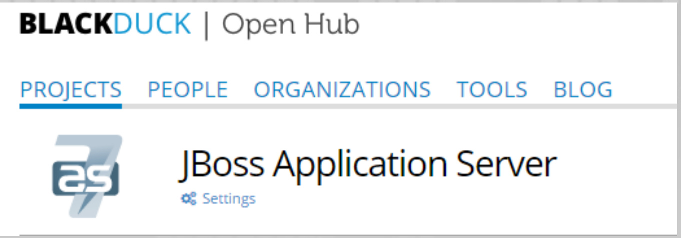
PROJECTS PEOPLE ORGANIZATIONS TOOLS BLOG

Apache Tomcat

Settings | Report Duplicate



Vulnerabilities per Version *(last 10 releases)*

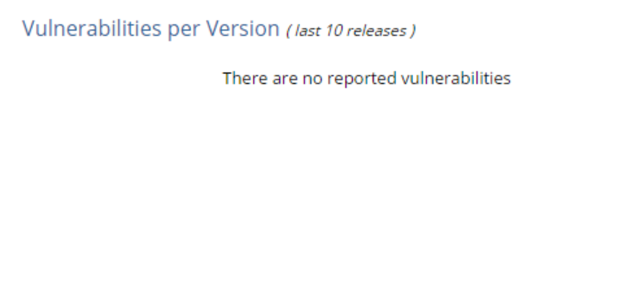


BLACKDUCK | Open Hub

PROJECTS PEOPLE ORGANIZATIONS TOOLS BLOG

JBoss Application Server

Settings



Vulnerabilities per Version *(last 10 releases)*

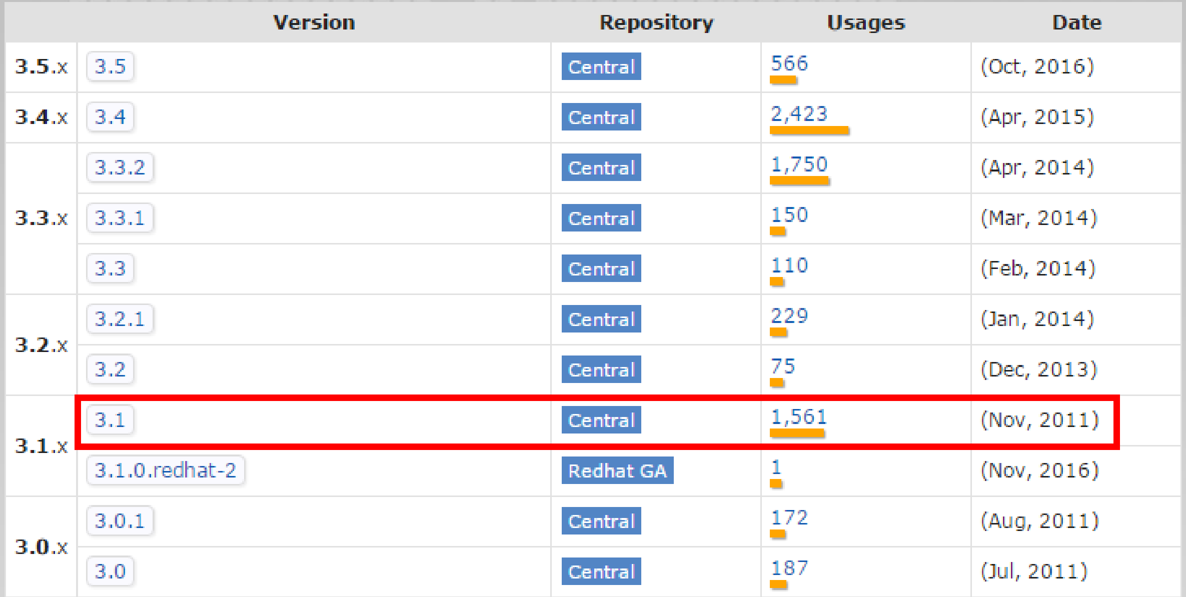
There are no reported vulnerabilities

Stay critical!

How to solve this problem? Monitoring

Apache Commons Lang

Apache Commons Lang, a package of Java utility classes for the classes that are in java.lang's hierarchy, or are considered to be so standard as to justify existence in java.lang.



Version		Repository	Usages	Date
3.5.x	3.5	Central	566	(Oct, 2016)
3.4.x	3.4	Central	2,423	(Apr, 2015)
3.3.x	3.3.2	Central	1,750	(Apr, 2014)
	3.3.1	Central	150	(Mar, 2014)
	3.3	Central	110	(Feb, 2014)
3.2.x	3.2.1	Central	229	(Jan, 2014)
	3.2	Central	75	(Dec, 2013)
3.1.x	3.1	Central	1,561	(Nov, 2011)
	3.1.0.redhat-2	Redhat GA	1	(Nov, 2016)
3.0.x	3.0.1	Central	172	(Aug, 2011)
	3.0	Central	187	(Jul, 2011)

How to get better?

Action plan

Immediate: Inventory

- Scan for libraries
- Create tracking sheet

Short term: analyze

- Purge unnecessary libraries
- Code review to check necessity
- Check signatures

Tactical: Control

- Centralize library control
- If possible: consider sanboxing

Monitor

- Manage your libraries
- Get security intelligence

Source: https://www.owasp.org/images/7/70/ASDC12-The_Unfortunate_Reality_of_Insecure_Libraries.pdf

Thank you!

Happy to discuss with you!

OWASP
Open Web Application
Security Project

Excurse Automation

Automated checking for vulnerable components:

https://dependabot.com/

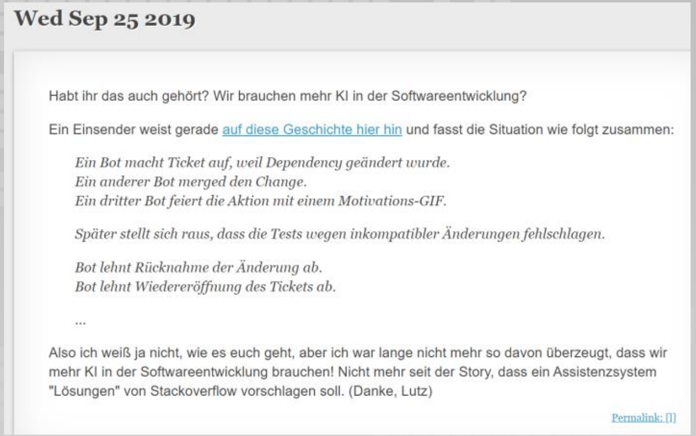
(free, integrated in GitHub)

Automated merging after unit testing:

https://github.com/marketplace/mergify

(commercial solution, simple to script)

Based on http://blog.fefe.de/?ts=a3756d7c



Wed Sep 25 2019

Habt ihr das auch gehört? Wir brauchen mehr KI in der Softwareentwicklung?

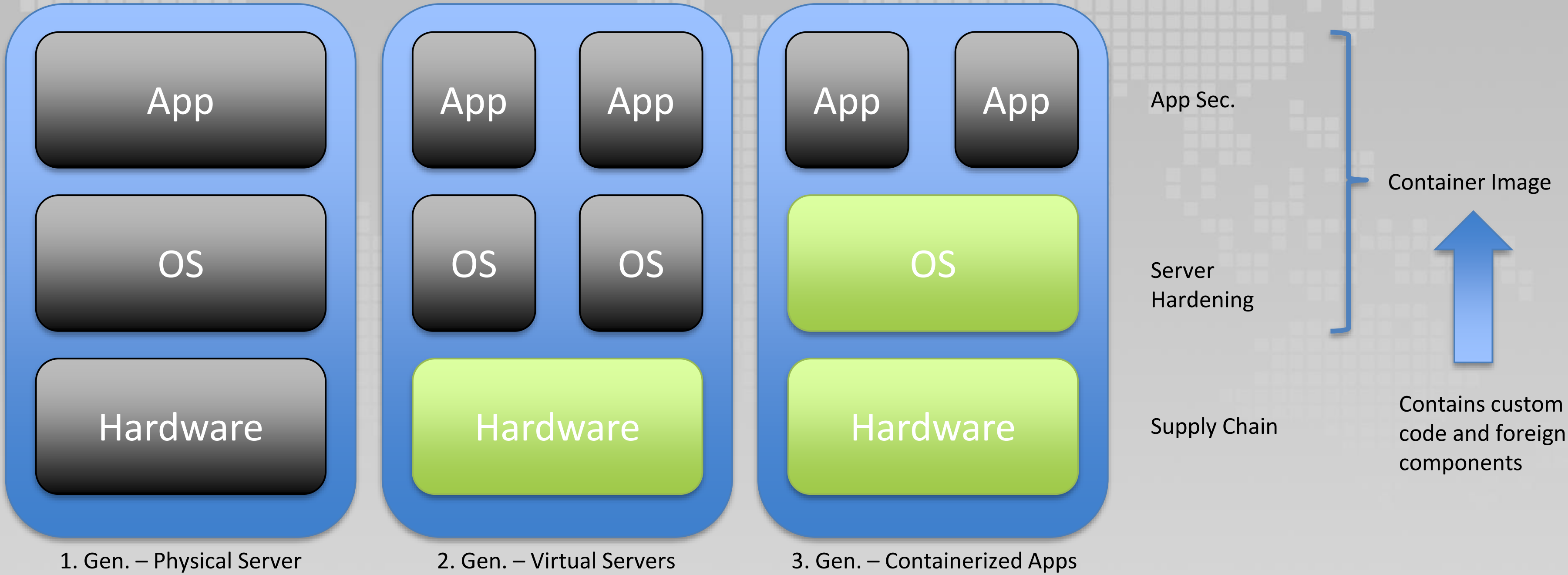
Ein Einsender weist gerade auf diese Geschichte hier hin und fasst die Situation wie folgt zusammen:

> *Ein Bot macht Ticket auf, weil Dependency geändert wurde.*
> *Ein anderer Bot merged den Change.*
> *Ein dritter Bot feiert die Aktion mit einem Motivations-GIF.*
>
> *Später stellt sich raus, dass die Tests wegen inkompatibler Änderungen fehlschlagen.*
>
> *Bot lehnt Rücknahme der Änderung ab.*
> *Bot lehnt Wiedereröffnung des Tickets ab.*
>
> ...

Also ich weiß ja nicht, wie es euch geht, aber ich war lange nicht mehr so davon überzeugt, dass wir mehr KI in der Softwareentwicklung brauchen! Nicht mehr seit der Story, dass ein Assistenzsystem "Lösungen" von Stackoverflow vorschlagen soll. (Danke, Lutz)

Permalink: [l]

Excursion Container (Docker)



1. Gen. – Physical Server

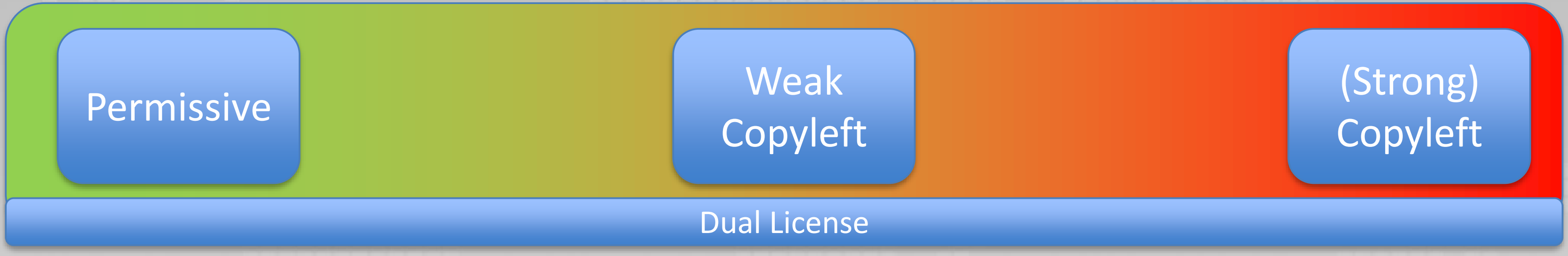
2. Gen. – Virtual Servers

3. Gen. – Containerized Apps

Contains custom code and foreign components

Excursion License

Which license suits best for my commercial project?



Permissive

Weak Copyleft

(Strong) Copyleft

Dual License

Examples:
- MIT
- BSD
- Apache v2

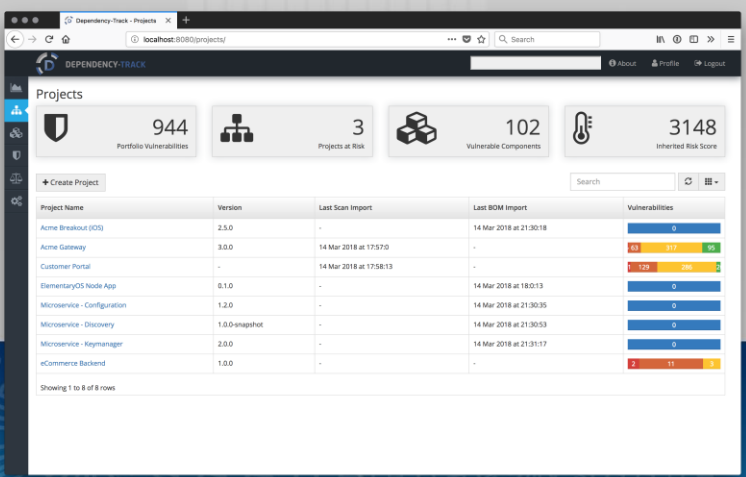
Examples:
- LGPL 2/3

Examples:
- GPL 2/3
- AGPLv3

Web hint: https://choosealicense.com/

OWASP Dependency Track

- Project by Steve Springett
- Allows centralized tracking of dependency vulnerabilities
- Utilizes SBoM file (CycloneDX)



Projects

944 Portfolio Vulnerabilities

3 Projects at Risk

102 Vulnerable Components

3148 Inherited Risk Score

Project Name	Version	Last Scan Import	Last BOM Import	Vulnerabilities
Acme Breakout (iOS)	2.5.0	-	14 Mar 2018 at 21:30:18	0
Acme Gateway	3.0.0	14 Mar 2018 at 17:57:0	-	63 317 95
Customer Portal	-	14 Mar 2018 at 17:58:13	-	1 129 286 2
ElementaryOS Node App	0.1.0	-	14 Mar 2018 at 18:0:13	0
Microservice - Configuration	1.2.0	-	14 Mar 2018 at 21:30:35	0
Microservice - Discovery	1.0.0-snapshot	-	14 Mar 2018 at 21:30:53	0
Microservice - Keymanager	2.0.0	-	14 Mar 2018 at 21:31:17	0
eCommerce Backend	1.0.0	-	-	2 11 3

Showing 1 to 8 of 8 rows

DEPENDENCY-TRACK

https://dependencytrack.org/

CycloneDX

https://cyclonedx.org/