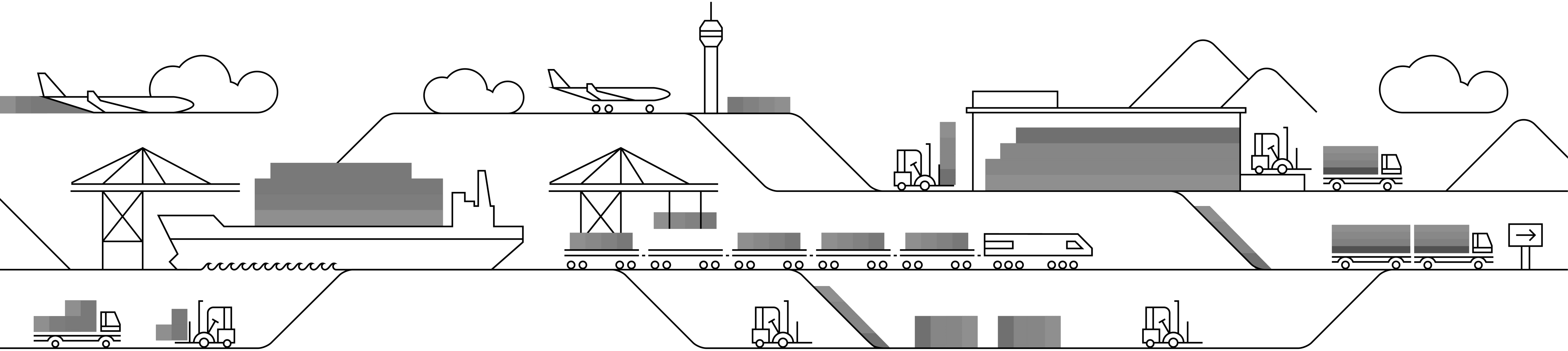


Vulnerability Assessment Tool : Vulas

SAP Security Research

Serena Elisa Ponta
October 17th, 2018

PUBLIC

Run Simple

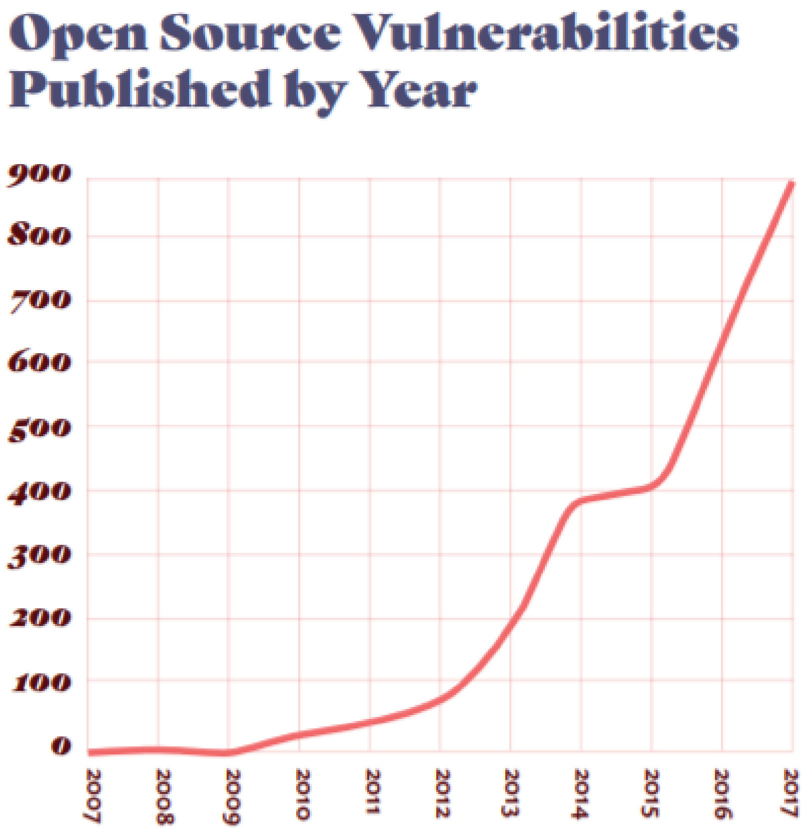
Motivation

Motivation

Vulnerable OSS components

- 80% to 90% of software products on the market include OSS component

- Number of vulnerabilities disclosed for OSS libraries steadily increasing since 2009



(“The State of Open Source Security”, Snyk, 2017)

Motivation

Using components with known vulnerabilities:

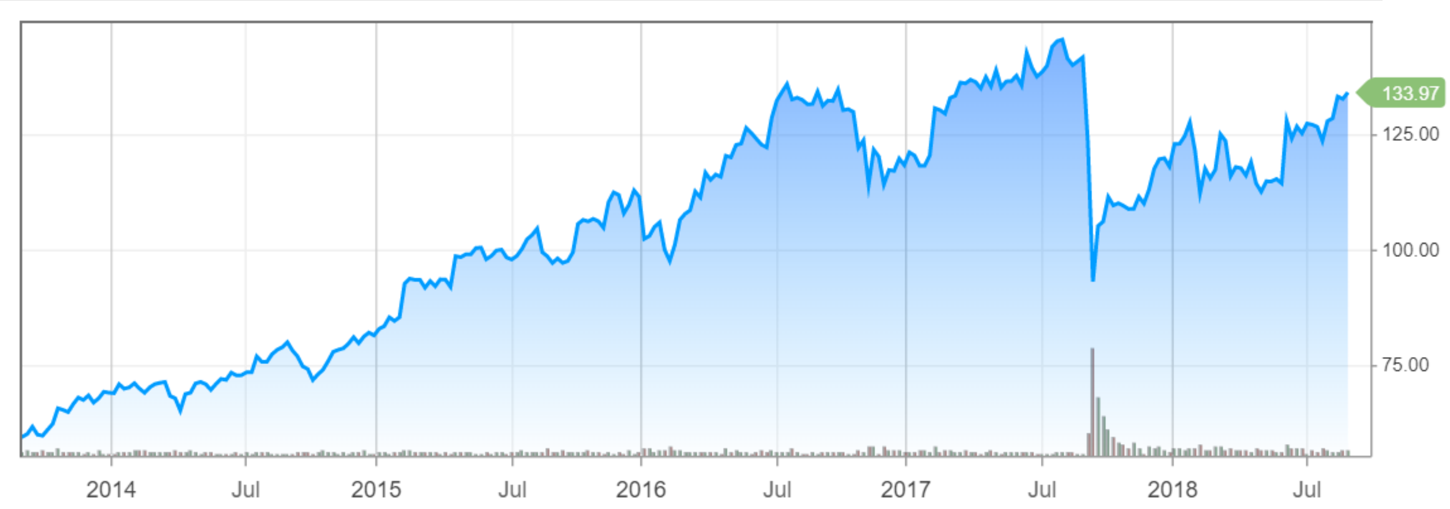
- ❑ Included in **OWASP Top 10** (2013-2017): A9
- ❑ Root cause of 12 of the top 50 data breaches in 2016

➢ Mossack Fonseca (***Panama Pa***

➢ ***Equifax*** breach

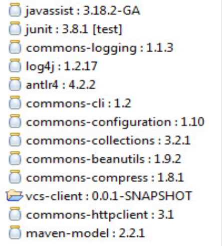
Consequences can be:

- Financial
- Loss of customers’ trust
- Loss of image/reputation



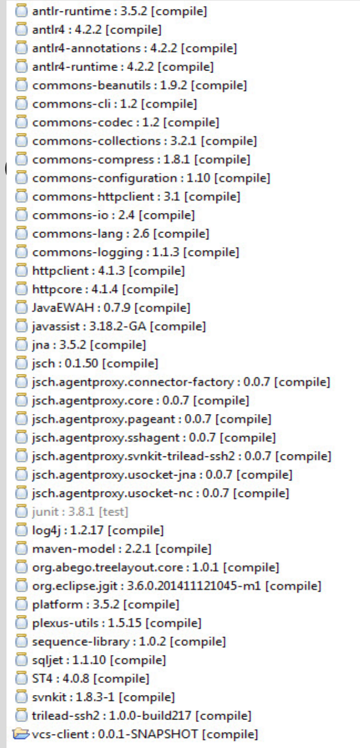
Just Update?

Perhaps during development, but **transitive** dependenci sidered and may be **unknown** or **not understood**



```
javassist : 3.18.2-GA
junit : 3.8.1 [test]
commons-logging : 1.1.3
log4j : 1.2.17
antlr4 : 4.2.2
commons-cli : 1.2
commons-configuration : 1.10
commons-collections : 3.2.1
commons-beanutils : 1.9.2
commons-compress : 1.8.1
vcs-client : 0.0.1-SNAPSHOT
commons-httpclient : 3.1
maven-model : 2.2.1
```

13 to 40



```
antlr-runtime : 3.5.2 [compile]
antlr4 : 4.2.2 [compile]
antlr4-annotations : 4.2.2 [compile]
antlr4-runtime : 4.2.2 [compile]
commons-beanutils : 1.9.2 [compile]
commons-cli : 1.2 [compile]
commons-codec : 1.2 [compile]
commons-collections : 3.2.1 [compile]
commons-compress : 1.8.1 [compile]
commons-configuration : 1.10 [compile]
commons-httpclient : 3.1 [compile]
commons-io : 2.4 [compile]
commons-lang : 2.6 [compile]
commons-logging : 1.1.3 [compile]
httpclient : 4.1.3 [compile]
httpcore : 4.1.4 [compile]
JavaEWAH : 0.7.9 [compile]
javassist : 3.18.2-GA [compile]
jna : 3.5.2 [compile]
jsch : 0.1.50 [compile]
jsch.agentproxy.connector-factory : 0.0.7 [compile]
jsch.agentproxy.core : 0.0.7 [compile]
jsch.agentproxy.pageant : 0.0.7 [compile]
jsch.agentproxy.sshagent : 0.0.7 [compile]
jsch.agentproxy.svnkit-trilead-ssh2 : 0.0.7 [compile]
jsch.agentproxy.usocket-jna : 0.0.7 [compile]
jsch.agentproxy.usocket-nc : 0.0.7 [compile]
junit : 3.8.1 [test]
log4j : 1.2.17 [compile]
maven-model : 2.2.1 [compile]
org.abego.treelayout.core : 1.0.1 [compile]
org.eclipse.jgit : 3.6.0.201411121045-m1 [compile]
platform : 3.5.2 [compile]
plexus-utils : 1.5.15 [compile]
sequence-library : 1.0.2 [compile]
sqljet : 1.1.10 [compile]
ST4 : 4.0.8 [compile]
svnkit : 1.8.3-1 [compile]
trilead-ssh2 : 1.0.0-build217 [compile]
vcs-client : 0.0.1-SNAPSHOT [compile]
```

Unrealistic for **live** applications

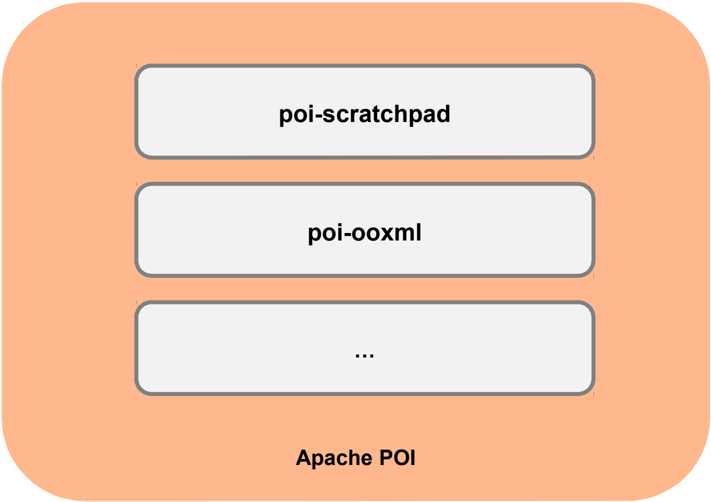
Impact analysis is difficult

Vulnerability description (in natural language) often not useful

***CVE-2012-5633**: "The URIMappingInterceptor in Apache CXF before 2.5.8, 2.6.x before 2.6.5, and 2.7.x before 2.7.2, when using the WSS4JInInterceptor, bypasses WS-Security processing, which allows remote attackers to obtain access to SOAP services via an HTTP GET request."*

Impact analysis is difficult

Vulnerabilities are assigned to entire projects (e.g., Apache POI, Tomcat), sub-components (e.g. Jar archives) are used separately



Existing approaches
based on meta-data

- Most tools “somehow” map finer-grained OSS components (e.g., JAR archives) to vulnerabilities using the project metadata
- **Actual code is ignored**

Limitations:

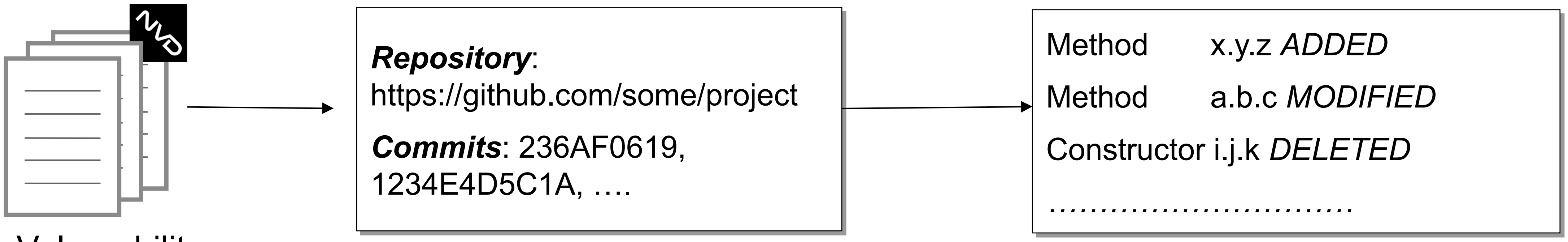
- False-positives (e.g., multi-module projects)
- False-negatives (e.g., re-bundling)
- Focus only on detection (no app-specific analysis)

Contributions

- From vulnerability to **vulnerable constructs** (actual code)
- Code-centric **detection** of known vulnerabilities
- Static and Dynamic **assessment** of vulnerable code
- Metrics to support selection of non-vulnerable libraries (**mitigation**)

Approach

Vulnerable Constructs



Vulnerability advisory

(CVE, GitHub issue)

Repository: https://github.com/some/project

Commits: 236AF0619, 1234E4D5C1A, ….

Patch Information

Method x.y.z *ADDED*

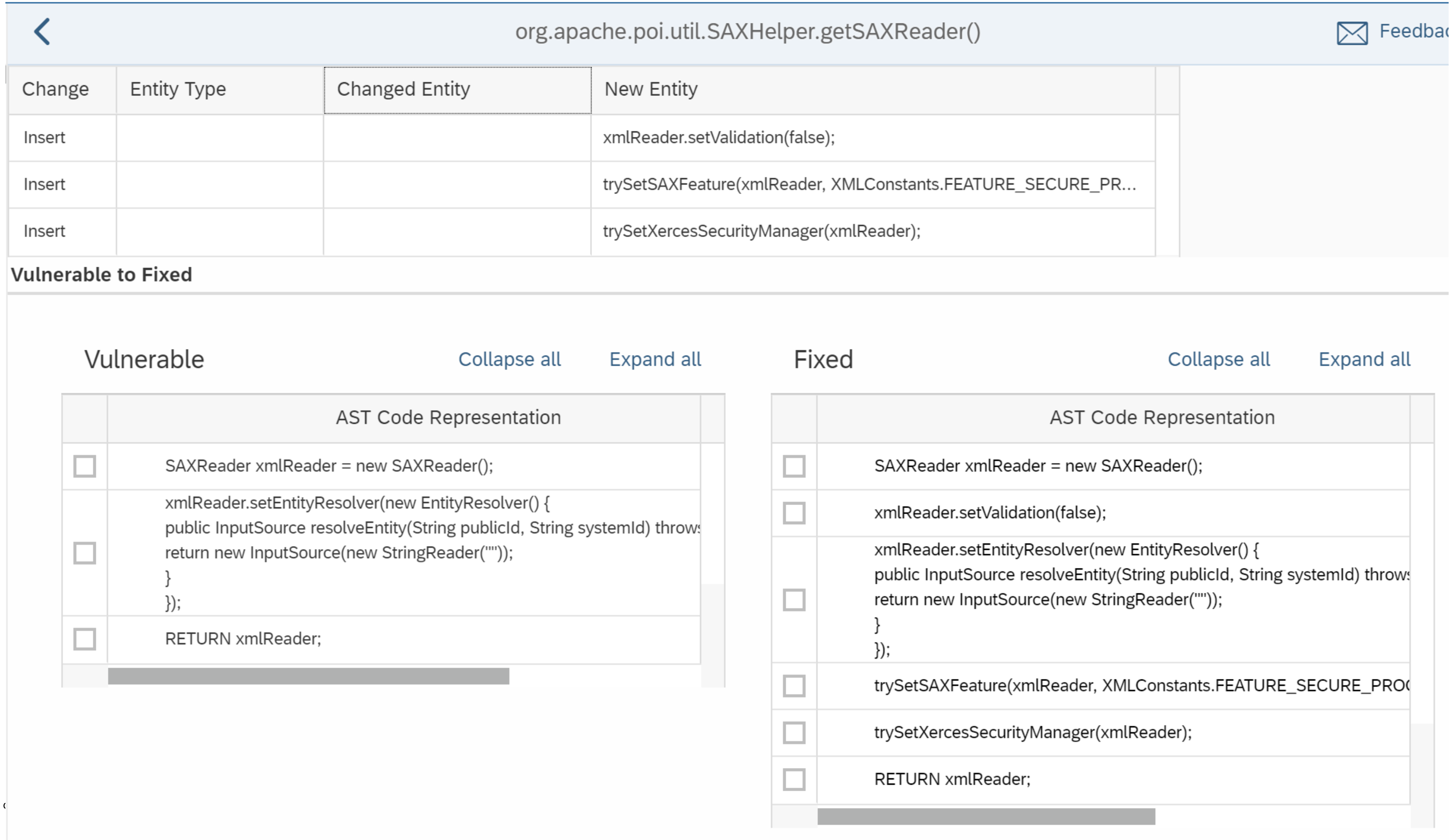
Method a.b.c *MODIFIED*

Constructor i.j.k *DELETED*

……………………………

Vulnerable Constructs

Vulnerable Constructs (CVE-2014-3574)



org.apache.poi.util.SAXHelper.getSAXReader()

Change	Entity Type	Changed Entity	New Entity
Insert			xmlReader.setValidation(false);
Insert			trySetSAXFeature(xmlReader, XMLConstants.FEATURE_SECURE_PR...
Insert			trySetXercesSecurityManager(xmlReader);

Vulnerable to Fixed

Vulnerable

	AST Code Representation
□	SAXReader xmlReader = new SAXReader();

□	xmlReader.setEntityResolver(new EntityResolver() {
	public InputSource resolveEntity(String publicId, String systemId) throws
	return new InputSource(new StringReader(""));
	}
	});

□	RETURN xmlReader;

Fixed

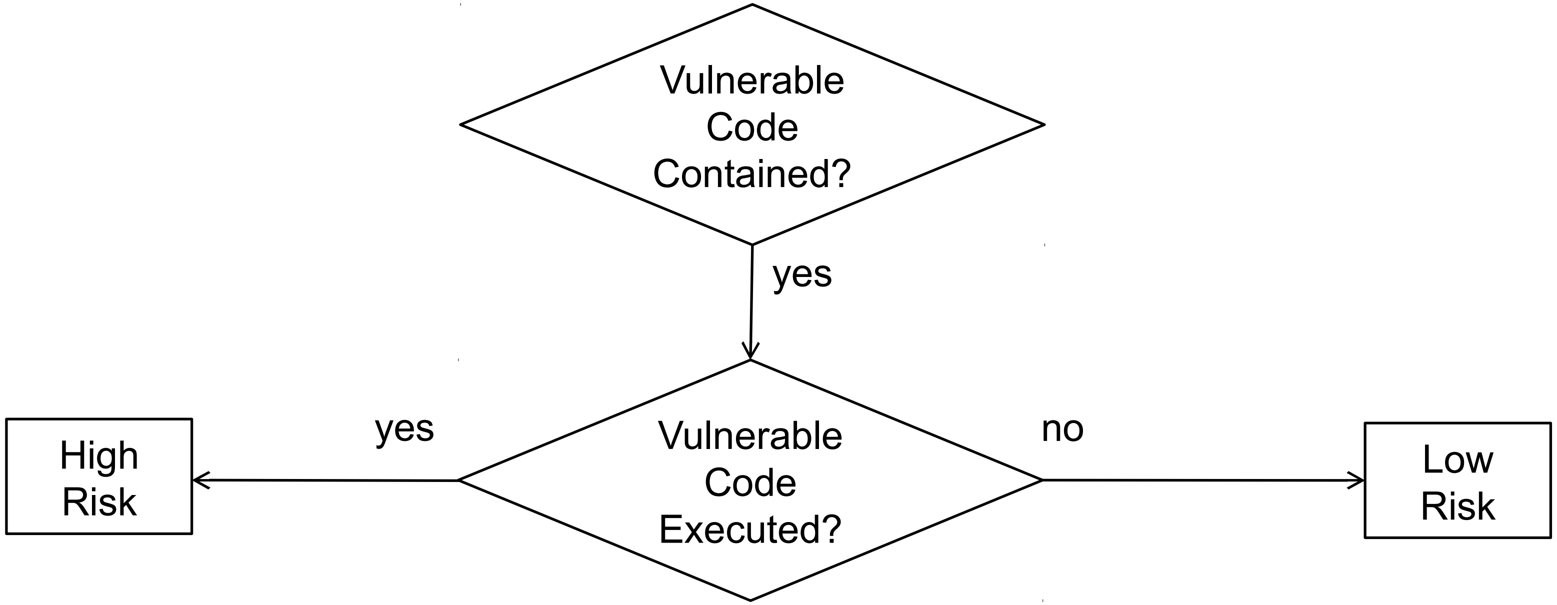
	AST Code Representation
□	SAXReader xmlReader = new SAXReader();
□	xmlReader.setValidation(false);

□	xmlReader.setEntityResolver(new EntityResolver() {
	public InputSource resolveEntity(String publicId, String systemId) throws
	return new InputSource(new StringReader(""));
	}
	});

□	trySetSAXFeature(xmlReader, XMLConstants.FEATURE_SECURE_PROC
□	trySetXercesSecurityManager(xmlReader);
□	RETURN xmlReader;

Approach

Assumption: If an application contains and executes vulnerable constructs, then there is a significant risk that the vulnerability can be exploited in the application context

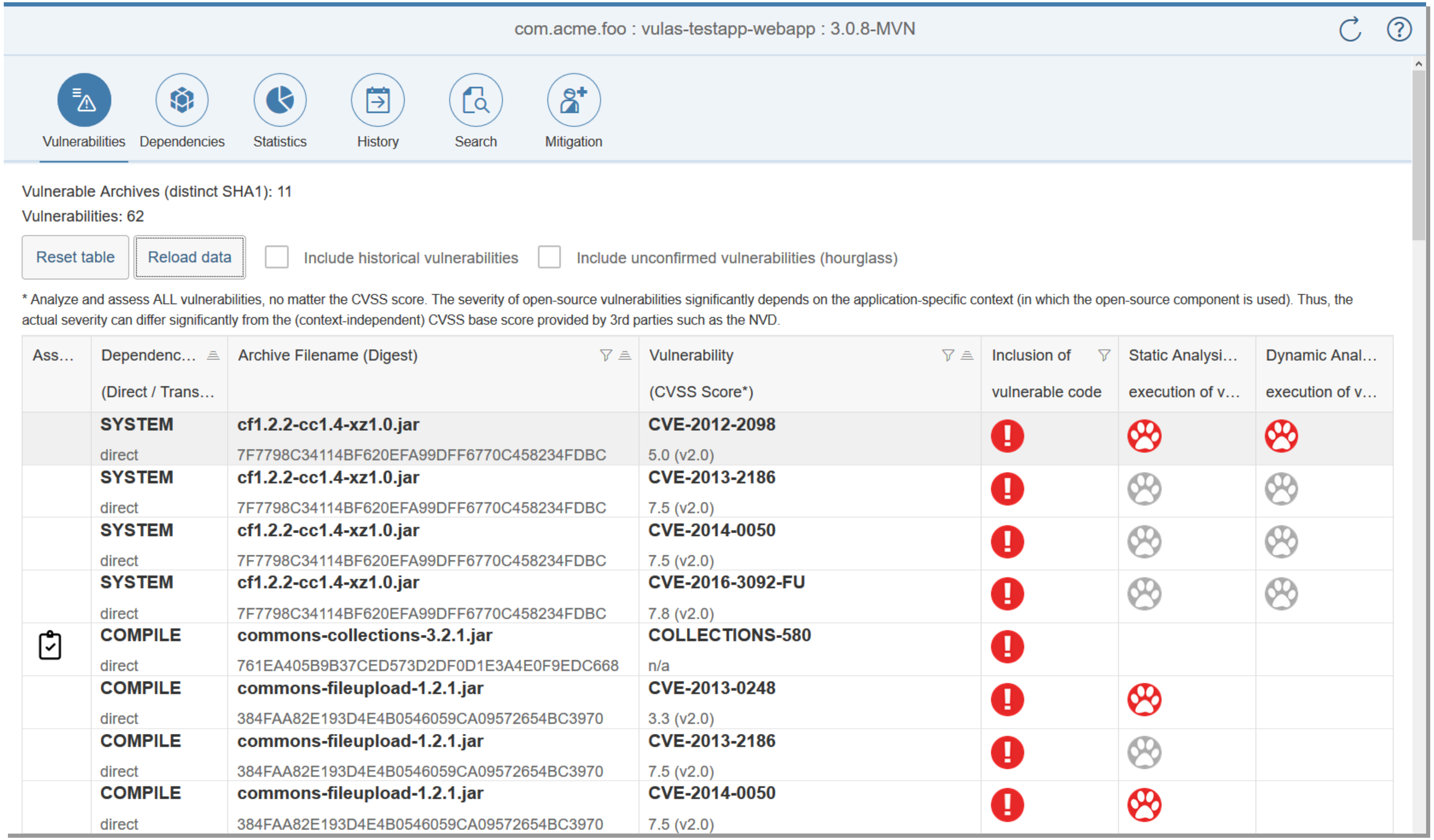


- ❑ Dynamic analysis
- ❑ Static reachability analysis
- ❑ Combination of static and dynamic analysis

Ponta, Plate, Sabetta,
"Beyond Metadata: Code-centric and Usage-based Analysis of Known Vulnerabilities in Open-source Software"
34th IEEE Int Conf on Software Maintenance and Evolution (ICSME), 2018

Demo

Vulnerability Overview



com.acme.foo : vulas-testapp-webapp : 3.0.8-MVN

Vulnerable Archives (distinct SHA1): 11

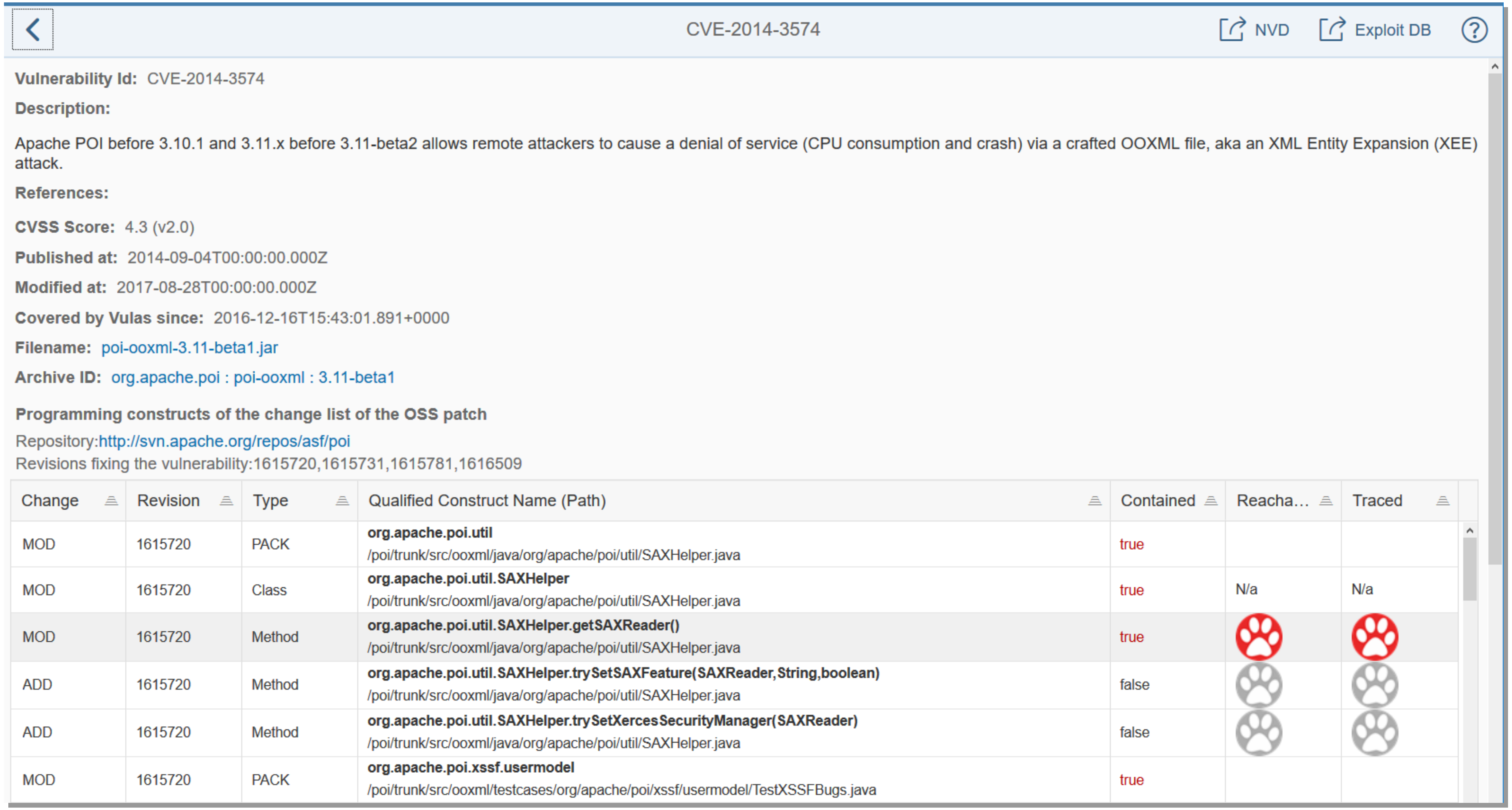
Vulnerabilities: 62

□ Include historical vulnerabilities □ Include unconfirmed vulnerabilities (hourglass)

* Analyze and assess ALL vulnerabilities, no matter the CVSS score. The severity of open-source vulnerabilities significantly depends on the application-specific context (in which the open-source component is used). Thus, the actual severity can differ significantly from the (context-independent) CVSS base score provided by 3rd parties such as the NVD.

Ass…	Dependenc… (Direct / Trans…	Archive Filename (Digest)	Vulnerability (CVSS Score*)	Inclusion of vulnerable code	Static Analysi… execution of v…	Dynamic Anal… execution of v…
	SYSTEM direct	**cf1.2.2-cc1.4-xz1.0.jar** 7F7798C34114BF620EFA99DFF6770C458234FDBC	**CVE-2012-2098** 5.0 (v2.0)	❗	🐾	🐾
	SYSTEM direct	**cf1.2.2-cc1.4-xz1.0.jar** 7F7798C34114BF620EFA99DFF6770C458234FDBC	**CVE-2013-2186** 7.5 (v2.0)	❗	○	○
	SYSTEM direct	**cf1.2.2-cc1.4-xz1.0.jar** 7F7798C34114BF620EFA99DFF6770C458234FDBC	**CVE-2014-0050** 7.5 (v2.0)	❗	○	○
	SYSTEM direct	**cf1.2.2-cc1.4-xz1.0.jar** 7F7798C34114BF620EFA99DFF6770C458234FDBC	**CVE-2016-3092-FU** 7.8 (v2.0)	❗	○	○
📋	**COMPILE** direct	**commons-collections-3.2.1.jar** 761EA405B9B37CED573D2DF0D1E3A4E0F9EDC668	**COLLECTIONS-580** n/a	❗		
	COMPILE direct	**commons-fileupload-1.2.1.jar** 384FAA82E193D4E4B0546059CA09572654BC3970	**CVE-2013-0248** 3.3 (v2.0)	❗	🐾	
	COMPILE direct	**commons-fileupload-1.2.1.jar** 384FAA82E193D4E4B0546059CA09572654BC3970	**CVE-2013-2186** 7.5 (v2.0)	❗	○	
	COMPILE direct	**commons-fileupload-1.2.1.jar** 384FAA82E193D4E4B0546059CA09572654BC3970	**CVE-2014-0050** 7.5 (v2.0)	❗	🐾	

Vulnerability Details



CVE-2014-3574

Vulnerability Id: CVE-2014-3574

Description:

Apache POI before 3.10.1 and 3.11.x before 3.11-beta2 allows remote attackers to cause a denial of service (CPU consumption and crash) via a crafted OOXML file, aka an XML Entity Expansion (XEE) attack.

References:

CVSS Score: 4.3 (v2.0)

Published at: 2014-09-04T00:00:00.000Z

Modified at: 2017-08-28T00:00:00.000Z

Covered by Vulas since: 2016-12-16T15:43:01.891+0000

Filename: poi-ooxml-3.11-beta1.jar

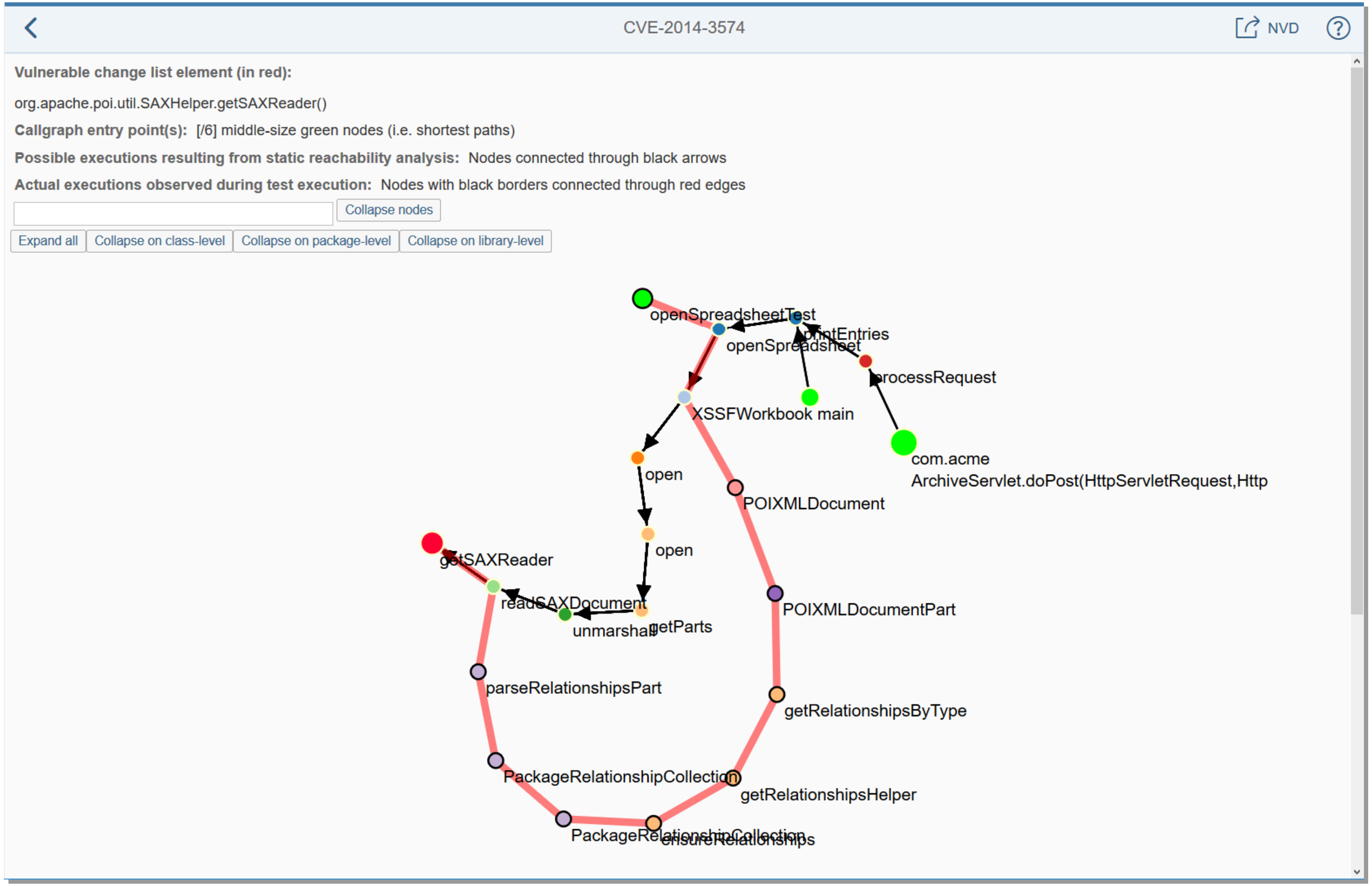
Archive ID: org.apache.poi : poi-ooxml : 3.11-beta1

Programming constructs of the change list of the OSS patch

Repository:http://svn.apache.org/repos/asf/poi
Revisions fixing the vulnerability:1615720,1615731,1615781,1616509

Change	Revision	Type	Qualified Construct Name (Path)	Contained	Reacha...	Traced
MOD	1615720	PACK	**org.apache.poi.util** /poi/trunk/src/ooxml/java/org/apache/poi/util/SAXHelper.java	true		
MOD	1615720	Class	**org.apache.poi.util.SAXHelper** /poi/trunk/src/ooxml/java/org/apache/poi/util/SAXHelper.java	true	N/a	N/a
MOD	1615720	Method	**org.apache.poi.util.SAXHelper.getSAXReader()** /poi/trunk/src/ooxml/java/org/apache/poi/util/SAXHelper.java	true	●	●
ADD	1615720	Method	**org.apache.poi.util.SAXHelper.trySetSAXFeature(SAXReader,String,boolean)** /poi/trunk/src/ooxml/java/org/apache/poi/util/SAXHelper.java	false	○	○
ADD	1615720	Method	**org.apache.poi.util.SAXHelper.trySetXercesSecurityManager(SAXReader)** /poi/trunk/src/ooxml/java/org/apache/poi/util/SAXHelper.java	false	○	○
MOD	1615720	PACK	**org.apache.poi.xssf.usermodel** /poi/trunk/src/ooxml/testcases/org/apache/poi/xssf/usermodel/TestXSSFBugs.java	true		

Vulnerability Details



CVE-2014-3574

Vulnerable change list element (in red):

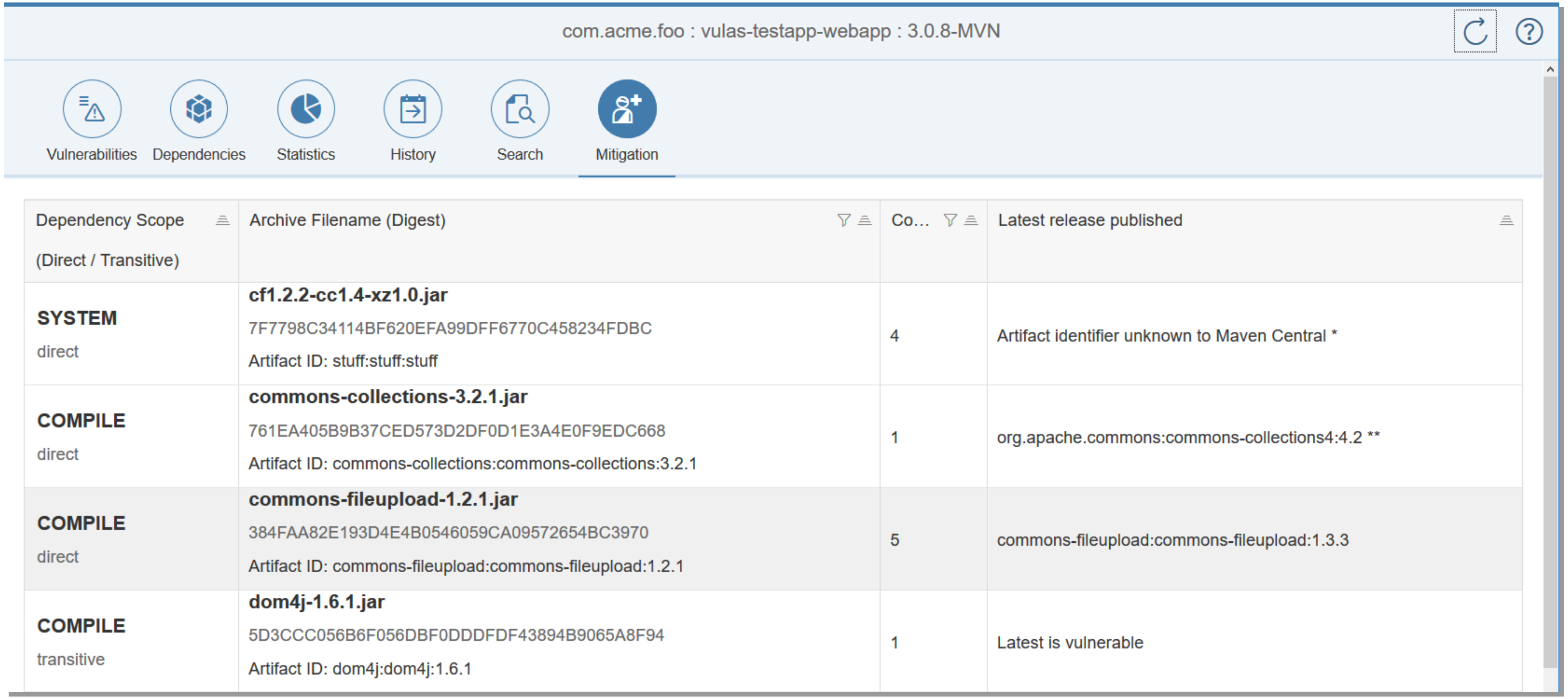
org.apache.poi.util.SAXHelper.getSAXReader()

Callgraph entry point(s): [/6] middle-size green nodes (i.e. shortest paths)

Possible executions resulting from static reachability analysis: Nodes connected through black arrows

Actual executions observed during test execution: Nodes with black borders connected through red edges

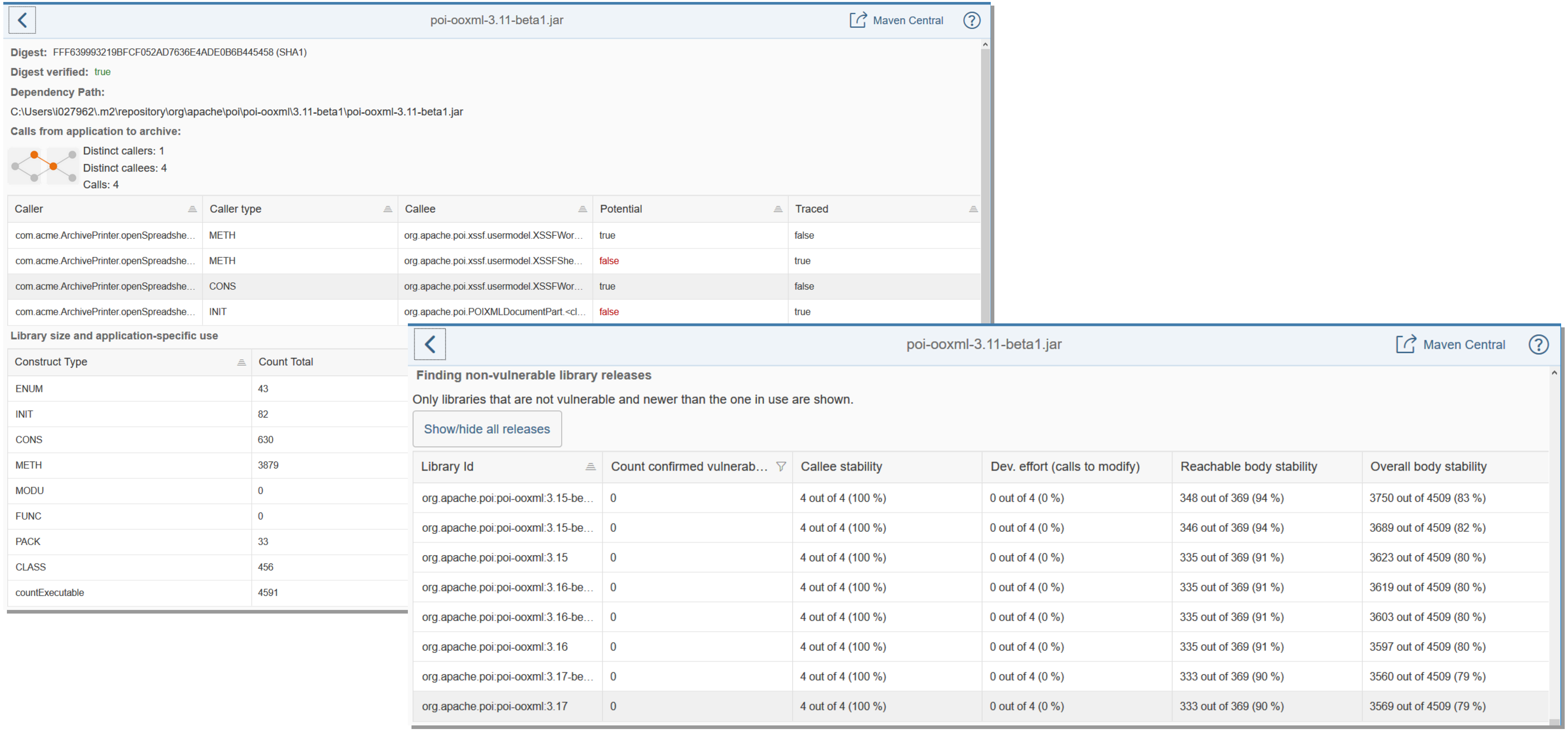
Mitigation Overview



com.acme.foo : vulas-testapp-webapp : 3.0.8-MVN

Dependency Scope (Direct / Transitive)	Archive Filename (Digest)	Co…	Latest release published
SYSTEM direct	**cf1.2.2-cc1.4-xz1.0.jar** 7F7798C34114BF620EFA99DFF6770C458234FDBC Artifact ID: stuff:stuff:stuff	4	Artifact identifier unknown to Maven Central *
COMPILE direct	**commons-collections-3.2.1.jar** 761EA405B9B37CED573D2DF0D1E3A4E0F9EDC668 Artifact ID: commons-collections:commons-collections:3.2.1	1	org.apache.commons:commons-collections4:4.2 **
COMPILE direct	**commons-fileupload-1.2.1.jar** 384FAA82E193D4E4B0546059CA09572654BC3970 Artifact ID: commons-fileupload:commons-fileupload:1.2.1	5	commons-fileupload:commons-fileupload:1.3.3
COMPILE transitive	**dom4j-1.6.1.jar** 5D3CCC056B6F056DBF0DDDFDF43894B9065A8F94 Artifact ID: dom4j:dom4j:1.6.1	1	Latest is vulnerable

Mitigation Details



Digest: FFF639993219BFCF052AD7636E4ADE0B6B445458 (SHA1)

Digest verified: true

Dependency Path:

C:\Users\i027962\.m2\repository\org\apache\poi\poi-ooxml\3.11-beta1\poi-ooxml-3.11-beta1.jar

Calls from application to archive:

Distinct callers: 1
Distinct callees: 4
Calls: 4

Caller	Caller type	Callee	Potential	Traced
com.acme.ArchivePrinter.openSpreadshe...	METH	org.apache.poi.xssf.usermodel.XSSFWor...	true	false
com.acme.ArchivePrinter.openSpreadshe...	METH	org.apache.poi.xssf.usermodel.XSSFShe...	false	true
com.acme.ArchivePrinter.openSpreadshe...	CONS	org.apache.poi.xssf.usermodel.XSSFWor...	true	false
com.acme.ArchivePrinter.openSpreadshe...	INIT	org.apache.poi.POIXMLDocumentPart.<cl...	false	true

Library size and application-specific use

Construct Type	Count Total
ENUM	43
INIT	82
CONS	630
METH	3879
MODU	0
FUNC	0
PACK	33
CLASS	456
countExecutable	4591

Finding non-vulnerable library releases

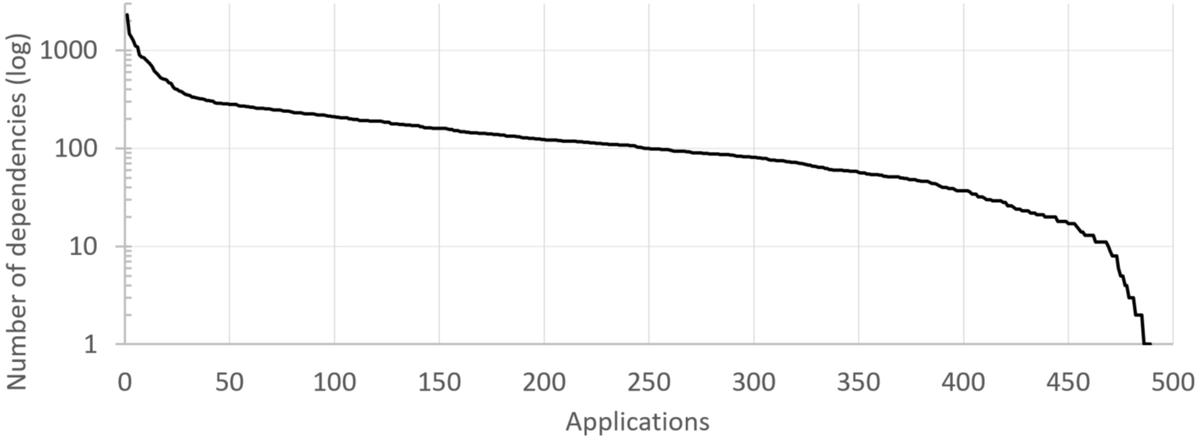
Only libraries that are not vulnerable and newer than the one in use are shown.

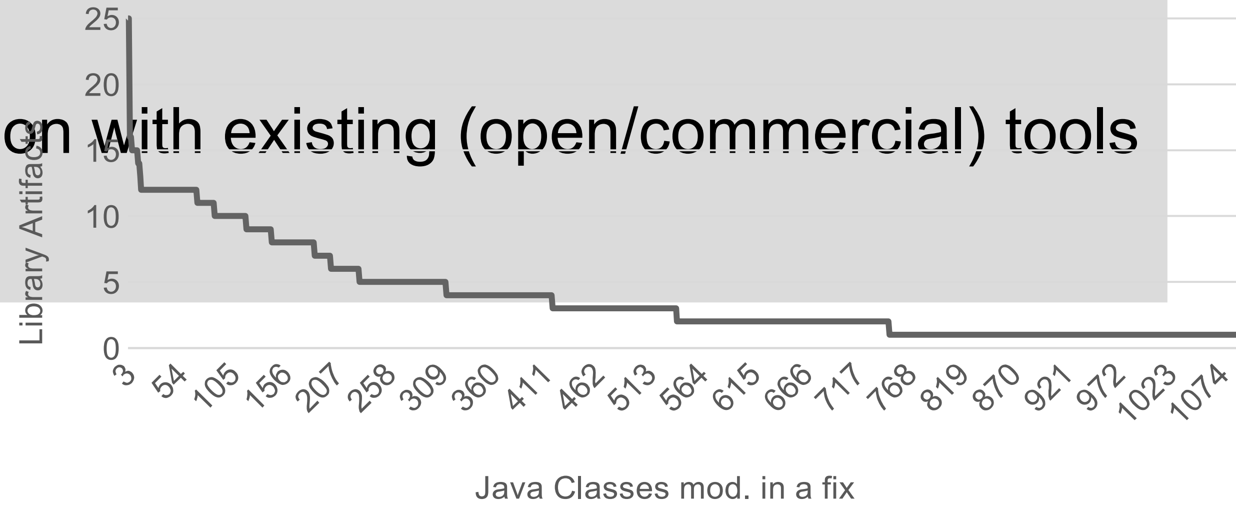
Library Id	Count confirmed vulnerab...	Callee stability	Dev. effort (calls to modify)	Reachable body stability	Overall body stability
org.apache.poi:poi-ooxml:3.15-be...	0	4 out of 4 (100 %)	0 out of 4 (0 %)	348 out of 369 (94 %)	3750 out of 4509 (83 %)
org.apache.poi:poi-ooxml:3.15-be...	0	4 out of 4 (100 %)	0 out of 4 (0 %)	346 out of 369 (94 %)	3689 out of 4509 (82 %)
org.apache.poi:poi-ooxml:3.15	0	4 out of 4 (100 %)	0 out of 4 (0 %)	335 out of 369 (91 %)	3623 out of 4509 (80 %)
org.apache.poi:poi-ooxml:3.16-be...	0	4 out of 4 (100 %)	0 out of 4 (0 %)	335 out of 369 (91 %)	3619 out of 4509 (80 %)
org.apache.poi:poi-ooxml:3.16-be...	0	4 out of 4 (100 %)	0 out of 4 (0 %)	335 out of 369 (91 %)	3603 out of 4509 (80 %)
org.apache.poi:poi-ooxml:3.16	0	4 out of 4 (100 %)	0 out of 4 (0 %)	335 out of 369 (91 %)	3597 out of 4509 (80 %)
org.apache.poi:poi-ooxml:3.17-be...	0	4 out of 4 (100 %)	0 out of 4 (0 %)	333 out of 369 (90 %)	3560 out of 4509 (79 %)
org.apache.poi:poi-ooxml:3.17	0	4 out of 4 (100 %)	0 out of 4 (0 %)	333 out of 369 (90 %)	3569 out of 4509 (79 %)

Experience Report

Evaluation

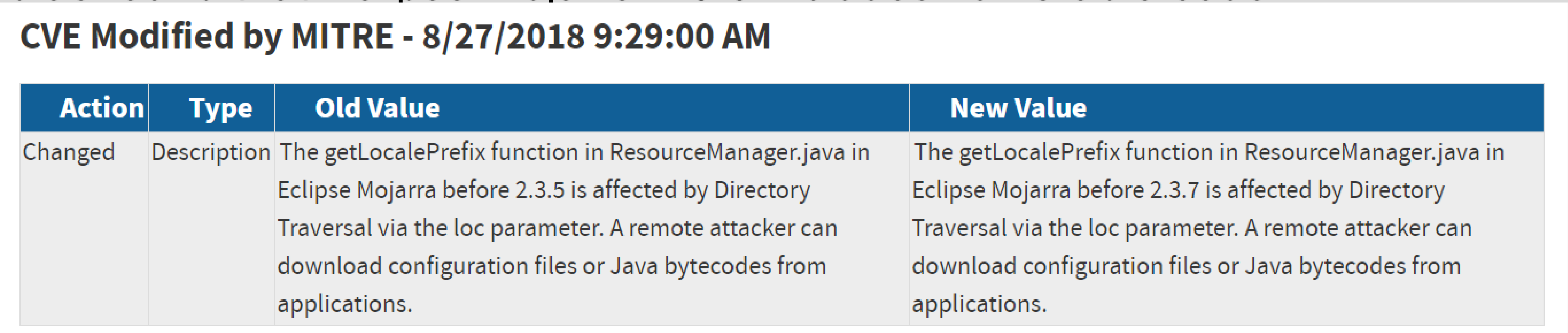
- No downtime since Dec 2016
- 250,000 scans of 500 applications
- Under-development/legacy applications
- arison with existing (open/commercial) tools





Beyond Metadata!

- NVD reported that “Eclipse Mojarra before 2.3.5 is affected by CVE-2018-14371”
- Vulas found that Eclipse Mojarra 2.3.5 includes vulnerable code
- C



CVE Modified by MITRE - 8/27/2018 9:29:00 AM

Action	Type	Old Value	New Value
Changed	Description	The getLocalePrefix function in ResourceManager.java in Eclipse Mojarra before 2.3.5 is affected by Directory Traversal via the loc parameter. A remote attacker can download configuration files or Java bytecodes from applications.	The getLocalePrefix function in ResourceManager.java in Eclipse Mojarra before 2.3.7 is affected by Directory Traversal via the loc parameter. A remote attacker can download configuration files or Java bytecodes from applications.

Vulas Summary

Client-side tools

- Plugins for Maven and Gradle (Java) and setuptools (Python)
- Command Line Interface (CLI) for everything else

Two server-side microservices

- Tenants and workspaces to separate scans
- RESTful interfaces

Enterprise-ready

- Usable in CI/CD pipelines, but also for legacy software
- Aggregated reports and audit of findings
- New vulnerabilities are detected without needing to re-scan
- Support of CERT: Which of our apps are impacted by vulnerability X?
- Non-disclosed (internal) vulnerabilities can also be added to knowledge base

Vulas is Open-source

Vulas is Open-source

Goal: Establish a collaboration among enterprises, universities and Open Source foundations to reduce the risk coming from the use of vulnerable OSS components, e.g.,

- Contributions to the vulnerability knowledge base
- New analysis techniques
- New languages

Already available

- Core components (client-side tools and server-side components)
- Docker files

Coming Soon

- Knowledge base with 780+ public vulnerabilities
- Other features: Gradle plugin, Python setuptools plugin, etc.

Subscribe to the newsletter! vulas-news-request@listserv.sap.com (“subscribe” in the body)

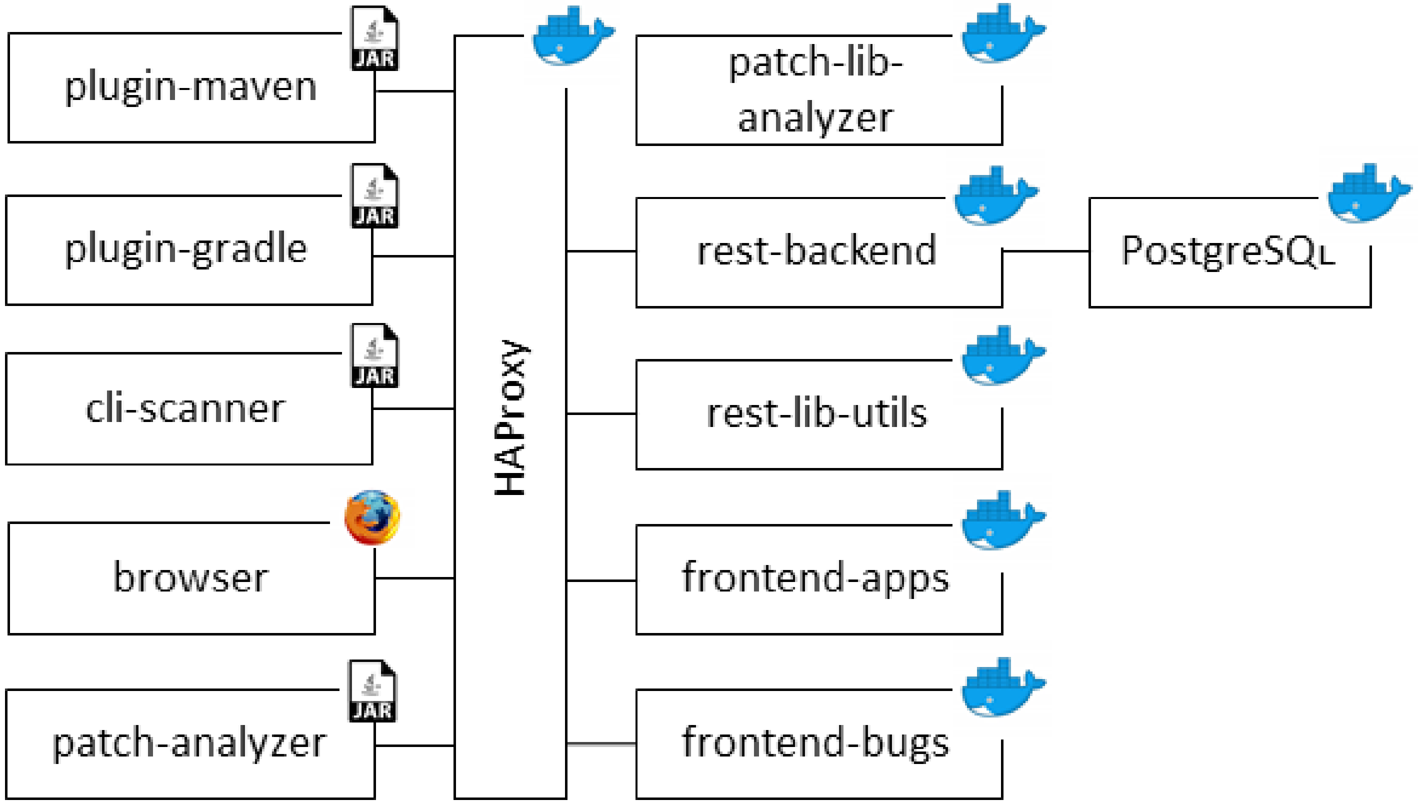
Try it out!

https://github.com/SAP/vulnerability-assessment-tool

Docker to

- build project
- facilitate operation of server-side components

- Interested in an introductory session?
  Contact: henrik.plate@sap.com



Newsletter: vulas-news-request@listserv.sap.com (“subscribe” in the body)

Thank you.
Questions?

Subscribe to receive news on **Vulas**: vulas-news@listserv.sap.com (“subscribe” in the body)
Request for participating to introductory session: henrik.plate@sap.com
https://github.com/SAP/vulnerability-assessment-tool