



Scaling the Security Researcher to Eliminate OSS Vulnerabilities Once and for All

- Jonathan Leitschuh -
- Patrick Way -

Hello!

- Jonathan Leitschuh -

Software Engineer & Security Researcher

Dan Kaminsky Fellowship @ HUMAN Security

GitHub Star & GitHub Security Ambassador

Twitter: @JLLeitschuh

GitHub: JLLeitschuh



Hello!

- Patrick Way -

Senior Software Engineer

OpenRewrite Team @ Moderne

Twitter: @WayPatrick

GitHub: pway99



Disclaimer

Supported by
The
Dan Kaminsky Fellowship
at
HUMAN Security



Chester Higgins/The New York Times

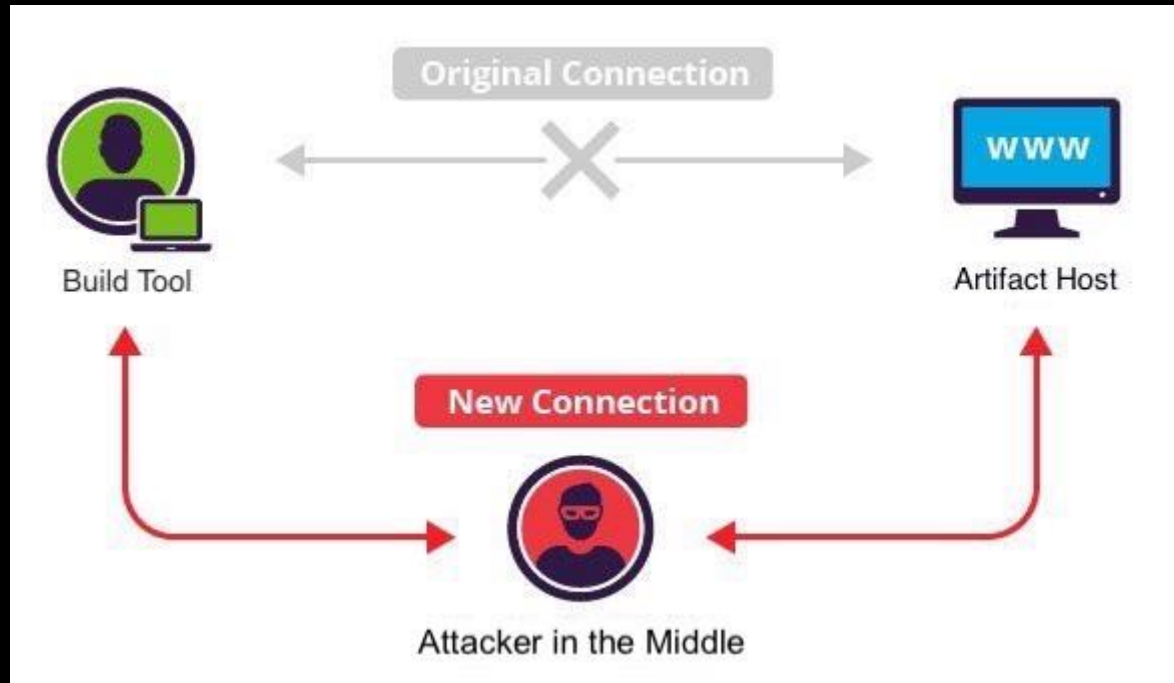
It Started With a Simple Vulnerability

```
// build.gradle
```

```
maven {  
    setUrl("http://dl.bintray.com/kotlin/ktor")  
}
```

HTTP Download of Dependencies in the Java Ecosystem

Why is HTTPS important?



```
<!-- Compiler & Test Dependencies -->
<repositories>
  <repository>
    <id>example-id</id>
    <name>Example insecure repository</name>
    <url>http://[SOME URL HERE]</url>
  </repository>
</repositories>
```

HTTP Download of Dependencies in the Java Ecosystem

```
<!-- Artifact upload - Credentials!! -->  
<distributionManagement>  
  <repository>  
    <id>example-id</id>  
    <name>Example insecure repository</name>  
    <url>http://[SOME URL HERE]</url>  
  </repository>  
</distributionManagement>
```

HTTP Download of Dependencies in the Java Ecosystem

This Vulnerability was Everywhere!



Who else was vulnerable?

ORACLE®



LinkedIn®

stripe

“25% of Sonatype Maven
Central downloads are still
using HTTP”

- Sonatype June 2019 -

How do we fix this?

Decommissioning HTTP Support

On or around January 15th, 2020

- Maven Central (Sonatype)
- JCenter (JFrog)
- Spring (Pivotal)
- Gradle Plugin Portal (Gradle)

“20% of Sonatype Maven
Central Traffic is STILL using
HTTP”

- Sonatype January 2020 -

You can imagine what happened...
January 15th, 2020

BROKEN SOFTWARE

BROKEN SOFTWARE EVERYWHERE

We stopped the bleeding

What about the other repositories?

Only the most commonly used repositories

- Maven Central (Sonatype)
- JCenter (JFrog)
- Spring (Pivotal)
- Gradle Plugin Portal (Gradle)

How do we fix the rest?

Bulk Pull Request Generation!

How?

CodeQL

```
import java
import semmlib.code.xml.MavenPom

private class DeclaredRepository extends PomElement {
  DeclaredRepository() {
    this.getName() = "repository" or
    this.getName() = "snapshotRepository" or
    this.getName() = "pluginRepository"
  }

  string getUrl() { result = getChild("url").(PomElement).getValue() }

  predicate isInsecureRepositoryUsage() {
    getUrl().matches("http://%") or
    getUrl().matches("ftp://%")
  }
}

from DeclaredRepository repository
where repository.isInsecureRepositoryUsage()
select repository,
  "Downloading or uploading artifacts over insecure protocol (eg. http or ftp) to/from repository " +
  repository.getUrl()
```

CodeQL scans 100Ks of OSS Projects

CodeQL

```
import java
import semmlie.code.xml.MavenPom

private class DeclaredRepository extends PomElement {
  DeclaredRepository() {
    this.getName() = "repository" or
    this.getName() = "snapshotRepository" or
    this.getName() = "pluginRepository"
  }

  string getUrl() { result = getChild("url").(PomElement).getValue() }

  predicate isInsecureRepositoryUsage() {
    getUrl().matches("http://%") or
    getUrl().matches("ftp://%")
  }
}

from DeclaredRepository repository
where repository.isInsecureRepositoryUsage()
select repository,
  "Downloading or uploading artifacts over insecure protocol (eg. http or ftp) to/from repository " +
  repository.getUrl()
```

\$2,300 Bounty

Pull Request Generator Version 1

- Python Based
- Wrapper over 'hub' CLI
- One Nasty Regular Expression
- Bouncing off GitHub's rate limiter

```

from vulnerability_fix_engine import VulnerabilityFixModule

@dataclass()
class PomVulnerabilityFixModule(VulnerabilityFixModule):
    branch_name: str = 'fix/JLL/use_https_to_resolve_dependencies'
    clone_repos_location: str = 'cloned_repos'
    data_base_dir: str = 'insecure_pom_data'
    save_point_location: str = 'save_points'
    pr_message_file_absolute_path: str = f'{str(pathlib.Path().absolute())}/PR_MESSAGE.md'
    commit_message: str = textwrap.dedent('''
Use HTTPS instead of HTTP to resolve dependencies

This fixes a security vulnerability in this project where the 'pom.xml'
files were configuring Maven to resolve dependencies over HTTP instead of
HTTPS.

Signed-off-by: Jonathan Leitschuh <Jonathan.Leitschuh@gmail.com>
''')

    p_fix_regex = \
    re.compile(
        r'(?!(?<<repository>)|(?<<pluginRepository>)|(?<<snapshotRepository>))((?!repository).)*(<url>\s*)http://(\s*)(\s*</url>)',
        re.IGNORECASE + re.MULTILINE + re.DOTALL
    )
    replacement = r'\1\2https://\3\4'

    async def do_fix_vulnerable_file(self, project_name: str, file: str, expected_fix_count: int) -> int:
        async with aiofiles.open(file, newline='') as vulnerableFile:
            contents: str = await vulnerableFile.read()

            new_contents, count = self.p_fix_regex.subn(self.replacement, contents)
            if count != expected_fix_count:
                logging.warning(
                    'Fix for "%s" did match expected fix count: (expected: %d, actual: %d)',
                    project_name,
                    expected_fix_count,
                    count
                )

            async with aiofiles.open(file, 'w', newline='') as vulnerableFile:
                await vulnerableFile.write(new_contents)

            return count

```

```
... p_fix_regex = \  
... re.compile(  
... r'(?:(?<=<repository>)|(?<=<pluginRepository>)|(?<=<snapshotRepository>))((?:?!repository).)*(<url>\s*http://(\S*)(\s*/url))',  
... re.IGNORECASE + re.MULTILINE + re.DOTALL  
... )  
... replacement = r'\1\2https://\3\4'
```


**I had a problem so I used regular
expressions**

**Now I have two
problems!**

It worked!



Created Assigned Mentioned Review requests

1,055 Open	504 Closed	Visibility	Organization	Sort
01Sharpshooter/Social [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#1 opened on Feb 11 by JLLeitschuh			
4thline/cling [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#250 opened on Feb 11 by JLLeitschuh			
1000Memories/photon-core [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#4 opened on Feb 11 by JLLeitschuh			
18838928050/ssmtest [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#1 opened on Feb 11 by JLLeitschuh			
2xel/spring-bootstrap-tiles [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#1 opened on Feb 11 by JLLeitschuh			
weamlady2/iOS_remote [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#23 opened on Feb 11 by JLLeitschuh			
yjshen/zzzzobspk [SECURITY] Use HTTPS to resolve dependencies in Maven Build ✓	#1 opened on Feb 11 by JLLeitschuh			1
wlu-mstr/hbase-ormlite [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#1 opened on Feb 11 by JLLeitschuh			
zhangdaiscott/jeecg [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#53 opened on Feb 11 by JLLeitschuh			
wso2/carbon-device-mgt-plugins [SECURITY] Use HTTPS to resolve dependencies in Maven Build ✗	#927 opened on Feb 11 by JLLeitschuh • Review required			21
wso2/product-iiots [SECURITY] Use HTTPS to resolve dependencies in Maven Build ✓	Resolution/State #1940 opened on Feb 11 by JLLeitschuh • Review required			17
xautlx/s2jh4net [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#30 opened on Feb 11 by JLLeitschuh			
xzer/run-jetty-run [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#214 opened on Feb 11 by JLLeitschuh			
yanghua/banyan [SECURITY] Use HTTPS to resolve dependencies in Maven Build	#3 opened on Feb 11 by JLLeitschuh			

```

@@ -19,15 +19,15 @@
19 .....</repository>
20 .....<repository>
21 .....<id>onarandombox</id>
22 .....<url>http://repo.onarandombox.com/content/groups/public</url>
23 .....</repository>
24 .....<repository>
25 .....<id>spigot</id>
26 .....<url>https://hub.spigotmc.org/nexus/content/groups/public</url>
27 .....</repository>
28 .....<repository>
29 .....<id>vault-repo</id>
30 .....<url>http://nexus.hc.to/content/repositories/pub_releases</url>
31 .....</repository>
32 .....<repository>
33 .....<id>minebench-repo</id>

@@ -36,15 +36,15 @@
36 .....<!-- Has a copy of metrics R8-SNAPSHOT !-->
37 .....<repository>
38 .....<id>elmakers-repo</id>
39 .....<url>http://maven.elmakers.com/repository</url>
40 .....</repository>
41 .....</repositories>
42 .....
43 .....<pluginRepositories>
44 .....<pluginRepository>
45 .....<id>doodleproject-repo</id>
46 .....<name>DoodleProject Maven 2 Repository</name>
47 .....<url>http://doodleproject.sourceforge.net/maven2/release</url>
48 .....<releases>
49 .....<enabled>true</enabled>
50 .....</releases>

@@ -353,11 +353,11 @@
353 .....<distributionManagement>
354 .....<repository>

```

HTTP Download of Dependencies

1,596

Pull Requests

~40%

Merged or Accepted

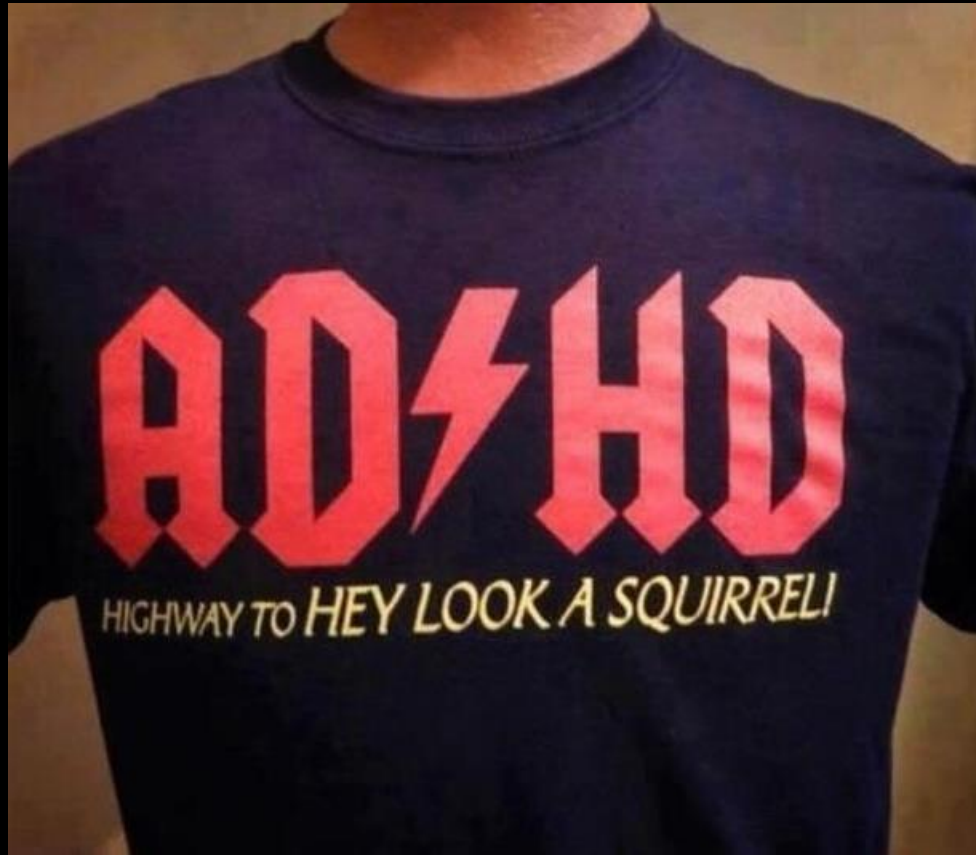
\$4,000

Thanks to the GitHub Security Lab!

I got hooked on Bulk Pull Request Generation



I have a Problem



I was finding too many security vulnerabilities!

```

<aws2/carbon-mediation>/components/./utils/SynapseArtifactInitUtils.java
T 1:254
255 // If the entry is a file, write the file
256 copyOutputStream(zipFile.getInputStream(entry),
257     new BufferedOutputStream(new FileOutputStream(destPath + entry.getName())));
Unsanitized archive entry, which may contain '..' is used in a file system operation. [Show paths]
258     }
259 }
I 260-276

<apache/druid>/indexing-hadoop/./indexer/JobHelper.java
T 1:768
769 try {FileOutputStream in = new FileOutputStream(fileSystem.open(dip, 1 <= 13)); {
770     for (ZipEntry entry = in.getNextEntry(); entry != null; entry = in.getNextEntry()) {
771         final String fileName = entry.getName();
Unsanitized archive entry, which may contain '..' is used in a file system operation. [Show paths]
772         final String outputPath = new File(outDir, fileName).getAbsolutePath();
773     }
I 774-804

<HongZhaohua/jstarcraft-core>/jstarcraft-core-common/./utility/PressUtility.java
T 1:179
180 ArchiveEntry archiveEntry;
181 while (null != (archiveEntry = archiveInputStream.getNextEntry())) {
182     File file = new File(toDirectory, archiveEntry.getName());
Unsanitized archive entry, which may contain '..' is used in a file system operation. [Show paths]
183     try {FileOutputStream fileOutputStream = new FileOutputStream(file); {
184         int length = -1;
I 185-217
T 218
219 ArchiveEntry archiveEntry;
220 while (null != (archiveEntry = archiveInputStream.getNextEntry())) {
221     File file = new File(toDirectory, archiveEntry.getName());
Unsanitized archive entry, which may contain '..' is used in a file system operation. [Show paths]
222     try {FileOutputStream fileOutputStream = new FileOutputStream(file); {
223         int length = -1;
I 223-233

<deepjava.library.djl>/api/./repository/AbstractRepository.java
T 1:245
246 TarArchiveEntry entry;
247 while ((entry = tis.getNextTarEntry()) != null) {
248     String entryName = entry.getName();
Unsanitized archive entry, which may contain '..' is used in a [ 2 Values ]. [Show paths]
249     if (entryName.contains("..")) {
250         throw new IOException("Malicious zip entry: " + entryName);
I 251-273

<deepjava.library.djl>/api/./util/ZipUtils.java
T 1:39
40 ZipEntry entry;
41 while ((entry = zis.getNextEntry()) != null) {
42     String name = entry.getName();
Unsanitized archive entry, which may contain '..' is used in a [ 2 Values ]. [Show paths]
43     if (name.contains("..")) {
44         throw new IOException("Malicious zip entry: " + name);
I 45-103

<cybertaxonomy/cdmlib/cdmlib-ext/./scratchpads/ScratchpadsService.java
T 1:109
109 System.out.println("Extracting: " + ze);
100 FileOutputStream fos = new FileOutputStream(ze.getName());
Unsanitized archive entry, which may contain '..' is used in a file system operation. [Show paths]

```

I was finding too many security vulnerabilities!

I was finding too many security vulnerabilities!

I needed automation!

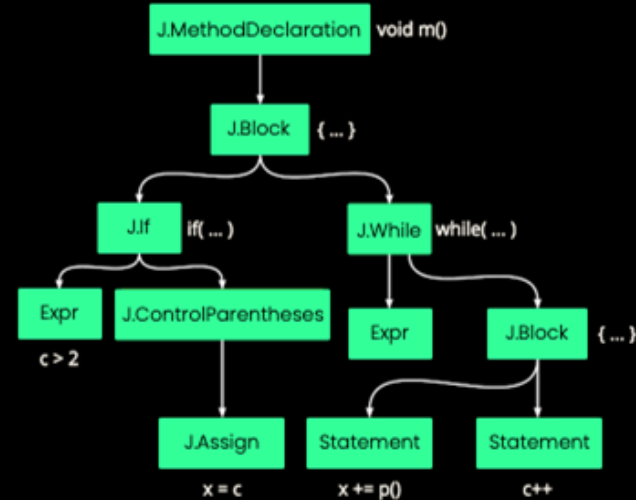
Automated Accurate Transformations at a Massive Scale

The logo icon consists of a square divided into four quadrants by a vertical and a horizontal line. A diagonal line runs from the top-left to the bottom-right. The top-right quadrant is filled with a white quarter-circle arc, while the other three quadrants are empty.

OpenRewrite

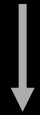
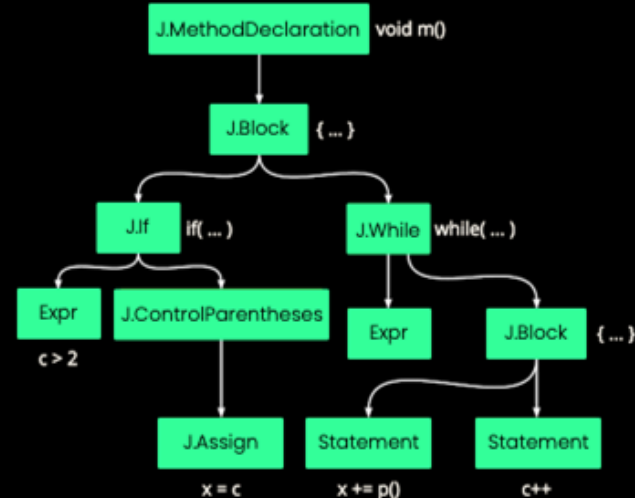
Abstract Syntax Tree (AST)

```
... /** myMethod */  
... void m(){  
...     if (c > 2) {  
...         // c is more than 2  
...         x = c;  
...     }  
...     while(c < 10) { // increment x  
...         x += p();  
...         c++;  
...     }  
... }
```



Abstract Syntax Tree (AST)

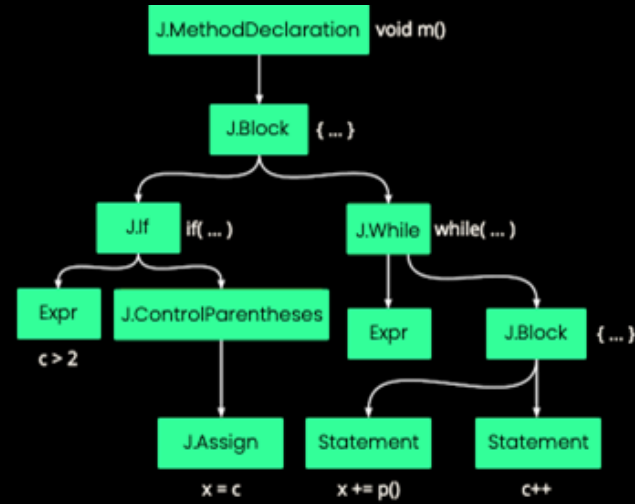
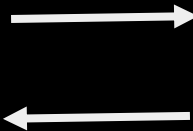
```
... /** myMethod */  
... void m(){  
...     if (c > 2) {  
...         // c is more than 2  
...         x = c;  
...     }  
...     while(c < 10) { // increment x  
...         x += p();  
...         c++;  
...     }  
... }
```



```
void m(){if(c>2){x=c;}while(c<10){x+=p();c++;}}
```

Format Preserving AST

```
... /** myMethod */
... void m(){
...     if (c > 2) {
...         // c is more than 2
...         x = c;
...     }
...     while(c < 10) { // increment x
...         x += p();
...         c++;
...     }
... }
```



Whitespace and comments are preserved

Generated code matches the Surrounding Formatting

Spaces

```
String name = entry.getName();
Path path = dir.resolve(name);
if (!path.normalize().startsWith(dir)) {
    ... throw new RuntimeException("Bad zip entry");
}
OutputStream os = Files.newOutputStream(path);
```

Tabs

```
Path path = dir.resolve(name);
if (!path.normalize().startsWith(dir)) {
    ... throw new RuntimeException("Bad zip entry");
}
OutputStream os = Files.newOutputStream(path);
```

Braces on new line

```
String name = entry.getName();
Path path = dir.resolve(name);
if (!path.normalize().startsWith(dir))
{
    ... throw new RuntimeException("Bad zip entry");
}
OutputStream os = Files.newOutputStream(path);
```

Accurate Transformations Require Fully Type-attributed ASTs

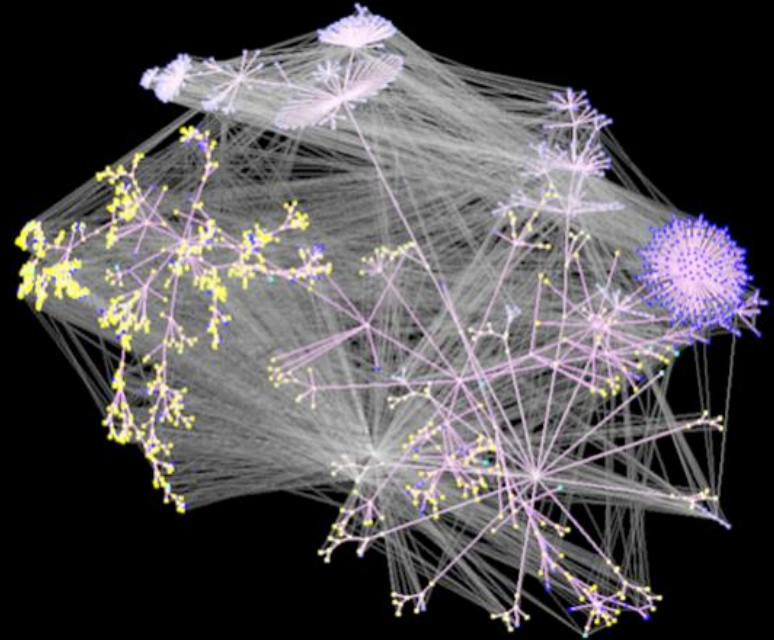
```
log.info("...");
```

Is that log4j, slf4j, LogBack?

The OpenRewrite AST is both Syntactically and Semantically aware.



Syntax alone



With type attribution and formatting

Even simple code produces complex AST



```
if (!path.normalize().startsWith(dir)) {  
    throw new RuntimeException("Bad zip entry");  
}
```



```
final JavaTemplate noZipSlipPathStartsWithPathTemplate =
JavaTemplate.builder(this::getCursor, code: "" +
"if (!#{any(java.nio.file.Path)}.normalize()" +
".....".startsWith(#{any(java.nio.file.Path)})) {\n" +
".....throw new RuntimeException(\"Bad zip entry\");\n" +
"}").build();
```

```
final JavaTemplate noZipSlipPathStartsWithPathTemplate =
JavaTemplate.builder(this::getCursor, code: "" +
"if (!#{any(java.nio.file.Path)}.normalize()" +
".....".startsWith(#{any(java.nio.file.Path)})) {\n" +
".....throw new RuntimeException(\"Bad zip entry\");\n" +
"}").build();
```

```
return b.withTemplate(
..... noZipSlipPathStartsWithPathTemplate,
..... zipSlipSimpleInjectGuardInfo.statement.getCoordinates().after(),
..... zipSlipSimpleInjectGuardInfo.zipEntry,
..... zipSlipSimpleInjectGuardInfo.parentDir
);
```

```
public class MyZipHelper {  
    public void m1(ZipEntry entry, Path dir) throws Exception {  
        String name = entry.getName();  
        Path path = dir.resolve(name);  
        OutputStream os = Files.newOutputStream(path);  
    }  
}
```



```
public class MyZipHelper {  
    public void m1(ZipEntry entry, Path dir) throws Exception {  
        String name = entry.getName();  
        Path path = dir.resolve(name);  
        if (!path.normalize().startsWith(dir)) {  
            throw new RuntimeException("Bad zip entry");  
        }  
        OutputStream os = Files.newOutputStream(path);  
    }  
}
```

What is possible now?

What other vulnerabilities can we fix?

Three Vulnerabilities

1. Temporary Directory Hijacking
2. Partial Path Traversal
3. Zip Slip

Vulnerability #1

Temporary Directory Hijacking

Temporary Directory on
Unix-Like Systems is
Shared between All Users

Temporary Directory Hijacking - Vulnerable

```
File f = File.createTempFile(  
    "prefix",  
    "suffix"  
);  
f.delete();  
f.mkdir();
```

ASK STACK OVERFLOW



GET VULNERABILITIES

imgflip.com

Temporary Directory Hijacking - Vulnerable

```
File f = File.createTempFile(  
    "prefix",  
    "suffix"  
);  
f.delete();  
f.mkdir();
```

Temporary Directory Hijacking - Vulnerable

```
File f = File.createTempFile(  
    "prefix",  
    "suffix"  
);  
f.delete();  
// 🚩 Race condition  
f.mkdir(); // Returns `false`
```

Temporary Directory Hijacking - Imperfect Fix

```
File f = File.createTempFile(  
    "prefix",  
    "suffix"  
);  
f.delete();  
if (!f.mkdir())  
    throw new IOException("Error");
```

Temporary Directory Hijacking - Fix

```
// Since Java 1.7
File f =
    Files
        .createTempDirectory("prefix")
        .ToFile();
```

Temporary Directory Hijacking - CVEs

- CVE-2022-27772 - Spring Boot
- CVE-2021-20202 - Keycloak
- CVE-2021-21331 - DataDog API
- CVE-2020-27216 - Eclipse Jetty
- CVE-2020-17521 - Apache Groovy
- CVE-2020-17534 - Apache netbeans-html4j

Temporary Directory Hijacking

Pull Request Statistics

Temporary Directory Hijacking

64 Pull Requests!

Temporary Directory Hijacking - Pull Requests

Temporary Directory Hijacking - Putting it all together

```
src/main/java/org/jenkinsci/backend/jpicreate/WebAppMain.java

@@ -10,6 +10,7 @@ org.openrewrite.java.security.UseFilesCreateTempDirectory
10 10 import javax.sound.midi.SysexMessage;
11 11 import java.io.File;
12 12 import java.io.IOException;
13 13 + import java.nio.file.Files;
13 14
14 15 /**
15 16 *

@@ -41,9 +42,7 @@
41 42 FileUtils.copyURLToFile(
42 43     getClass().getClassLoader().getResource("maven.zip"),
43 44     zip);
44 44 - File bin = File.createTempFile("maven","bin");
45 45 - bin.delete();
46 46 - bin.mkdirs();
45 45 + File bin = Files.createTempDirectory("maven" + "bin").toFile();
47 46
48 47 Process unzip = new ProcessBuilder("unzip", zip.getAbsolutePath()
49 48     .directory(bin).redirectErrorStream(true).start();
```

Temporary Directory Hijacking - Putting it all together

src/test/java/com/google/jenkins/plugins/credentials/oauth/JsonServiceAccountConfigTestUtil.java

@@ -22,6 +22,7 @@ org.openrewrite.java.security.UseFilesCreateTempDirectory

```
22 22 import java.io.IOException;
23 23 import java.io.StringWriter;
24 24 import java.nio.charset.Charset;
25 + import java.nio.file.Files;
25 26 import java.security.KeyPair;
26 27 import java.security.KeyPairGenerator;
27 28 import java.security.NoSuchAlgorithmException;
```

@@ -64,13 +65,7 @@

```
64 65
65 66 private static File getTempFolder() throws IOException {
66 67     if (tempFolder == null) {
67 -         tempFolder = File.createTempFile("temp", Long.toString(System.nanoTime()));
68 -         if (!tempFolder.delete()) {
69 -             throw new IOException("Could not delete temp file: " + tempFolder.getAbsolutePath());
70 -         }
71 -         if (!tempFolder.mkdir()) {
72 -             throw new IOException("Could not create temp directory: " + tempFolder.getAbsolutePath());
73 -         }
68 +         tempFolder = Files.createTempDirectory("temp" + Long.toString(System.nanoTime())).toFile();
74 69         tempFolder.deleteOnExit();
75 70     }
76 71     return tempFolder;
```

Vulnerability #2

Partial Path Traversal

Partial Path Traversal

```
"/user/sam"
```

Partial Path Traversal

```
"/user/sam"
```

```
"/user/samantha"
```

Partial Path Traversal

Allows an attacker access to a sibling directory with the same prefix

Partial Path Traversal

```
"/user/sam"
```

Allows an attacker access to a sibling directory with the same prefix

Partial Path Traversal

```
"/user/sam"
```

Allows an attacker access to a sibling directory with the same prefix

```
"/user/samantha"
```

Partial Path Traversal

```
"/user/sam"
```

Allows an attacker access to a sibling directory with the same prefix

```
"/user/samantha"
```

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    parent, userControlled()  
);  
  
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    throw new IOException(  
        "Detected path traversal attack!"  
    );  
}
```

```
new File("/user/sam/")
```

```
new File("/user/sam/")
```

```
File.getCanonicalPath()
```

```
new File("/user/sam/")
```

```
File.getCanonicalPath()
```

```
"/user/sam"
```

```
new File("/user/sam/")
```

```
File.getCanonicalPath()
```

```
"/user/sam"
```



Partial Path Traversal - Vulnerability

```
File dir = new File(  
    parent, userControlled()  
);  
  
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    throw new IOException(  
        "Detected path traversal attack!"  
    );  
}
```

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    "/user/sam/", userControlled()  
);  
  
if (!dir.getCanonicalPath()  
    .startsWith("/user/sam")) {  
    ...  
  
}
```

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    "/user/sam/", "../samantha/baz"  
);  
  
if (!dir.getCanonicalPath()  
    .startsWith("/user/sam")) {  
    ...  
}
```

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    "/user/sam/", "../samantha/baz"  
);  
  
if (!"/user/samantha/baz"  
    .startsWith("/user/sam")) {  
    ...  
}
```

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    "/user/sam/", "../samantha/baz"  
);  
  
if (!"/user/samantha/baz"  
    .startsWith("/user/sam")) {  
    throw new IOException(  
        "Detected path traversal attack!"  
    );  
}
```



Partial Path Traversal Fix!

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    parent, userControlled()  
);  
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    throw new IOException(  
        "Detected path traversal attack!"  
    );  
}
```

Partial Path Traversal - Vulnerability

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    ...  
  
}
```


Partial Path Traversal - Fix #1

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath() +  
               File.separatorChar)) {  
    ...  
}
```

Partial Path Traversal - Fix #2

```
if (!dir.getCanonicalFile()  
    .toPath().startsWith(  
        parent.getCanonicalFile().toPath())) {  
    ...  
}
```

Partial Path Traversal - Fix #2 - Better

```
if (!dir.getCanonicalFile()  
    .toPath().startsWith(  
        parent.getCanonicalFile().toPath())) {  
    ...  
}
```



How do we find this vulnerability?

Partial Path Traversal - Vulnerability

```
File dir = new File(  
    parent, userControlled()  
);  
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    throw new IOException(  
        "Detected path traversal attack!"  
    );  
}
```

Partial Path Traversal - Vulnerability

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    ...  
  
}
```

Partial Path Traversal - Vulnerability

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    ...  
  
}
```

Partial Path Traversal - Safe

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath() +  
               File.separatorChar)) {  
    ...  
}
```


It can't be that easy, can it?

Partial Path Traversal - Vulnerability

```
if (!dir.getCanonicalPath()  
    .startsWith(parent.getCanonicalPath())) {  
    ...  
  
}
```

Partial Path Traversal - Vulnerability

```
String dirCanonical = dir.getCanonicalPath();  
  
if (!dirCanonical  
    .startsWith(parent.getCanonicalPath())) {  
    ...  
  
}
```

Partial Path Traversal - Vulnerability

```
String dirCanonical = dir.getCanonicalPath();
String pCanonical = parent.getCanonicalPath();

if (!dirCanonical
    .startsWith(pCanonical)) {
    ...
}
```

Partial Path Traversal - Vulnerability

```
String dirCanonical = dir.getCanonicalPath();
String pCanonical = parent.getCanonicalPath() +
                        File.separatorChar;
if (!dirCanonical
    .startsWith(pCanonical)) {
    ...
}
```

We need Data Flow Analysis

Partial Path Traversal - DataFlow


```
String dirCanonical = dir.getCanonicalPath();  
String pCanonical = parent.getCanonicalPath() +  
                    File.separatorChar;  
  
if (!dirCanonical  
    .startsWith(pCanonical)) {  
    ...  
}
```

Partial Path Traversal - Data Flow

```
String dirCanonical ← dir.getCanonicalPath();  
String pCanonical = parent.getCanonicalPath() +  
                    File.separatorChar;  
  
if (!dirCanonical  
    .startsWith(pCanonical)) {  
    ...  
}
```


Partial Path Traversal - Data Flow

```
String dirCanonical ← dir.getCanonicalPath();  
String pCanonical = parent.getCanonicalPath() +  
                    File.separatorChar;  
  
if (!dirCanonical  
    .startsWith(pCanonical)) {  
    ...  
}
```

A yellow arrow points from the variable *dirCanonical* in the first line of code down to the *dirCanonical* variable in the *startsWith* method call of the *if* statement.

Partial Path Traversal - Data Flow

```
String dirCanonical ← dir.getCanonicalPath();  
String pCanonical ← parent.getCanonicalPath() +  
File.separatorChar;  
  
if (!dirCanonical  
    .startsWith(pCanonical)) {  
    ...  
}
```

The diagram illustrates the data flow in the provided code. An orange arrow points from the `dir.getCanonicalPath()` call to the `dirCanonical` variable. A green arrow points from the `parent.getCanonicalPath() + File.separatorChar` call to the `pCanonical` variable. A second green arrow points from the `pCanonical` variable to the `startsWith(pCanonical)` method call within the `if` statement.

Partial Path Traversal - Data Flow

```
String dirCanonical ← dir.getCanonicalPath();  
String pCanonical ← parent.getCanonicalPath() +  
                    File.separatorChar;  
String pCanonical2 ← pCanonical;  
if (!dirCanonical  
    .startsWith(pCanonical2)) {  
    ...  
}
```

Data Flow

Uncovers hard to find Vulnerabilities
and prevents
False Positives

Data Flow Analysis

```
class GetCanonicalPathToStartsWithLocalFlow extends LocalFlowSpec<J.MethodInvocation, Expression> {

    @Override
    public boolean isSource(J.MethodInvocation methodInvocation, Cursor cursor) {
        return new MethodMatcher("java.io.File getCanonicalPath()")
            .matches(methodInvocation);
    }

    @Override
    public boolean isSink(Expression expression, Cursor cursor) {
        return InvocationMatcher
            .fromMethodMatcher(
                new MethodMatcher(
                    "java.lang.String startsWith(java.lang.String)"
                )
            )
            .advanced()
            .isSelect(cursor);
    }
}
```

Partial Path Traversal - Putting it all together

src/main/java/de/neemann/digital/draw/library/ElementLibrary.java

@@ -412,7 +412,7 @@ org.openrewrite.java.security.PartialPathTraversalVulnerability

```
412 412         try {
413 413             String root = rootLibraryPath.getCanonicalPath();
414 414             String path = file.getParentFile().getCanonicalPath();
415 -         return path.startsWith(root);
415 +         return file.getParentFile().getCanonicalFile().toPath().startsWith(root);
416 416         } catch (IOException e) {
417 417             return false;
418 418         }
```

Example Case: AWS Java SDK CVE-2022-31159

aws-sdk-java/aws-java-sdk-s3/src/main/java/com/amazonaws/services/s3/transfer/TransferManager.java

Lines 1513 to 1519 in 5be0807

```
1513     private boolean leavesRoot(File localBaseDirectory, String key) {
1514         try {
1515             return !new File(localBaseDirectory, key).getCanonicalPath().startsWith(localBaseDirectory.getCanonicalPath());
1516         } catch (IOException e) {
1517             throw new RuntimeException("Unable to canonicalize paths", e);
1518         }
1519     }
```


aws-sdk-java/aws-java-sdk-s3/src/main/java/com/amazonaws/services/s3/transfer/TransferManager.java

Lines 1420 to 1423 in ae88c8a

```
1420     if ( leavesRoot(destinationDirectory, s.getKey()) ) {
1421         throw new RuntimeException("Cannot download key " + s.getKey() +
1422             ", its relative path resolves outside the parent directory.");
1423     }
```

Vulnerability Disclosure Drama!

Aside: Email with AWS Security Team

AWS: We'd like to award you a bug bounty, however you'd need to sign an NDA.

Aside: Email with AWS Security Team

AWS: We'd like to award you a bug bounty, however you'd need to sign an NDA.

Jonathan: I don't normally agree to NDA's. Can I read it first before potentially agreeing?

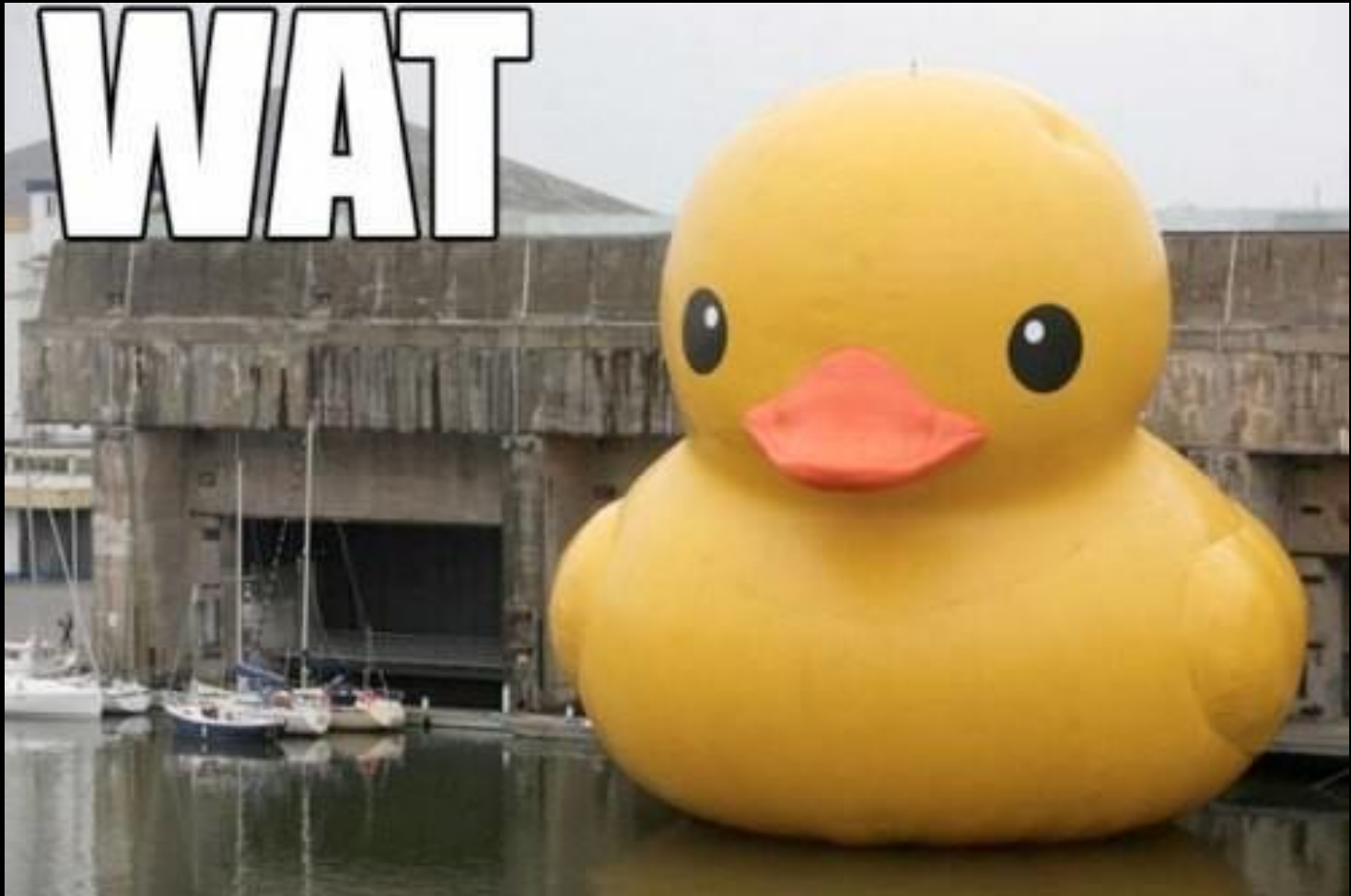
Aside: Email with AWS Security Team

AWS: We'd like to award you a bug bounty, however you'd need to sign an NDA.

Jonathan: I don't normally agree to NDA's. Can I read it first before potentially agreeing?

AWS: We're unable to share the bug bounty program NDA since it and other contract documents are considered sensitive by the legal team.

WAT



AMAZON WEB SERVICES
used LEGALESE !

AMAZON WEB SERVICES
used LEGALESE !

It hurt itself in
its confusion!

imgflip.com

Vulnerability #3

Zip Slip

Zip Slip

Path Traversal Vulnerability
while
Unpacking Zip File Entries

Zip Slip

```
void zipSlip(File destination, ZipFile zip) {
    Enumeration<? extends ZipEntry> entries = zip.entries();
    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(destination, e.getName());
        IOUtils.copy(
            zip.getInputStream(e),
            new FileOutputStream(f)
        );
    }
}
```

Zip Slip

```
ZipEntry e = entries.nextElement();  
File f = new File(destination, e.getName());  
IOUtils.copy(  
    zip.getInputStream(e),  
    new FileOutputStream(f)  
);
```

Zip Slip is Complicated

Zip Slip

```
ZipEntry e = ...  
File f = new File(destination, e.getName());  
  
IOUtils.copy(  
    zip.getInputStream(e),  
    new FileOutputStream(f)  
);
```

Zip Slip

```
ZipEntry e = ...
File f = new File(destination, e.getName());
if (!f.toPath().startsWith(destination.toPath())) {
    throw new IOException("Bad Zip Entry!");
}
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

The Problem with Zip Slip

Zip Slip

```
ZipEntry e = ...
File f = new File(destination, e.getName());
if (!f.toPath().startsWith(destination.toPath())) {
    throw new IOException("Bad Zip Entry!");
}
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

Zip Slip

```
ZipEntry e = ...
File f = new File(destination, e.getName());
if (f.toPath().startsWith(destination.toPath())) {
    IOUtils.copy(
        zip.getInputStream(e),
        new FileOutputStream(f)
    );
}
```

Control Flow Analysis

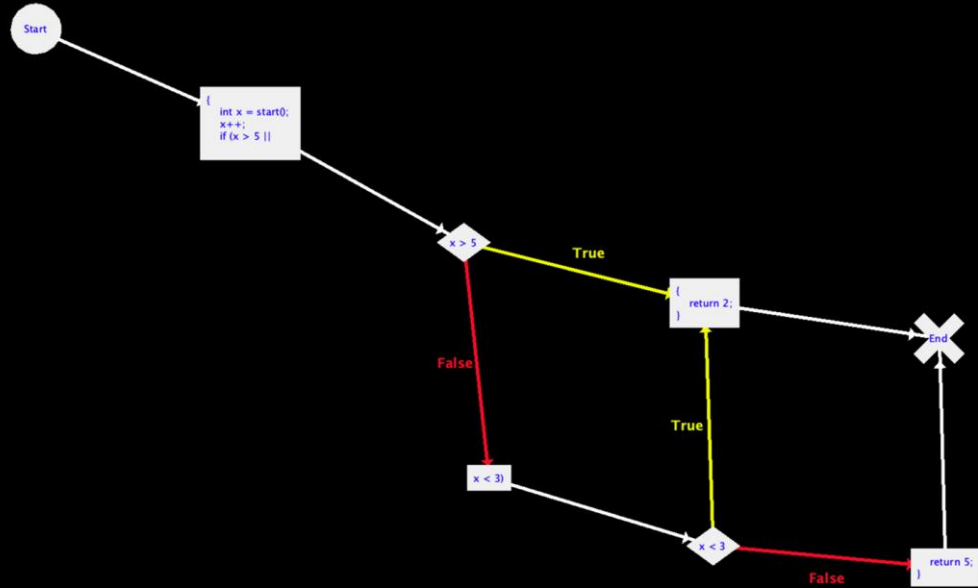
Control Flow Analysis

```
File f = new File(destination, e.getName());
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

```
File f = new File(destination, e.getName());
if
(!f.toPath().startsWith(destination.toPath())){
    throw new IOException("Bad Zip Entry!");
}
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

Control Flow - OpenRewrite

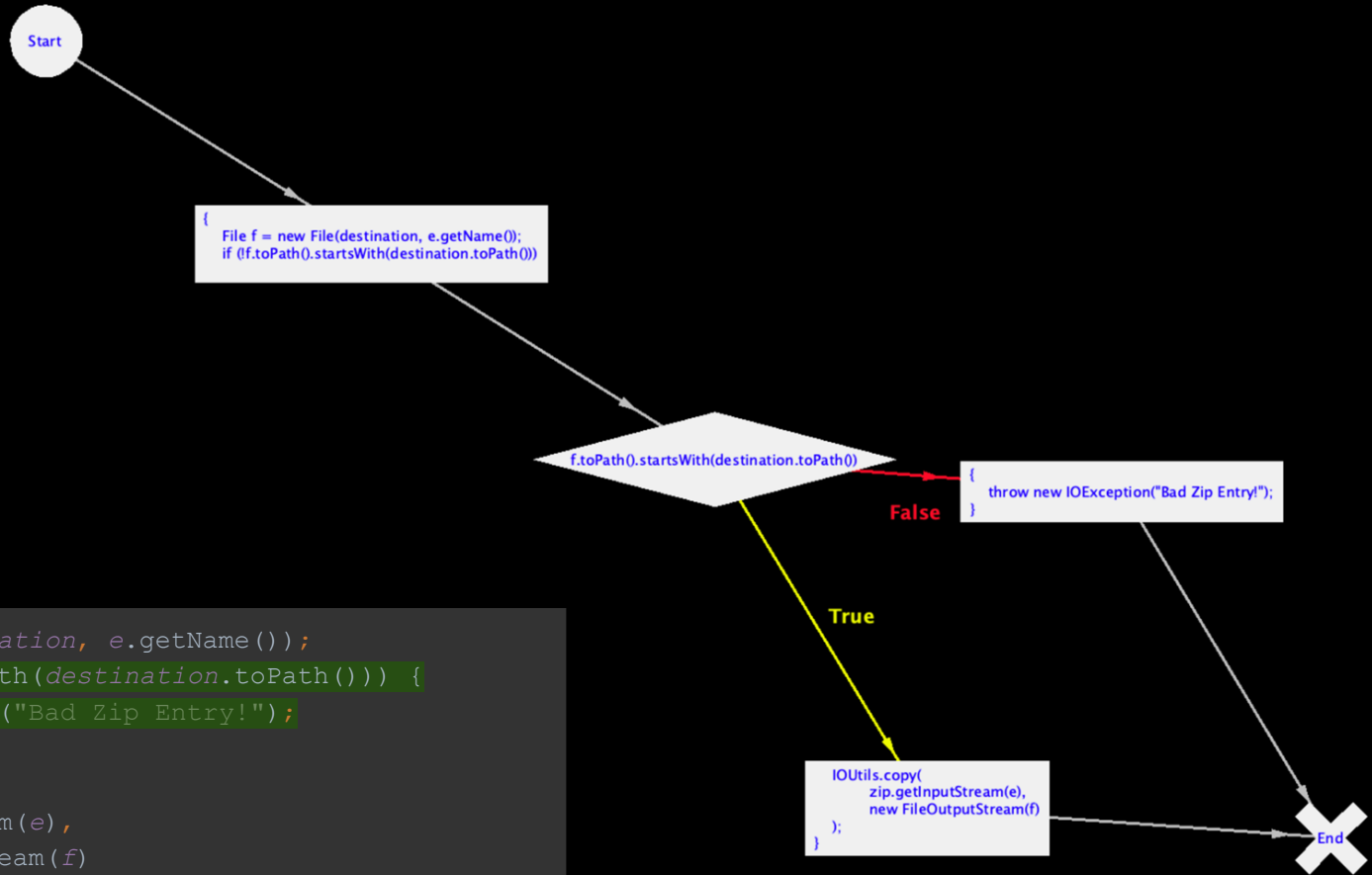
```
abstract class Test {  
    abstract int start();  
    int test() {  
        int x = start();  
        x++;  
        if (x > 5 || x < 3) {  
            return 2;  
        }  
        return 5;  
    }  
}
```



Zip Slip

```
ZipEntry e = ...
File f = new File(destination, e.getName());
if (!f.toPath().startsWith(destination.toPath())) {
    throw new IOException("Bad Zip Entry!");
}
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

Zip Slip



```
File f = new File(destination, e.getName());  
if (!f.toPath().startsWith(destination.toPath())) {  
    throw new IOException("Bad Zip Entry!");  
}  
IOUtils.copy(  
    zip.getInputStream(e),  
    new FileOutputStream(f)  
);
```

Zip Slip - Putting it all together

src/main/java/org/owasp/webgoat/lessons/path_traversal/ProfileZipSlip.java

@@ -58,6 +58,9 @@ org.openrewrite.java.security.ZipSlip

```
58 58         while (entries.hasMoreElements()) {
59 59             ZipEntry e = entries.nextElement();
60 60             File f = new File(tmpZipDirectory.toFile(), e.getName());
61 +             if (!f.toPath().normalize().startsWith(tmpZipDirectory.toFile().toPath())) {
62 +                 throw new RuntimeException("Bad zip entry");
63 +             }
61 64             InputStream is = zip.getInputStream(e);
62 65             Files.copy(is, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
63 66         }
```


Zip Slip - Putting it all together

jbake-core/src/main/java/org/jbake/app/ZipUtil.java

@@ -28,7 +28,10 @@ org.openrewrite.java.security.ZipSlip

```
28 28         byte[] buffer = new byte[1024];
29 29
30 30         while ((entry = zis.getNextEntry()) != null) {
31 -         File outputFile = new File(outputFolder.getCanonicalPath() + File.separatorChar + entry.getName());
31 +         File outputFile = new File(outputFolder.getCanonicalPath(), entry.getName());
32 +         if (!outputFile.toPath().normalize().startsWith(outputFolder.getCanonicalPath())) {
33 +             throw new RuntimeException("Bad zip entry");
34 +         }
32 35         File outputParent = new File(outputFile.getParent());
33 36         outputParent.mkdirs();
```

Pull Request Generation!

GOT SECURITY VULNERABILITIES?



**YOU GET A PULL REQUEST!
YOU GET A PULL REQUEST!
EVERYBODY GETS A PULL REQUEST!!!**

imgflip.com

Problems with Pull Request Generation

How fast can we generate
Pull Requests?

Pull Request Generation Steps

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub
4. Rename Repository on GitHub

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub
4. Rename Repository on GitHub
5. Push changes

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub
4. Rename Repository on GitHub
5. Push changes
6. Create Pull Request on GitHub

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub
4. Rename Repository on GitHub
5. Push changes
6. Create Pull Request on GitHub

File IO

Git Operation

GitHub API

Pull Request Generation Steps

1. Checkout (ie. Download) code Repository
2. Branch, Apply Diff, & Commit
3. Fork Repository on GitHub
4. Rename Repository on GitHub
5. Push changes
6. Create Pull Request on GitHub

File IO

Git Operation

GitHub API

**IF YOU COULD STOP
RATE LIMITING YOUR API**

THAT WOULD BE GREAT

imgflip.com

We've made it this far

- ✓ Vulnerabilities Detected
- ✓ Style Detected
- ✓ Code Fixed & Diff Generated
- ✓ Rate Limit Bypassed

We've made it this far

- ✓ Vulnerabilities Detected
- ✓ Style Detected
- ✓ Code Fixed & Diff Generated
- ✓ Rate Limit Bypassed

How do we do this for all the repositories?

Moderne

- Free for Open Source Projects!
- ~7,000 Repositories indexed
- Run Open Rewrite Transformations at Scale
- Generates and Updates Pull Requests

800+ OpenRewrite Recipes including complete Framework Migrations

The image shows a screenshot of the Moderne website, which is a platform for OpenRewrite recipes. The interface is clean and organized into several sections:

- Analyze your code:** This section contains four recipes:
 - Find method usages:** Find method usages by pattern.
 - Change method name:** Rename a method.
 - Find types:** Find type references by name.
 - Find missing configuration:** Find Kubernetes resources with missing configuration.
- Modernize your code:** This section is further divided into:
 - Java recipes »:**
 - Migrate Java 8 to Java 11:** This recipe will apply changes commonly needed when migrating...
 - Java security best practices:** Applies security best practices to Java code.
 - Format Java code:** Format Java code using a standard comprehensive set of Java formatting...
 - Migrate JUnit asserts to AssertJ:** AssertJ provides a rich set of assertions, truly helpful error...
 - Spring recipes »:**
 - Spring Boot 2.x migration from Spring Boot 1.x:** Migrates Spring Boot 1.x to 2.x including best practices.
 - Spring Boot 2.x best practices:** Applies best practices to Spring Boot 2 applications.
 - JUnit Jupiter for Spring Boot 2.x projects:** Migrates Spring Boot 2.x projects having JUnit 4.x tests to JUnit Jupiter.
 - Remove @RequestMapping annotations:** Replace method declaration @RequestMapping annotations with...
 - Kubernetes recipes »:**
 - Kubernetes best practices:** Applies best practices to Kubernetes manifests.
 - Ensure liveness probe is configured:** The kubelet uses liveness probes to know when to schedule restarts for...
 - Ensure readiness probe is configured:** Using the Readiness Probe ensures teams define what actions need to b...
 - Cap exceeds resource value:** Cap resource values that exceed a specific maximum.
 - Maven recipes »:**
 - Manage dependencies:** Make existing dependencies managed by moving their version to...
 - Maven dependency insight:** Find direct and transitive dependencies matching a group...
 - Remove redundant explicit dependency versions:** Remove explicitly-specified dependency versions when a parent...
 - Upgrade Maven dependency version:** Upgrade the version of a dependency by specifying a group or group and...
- ... and much, much more »**

Bulk Pull Request Generation - public.moderne.io

The Moderne interface displays the following commit results:

Recipe	Success	Progress	Started
sulIAO	93%	100%	1 day ago

Commit title: vuln-fix: Use HTTPS instead of HTTP to resolve dependencies

Commit messages: This fixes a security vulnerability in this project where the build.gradle files were configuring Gradle to resolve dependencies over HTTP instead of HTTPS.

Weakness: CWE-829: Inclusion of Functionality from Untrusted Control Sphere Severity: High CVSS: 8.1 Detection: OpenRewrite

Reported-by: Jonathan Leitschuh Jonathan.Leitschuh@gmail.com **Signed-off-by:** Jonathan Leitschuh Jonathan.Leitschuh@gmail.com

Bug-tracker: <https://github.com/JLLeitschuh/security-research/issues/9>

Below the commit details is a table of repository changes:

Status	Repository	Modified	Result
No changes	lucene-gosen/lucene-gosen	about 22 hours ago	
Completed	sonalake/swagger-changelog-gradle-plugin	1 day ago	View commit
Completed	SmartReceipts/SmartReceiptsLibrary	1 day ago	View commit
Completed	jmad/jmad-core	1 day ago	View commit
Completed	sitewhere/sitewhere	1 day ago	View commit
Completed	ning377/UnblockMusicPro_Xposed	1 day ago	View commit
Completed	Mocha-LQiuJing	1 day ago	View commit

The screenshot shows two GitHub pull request pages. The top page is for `sonalake/swagger-changelog-gradle-plugin` with the title "[SECURITY] Use HTTPS to resolve dependencies in Gradle Build #13". The bottom page is for `sitewhere/sitewhere` with the title "[SECURITY] Use HTTPS to resolve dependencies in Gradle Build #982".

The `sitewhere/sitewhere` page includes a diagram illustrating the security fix:

- Original Connection:** A diagram showing a "Build Tool" connected to a "WWW" server.
- New Connection:** A diagram showing a "Build Tool" connected to a "Man in the Middle" (represented by a red skull icon), which is then connected to the "WWW" server.
- Text:** "This is a security fix. The build files indicate this leaves your build computer or CI/CD system vulnerable to a Man in the Middle (MITM) attack." "This vulnerability has a CVSS v3.0 Base Score of 8.1/10." "MITM attacks against HTTP are increasingly common, for example Comcast is known to have done it to their own users."

Use Files#createTempDirectory

REPLAY

SUMMARY

Status	Started	Estimated time savings	Repositories searched	Repositories changed	Files searched	Files changed
Finished	about 2 hours ago	17 hours, 50 minutes	6.21k	80	1.26M	107

Q Search...

Hide no results

SELECT ALL WITH RESULTS

<input type="checkbox"/>	Status	Repository	Branch	Total results	Files searched	Actions
<input type="checkbox"/>	Finished	broadinstitute/picard	master	4	6,501	DIFF
<input type="checkbox"/>	Finished	jenkinsci/google-oauth-plugin	develop	3	218	DIFF
<input type="checkbox"/>	Finished	andyglick/jenesis4java	master	3	279	DIFF
<input type="checkbox"/>	Finished	salesforce/ImageOptimization	master	3	244	DIFF
<input type="checkbox"/>	Finished	sonatype/plexus-archiver	master	3	580	DIFF
<input type="checkbox"/>	Finished	zereturnaround/zt-zip	master	3	230	DIFF
<input type="checkbox"/>	Finished	jenkinsci/acceptance-test-harness	master	3	786	DIFF
<input type="checkbox"/>	Finished	jenkinsci/jacoco-plugin	master	3	333	DIFF
<input type="checkbox"/>	Finished	smacke/jaydio	master	2	154	DIFF
<input type="checkbox"/>	Finished	jaltekruse/OpenNotebook	master	2	438	DIFF
<input type="checkbox"/>	Finished	native4java/BridJ	master	2	627	DIFF
<input type="checkbox"/>	Finished	jenkinsci/backend-jpi-create	master	2	134	DIFF
<input type="checkbox"/>	Finished	libgdx/libgdx	master	2	3,718	DIFF
<input type="checkbox"/>	Finished	koraktor/mavanagaiata	master	2	75	DIFF
<input type="checkbox"/>	Finished	Graylog2/JadConfig	master	2	357	DIFF
<input type="checkbox"/>	Finished	jbossas/jboss-vfs	master	2	279	DIFF

[Home](#) > [Recent commits](#) > Commit job 1449e2d1-7e24-4d78-9798-ba06caa1c1a2

Commit results

Recipe e52VD	Success 80%	Progress 100%	Started about 6 hours ago
Commit title vuln-fix: Temporary Directory Hijacking or Information Disclosure			
Commit messages			
This fixes either Temporary Directory Hijacking, or Temporary Directory Local Information Disclosure.			
Weakness: CWE-379: Creation of Temporary File in Directory with Insecure Permissions Severity: High CVSS: 7.3 Detection: CodeQL & OpenRewrite (https://public.moderne.io/recipes/org.openrewrite.java.security.UseFilesCreateTempDirectory)			
Reported-by: Jonathan Leitschuh Jonathan.Leitschuh@gmail.com Signed-off-by: Jonathan Leitschuh Jonathan.Leitschuh@gmail.com			
Bug-tracker: https://github.com/JLLeitschuh/security-research/issues/10			

Q Search...

[RERUN FAILED JOBS](#)

Status ↑	Repository	Modified	Result
COMPLETED	sanity/tahrir	about 6 hours ago	View commit
COMPLETED	broadinstitute/picard	about 6 hours ago	View commit
COMPLETED	Anuken/Arc	about 5 hours ago	View commit
COMPLETED	talsma-ict/umldoclet	about 5 hours ago	View commit
COMPLETED	jenkinsci/jenkins-test-harness	about 5 hours ago	View commit
COMPLETED	searls/jasmine-maven-plugin	about 5 hours ago	View commit
COMPLETED	vert-x3/vertx-amqp-bridge	about 5 hours ago	View commit
COMPLETED	reactor/reactor-netty	about 5 hours ago	View commit
COMPLETED	libgdx/libgdx	about 5 hours ago	View commit
COMPLETED	Karatemp/PublicationSign	about 5 hours ago	View commit

But there are more than just 7,000
repositories in the world

How do we find the other vulnerable projects?

CodeQL

CodeQL

100k+ OSS Projects Indexed
35k+ OSS Java Projects

https://github.com/moderneinc/jenkins-ingest

main | jenkins-ingest / repos.csv

tkvangorder Fixing issues with requested repos | Latest commit c6d166d 3 days ago | History

7 contributors

9877 lines (9877 sloc) | 433 KB

Raw | Blame

Search this file...

1	Offz/gpr-for-gradle	master	8	gradle
2	Opslab/opslabJutil	master	8	maven
3	105032013072/javaparser	master	8	maven
4	15189611/jumpAop	master	8	gradlew
5	18824863285/BaseFlutter	master	8	gradlew
6	1and1/cosmo	master	8	maven
7	1and1/reactive	master	8	maven
8	1c-syntax/bslls-dev-tools	develop	8	gradlew
9	275593469/study	master	8	maven
10	2dxgujan/AndroidTagGroup	master	8	gradlew
11	2pure/CodeDesign-HomeWork1	master	8	maven
12	3bleinaD/tdd-gradle-plugin	master	8	gradlew
13	3esi/dotnet-plugin	master	8	gradle
14	3esi/gitversion-plugin	master	8	gradle

Finally!

Let's generate some
Open Source Software
Pull Requests!

Bulk Pull Request Generation Statistics

Project	PR Generator	Pull Requests	Merge Rate
HTTP Download of Dependencies	Python Bot	1,596	40%
CVE-2019-16303: JHipster RNG Vulnerability	Python Bot + Moderne	3,467	2.3%
CVE-2020-8597: rhostname array overflow	Python Bot	1,885	7.6%
Temporary Directory Hijacking	Moderne	64	TBD
Partial Path Traversal	Moderne	32	TBD
Zip Slip	Moderne	100	TBD

Bulk Pull Request Generation Statistics

Project	PR Generator	Pull Requests	Merge Rate
HTTP Download of Dependencies	Python Bot	1,596	40%
CVE-2019-16303: JHipster RNG Vulnerability	Python Bot + Moderne	3,467	2.3%
CVE-2020-8597: rhostname array overflow	Python Bot	1,885	7.6%
Temporary Directory Hijacking	Moderne	64	TBD
Partial Path Traversal	Moderne	32	TBD
Zip Slip	Moderne	100	TBD

New Pull Requests Generated in 2022: 590+

Bulk Pull Request Generation Statistics

Project	PR Generator	Pull Requests	Merge Rate
HTTP Download of Dependencies	Python Bot	1,596	40%
CVE-2019-16303: JHipster RNG Vulnerability	Python Bot + Moderne	3,467	2.3%
CVE-2020-8597: rhostname array overflow	Python Bot	1,885	7.6%
Temporary Directory Hijacking	Moderne	64	TBD
Partial Path Traversal	Moderne	32	TBD
Zip Slip	Moderne	100	TBD

Personally Generated: 5,200+ Pull Requests



zeroturnaround / **zt-zip**

Public



Watch

Fork

Star 1.3k

Code

Issues 27

Pull requests 3

Actions

Projects

Security

Insights

Filters

is:pr is:open sort:updated-desc

Labels 2

Milestones 1

New pull request

Clear current search query, filters, and sorts

3 Open 37 Closed Merged

Open all

Author

Label

Projects

Milestones

Reviews

Assignee

Sort

[SECURITY] Fix Zip Slip Vulnerability

#149 opened 2 minutes ago by *JLLeitschuh*

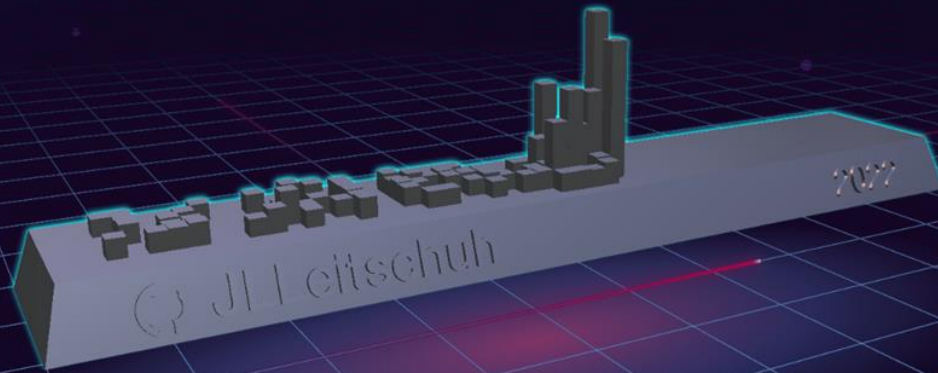
[SECURITY] Fix Partial Path Traversal Vulnerability

#148 opened 6 hours ago by *JLLeitschuh* updated 6 hours ago

[SECURITY] Fix Temporary Directory Hijacking or Information Disclosure Vulnerability

#147 opened 2 days ago by *JLLeitschuh*

ProTip! Find all pull requests that aren't related to any open issues with `-linked:issue`.



Best Practices for Bulk Pull Request Generation

Messaging!

All Software Problems are
People Problems
In Disguise

Lesson 1

Sign off all Commits

--signoff

Sign off on Commits

Signed-off-by: Jonathan Leitschuh <Jonathan.Leitschuh@gmail.com>

Sign off on Commits

Why?!

Sign off on Commits

“It was introduced in the wake of the SCO lawsuit, (and other accusations of copyright infringement from SCO, most of which they never actually took to court), as a Developers Certificate of Origin. It is used to say that you certify that you have created the patch in question, or that you certify that to the best of your knowledge, it was created under an appropriate open-source license, or that it has been provided to you by someone else under those terms.”

- [Stack Overflow](#)

TL;DR

Lawyers

Lesson 2


Be a good commitizen


Lesson 2

Be a good commitizen GPG Sign your Commits

Enjoy!

[Browse files](#)

 master

 **torvalds** committed on Aug 4, 2015 0 parents commit 9b0562595cc479ac8696110cb0a2d33f8f2b7d29 [patch](#) [diff](#)

No Whitespace

Showing 1 changed file with 10 additions and 0 deletions.

[Split](#)

[Unified](#)

10  README.md 

  ...

... @@ -0,0 +1,10 @@

```
1 Instructions on masquerading as other users in git:
2
3 ```bash
4 export GIT_AUTHOR_NAME="Linus Torvalds"
5 export GIT_AUTHOR_EMAIL="torvalds@linux-foundation.org"
6 export GIT_COMMITTER_NAME="$GIT_AUTHOR_NAME"
7 export GIT_COMMITTER_EMAIL="$GIT_AUTHOR_EMAIL"
8
9 git commit -m "Enjoy!"
10 ```
```

Lesson 3
SECOM
Commit Format

SECOM

```
1 vuln-fix: subject/header containing summary of changes in ~50 characters (Vuln-ID,)
2
3 Detailed explanation of the subject/header in ~75 words.
4 (what) Explain the security issue(s) that this commit is patching.
5 (why) Focus on why this patch is important and its impact.
6 (how) Describe how the issue is patched.
7
8 [For Each Weakness in Weaknesses:]
9 Weakness: weakness identification or CWE-ID.
10 Severity: severity of the issue (Low, Medium, High, Critical).
11 CVSS: numerical representation (0-10) of the vulnerability severity.
12 Detection: method used to detect the issue (Tool, Manual, Exploit).
13 Report: http://link-to-report/
14 Introduced in: commit hash.
15 [End]
16
17 Reported-by: reporter name 1 <reporter-email-1@host.com>
18 Reported-by: reporter name 2 <reporter-email-2@host.com>
19 Signed-off-by: your name <your-email@yourhost.com>
20
21 [If you use an issue tracker, add reference to it here:]
22 [if external issue tracker:]
23 Bug-tracker: https://link-to-bug-tracker/id
24
25 [if github used as issue tracker:]
26 Resolves: #123
27 See also: #456, #789
```


Lesson 4

There are risks using your personal GitHub Account

Anyone here familiar with
GitHub's
Angry Unicorn?



This page is taking way too long to load.

Sorry about that. Please try refreshing and contact us if the problem persists.

[Contact Support](#) — [GitHub Status](#) — [@githubstatus](#)



This was my GitHub Profile Page for most of 2020

Lesson 5

Coordinate with GitHub

Before Attempting

Reach out to GitHub!

SecurityLab@github.com

Lesson 5

Consider the Implications



Is this responsible disclosure?



#11 opened 4 hours ago



updated 35 minutes ago

Conclusion

As Security Researchers

We have an obligation to society

We know these vulnerabilities are out there

“For every 500 developers
you have one security
researcher.”

- GitHub 2020

“ We can fix it. We have the technology. OK. We need to create the technology. Alright. The policy guys are mucking with the technology. Relax. WE'RE ON IT.

- Dan Kaminsky (1979 – 2021)

Sound Bytes

- Learn CodeQL! Seriously! It's an incredibly powerful language!
- Contribute to OpenRewrite! Deploy your security fixes at scale!
- Join the GitHub Security Lab & OpenRewrite Slack Channels!

Thanks



Lidia Giuliano

Shyam Mehta