

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

- Jonathan Leitschuh -



# 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


Spoilers!




[Created](#)[Assigned](#)[Mentioned](#)[Review requests](#)[Commented](#)[Yours](#)


✓ 103 Open 49 Closed Merged


Open all Visibility Organization Sort


 arquillian/arquillian-cube [SECURITY] Fix Zip Slip Vulnerability ✕  
#1250 opened 6 days ago by [JLLeitschuh](#) · 1 review approval updated 18 hours ago


 jenkinsci/custom-war-packager [SECURITY] Fix Zip Slip Vulnerability ✕ 3  
#239 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago


 mgarin/webjars [SECURITY] Fix Zip Slip Vulnerability  
#705 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago

 mawensen/HRAAdmin-Backend [SECURITY] Fix Zip Slip Vulnerability  
#1 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago

 javaparser/javaparser [SECURITY] Fix Zip Slip Vulnerability ✓ 1  
#3684 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago

 forge/furnace [SECURITY] Fix Zip Slip Vulnerability 2  
#132 opened 6 days ago by [JLLeitschuh](#) · Changes requested updated 6 days ago


 Fernando-Cos/App\_DSpace\_teste [SECURITY] Fix Zip Slip Vulnerability  
#1 opened 6 days ago by [JLLeitschuh](#)

 YANG-DB/yang-db [SECURITY] Fix Zip Slip Vulnerability ✓ 1  
#143 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago

 stoicflame/enunciate [SECURITY] Fix Zip Slip Vulnerability  
#1132 opened 6 days ago by [JLLeitschuh](#) updated 6 days ago 2.15.0

 isislab-unisa/sof [SECURITY] Fix Zip Slip Vulnerability  
#2 opened on Jul 29 by [JLLeitschuh](#) updated 6 days ago

 unifi/unifi [SECURITY] Fix Zip Slip Vulnerability  
#35 opened 6 days ago by [JLLeitschuh](#)

 restx/restx [SECURITY] Fix Zip Slip Vulnerability  
#345 opened 6 days ago by [JLLeitschuh](#)

 mulesoft-labs/rhinodo [SECURITY] Fix Zip Slip Vulnerability ✓  
#8 opened 6 days ago by [JLLeitschuh](#)

Zip Slip  
152 Pull Requests!

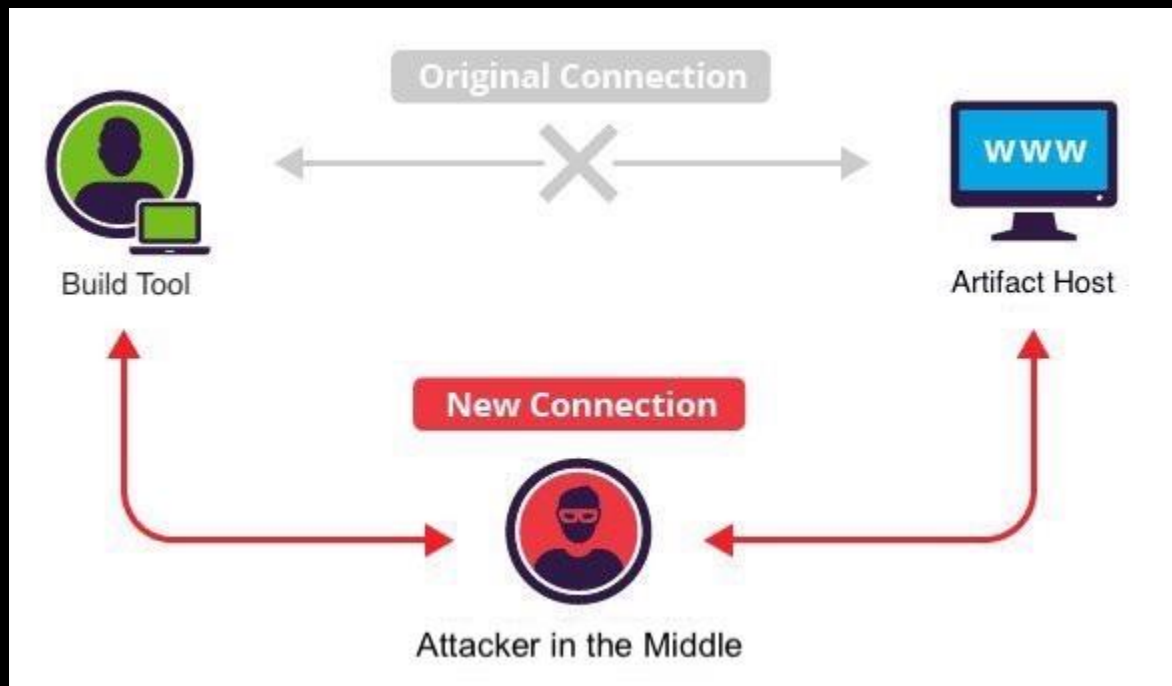
It Started  
With a Simple Vulnerability

```
// 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)

Tue 30 April 2019 Terry Yanko

Beginning January 15, 2020, The Central Repository will no longer support communication over HTTP. <http://repo1.maven.org> and <http://repo.maven.apache.org/> will no longer resolve, and users will need to update their builds to resolve dependencies over HTTPS.

### Why is this happening?

- It's time to make this change. Maven Central has taken steps in the past to improve its security posture. In May 2018, we announced the end of [support for TLSv1.1 and below](#). Almost a year later, deciding to deprecate support for unencrypted access to Maven Central is the logical continuation of this journey. We would like to credit Jonathan Leitschuh for pushing this initiative across the ecosystem. You can see his full writeup [here](#)

### My environment does not support HTTPS, what can I do?

- We recognize that for some of our users, there may be major technical limitations that prohibit making the switch to HTTPS, e.g. build environments still running JDK6. For those users, we will provide a separate domain to accommodate insecure traffic. As we approach the cut over date, we'll provide the exact HTTP URL to replace all your existing references to <http://repo1.maven.org> or <http://repo.maven.apache.org/>.

You can check out our [FAQ Page](#) or follow [@sonatype\\_ops](#) on Twitter for a more detailed schedule of changes as we get closer to January 15, 2020.



## Spring Blog

# Goodbye <http://repo.spring> (use [https](https://repo.spring.io))

ENGINEERING | ROB WINCH | SEPTEMBER 16, 2019 11 COMMENTS

In response to our [nohttp announcement](#), [Maven Central's announcement](#), and [JFrog's announcement](#), beginning January 15 2020, Spring's Maven Repository will no longer support HTTP. More concretely, <http://repo.spring.io> will not respond to requests. Users will need to ensure that they are using <https://repo.spring.io>

## Secure JCenter with HTTPS

By Baruch Sadogursky | April 30, 2021  
4 min read

SHARE: Facebook LinkedIn Twitter



Are you using Bintray JCenter to find and share public OSS JVM language packages? If so, we have some important news for you to help keep your builds running without interruption.

Starting in January 2020, **JCenter will only serve requests made with HTTPS**. From that point on, all requests made with HTTP will be denied and any builds that use a JCenter URL with the non-secure HTTP protocol will fail.

The TL;DR? Update your tools with a URL that uses HTTPS as soon as you can. That's all you really need to do to be certain all your builds will continue to run smoothly with JCenter.



## Decommissioning HTTP for Gradle Services

October 17, 2019 Jonathan Leitschuh Security

Starting in January 2020, Gradle services will only serve requests made with HTTPS. From that point on, all requests made with HTTP will be denied and any builds and artifact mirrors that use a Gradle URL with the non-secure HTTP protocol will fail.

If you are proxying our services through your own artifact servers like Artifactory or Nexus, you will need to ensure that you update your mirror configurations so they are using HTTPS instead of HTTP.

However!

“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 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()
```

CodeQL scans 100Ks of OSS Projects

# CodeQL

```
import java
import semmle.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()
```

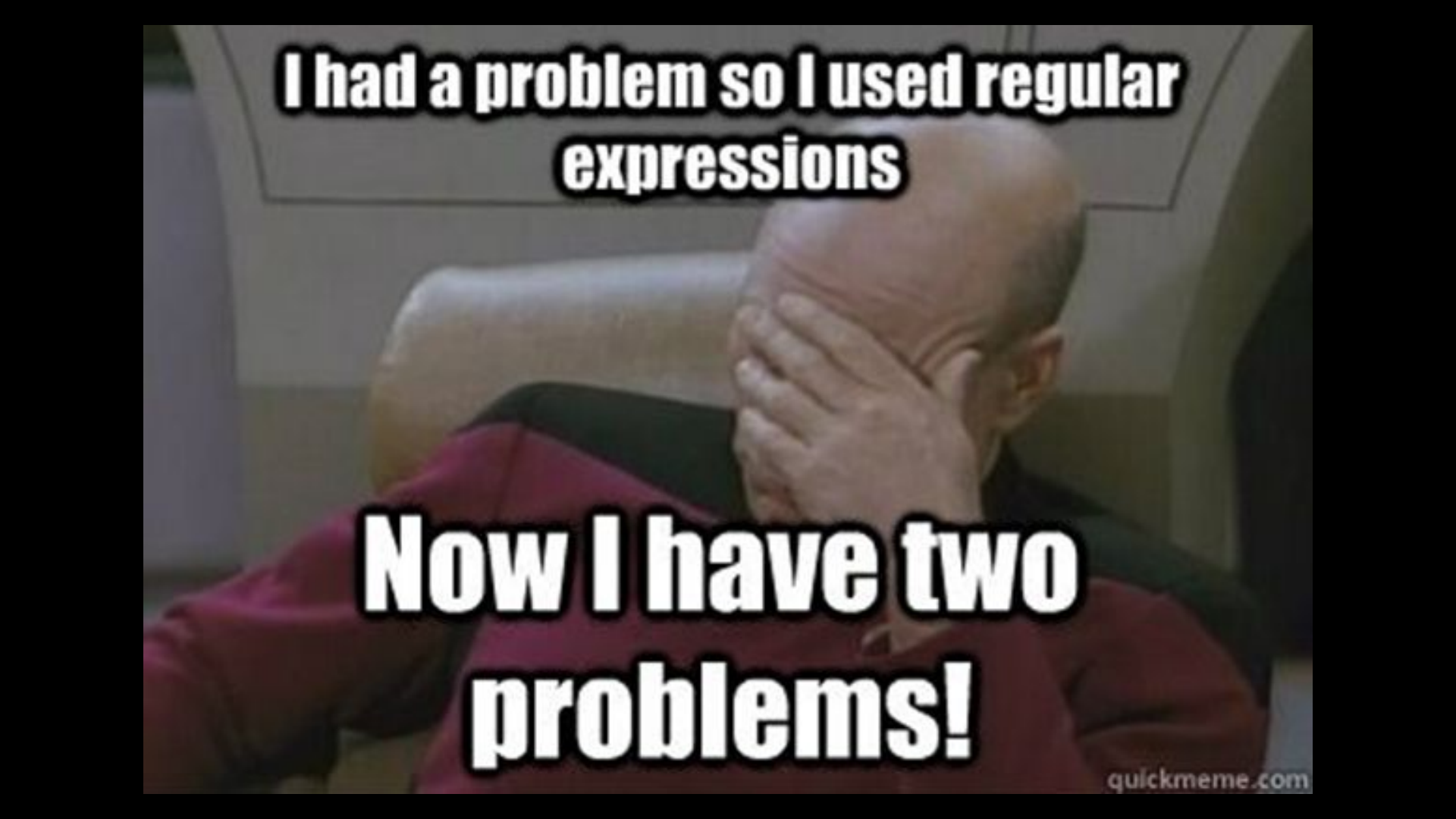
## \$2,300 Bounty

# Pull Request Generator Version 1

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



```
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'
```

A photograph of a man with a shaved head, wearing a maroon shirt, sitting in a white chair and covering his face with his hands in a gesture of embarrassment or frustration. The background is a plain, light-colored wall.

**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/zzzobspk [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/jeeecg [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-iots [SECURITY] Use HTTPS to resolve dependencies in Maven Build ✓	<span>Resolution/Stale</span> #1940 opened on Feb 11 by JLLeitschuh • Review required			17
xautix/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>
```

```
19 .....</repository>
20 .....<repository>
21 .....<id>onarandombox</id>
22 .....<url>https://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>https://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>
```

```
36 .....<!-- Has a copy of metrics R8-SNAPSHOT !-->
37 .....<repository>
38 .....<id>elmakers-repo</id>
39 .....<url>https://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>https://doodleproject.sourceforge.net/maven2/release</url>
48 .....<releases>
49 .....<enabled>true</enabled>
50 .....</releases>
```

@@ -353,11 +353,11 @@

```
353 .....<distributionManagement>
354 .....<repository>
```

```
353 .....<distributionManagement>
354 .....<repository>
```

HTTP Download of Dependencies

1,596

Pull Requests

~40%

Merged or Accepted

**\$4,000**

Thanks to the GitHub Security Lab!

There's more still out there

Showing 1 - 20 of 100+ files found in 372 milliseconds

<pre> 13 &lt;!--repository--&gt; 14 &lt;id&gt;codehaus&lt;/id&gt; 15 &lt;name&gt;Codehaus Maven Repository&lt;/name&gt; 16 &lt;url&gt;http://repository.codehaus.org/&lt;/url&gt; 17 &lt;/repository&gt; 18 &lt;/repositories&gt; </pre>	3 matches	View POM	Trunk
<pre> 84 &lt;!--repository--&gt; 85 &lt;id&gt;codehaus&lt;/id&gt; 86 &lt;name&gt;Codehaus Maven Repository&lt;/name&gt; 87 &lt;url&gt;http://repository.codehaus.org/codehaus/&lt;/url&gt; 88 &lt;/repository&gt; 89 &lt;/repositories&gt; 90 &lt;/release&gt; 91 &lt;/build&gt; </pre>	8 matches	View POM	master
<pre> 111 &lt;!--repository--&gt; 112 &lt;id&gt;apache.snapshots&lt;/id&gt; 113 &lt;name&gt;Apache Snapshot Repository - Maven&lt;/name&gt; 114 &lt;url&gt;http://repository.apache.org/snapshots/&lt;/url&gt; 115 &lt;layout&gt;default&lt;/layout&gt; 116 &lt;/repository&gt; 117 &lt;/repositories&gt; 118 &lt;/release&gt; 119 &lt;/build&gt; </pre>	10 matches	View POM	Trunk
<pre> 13 &lt;!--repository--&gt; 14 &lt;id&gt;apache2.snapshots&lt;/id&gt; 15 &lt;name&gt;Apache Snapshot Repository - Maven&lt;/name&gt; 16 &lt;url&gt;http://repository.apache.org/snapshots/&lt;/url&gt; 17 &lt;/repository&gt; 18 &lt;/repositories&gt; 19 &lt;/release&gt; 20 &lt;/build&gt; </pre>	10 matches	View POM	st-1.1.1
<pre> 39 &lt;!--repository--&gt; 40 &lt;id&gt;boss.maven&lt;/id&gt; 41 &lt;in&gt;boss.maven&lt;/in&gt; 42 &lt;name&gt;Boss Maven Repository&lt;/name&gt; 43 &lt;url&gt;http://repository.boss.in/news/content/groups/public/&lt;/url&gt; 44 &lt;/repository&gt; 45 &lt;/repositories&gt; 46 &lt;/release&gt; 47 &lt;/build&gt; </pre>	6 matches	View POM	Trunk
<pre> 213 &lt;!--repository--&gt; 214 &lt;id&gt;openjdk&lt;/id&gt; 215 &lt;name&gt;OpenJDK Snapshot Repository&lt;/name&gt; 216 &lt;url&gt;http://repository.openjdk.org/news2/&lt;/url&gt; 217 &lt;/repository&gt; 218 &lt;/repositories&gt; 219 &lt;/release&gt; 220 &lt;/build&gt; </pre>	7 matches	View POM	Trunk
<pre> 29 &lt;!--repository--&gt; 30 &lt;id&gt;apache2&lt;/id&gt; 31 &lt;name&gt;Apache Snapshot Repository&lt;/name&gt; 32 &lt;url&gt;http://repository.apache.org/snapshots/&lt;/url&gt; 33 &lt;/repository&gt; 34 &lt;/repositories&gt; 35 &lt;/build&gt; </pre>	11 matches	View POM	Trunk
<pre> 181 &lt;!--repository--&gt; 182 &lt;id&gt;apache.snapshots&lt;/id&gt; 183 &lt;name&gt;Apache Snapshot Repository&lt;/name&gt; 184 &lt;url&gt;http://repository.apache.org/snapshots/&lt;/url&gt; 185 &lt;/repository&gt; 186 &lt;/repositories&gt; 187 &lt;/release&gt; 188 &lt;/build&gt; </pre>	8 matches	View POM	Trunk
<pre> 46 &lt;!--repository--&gt; 47 &lt;id&gt;apache.snapshots&lt;/id&gt; 48 &lt;name&gt;Apache Snapshot Repository&lt;/name&gt; 49 &lt;url&gt;http://repository.apache.org/snapshots/&lt;/url&gt; 50 &lt;/repository&gt; 51 &lt;/repositories&gt; 52 &lt;/build&gt; </pre>	10 matches	View POM	Trunk

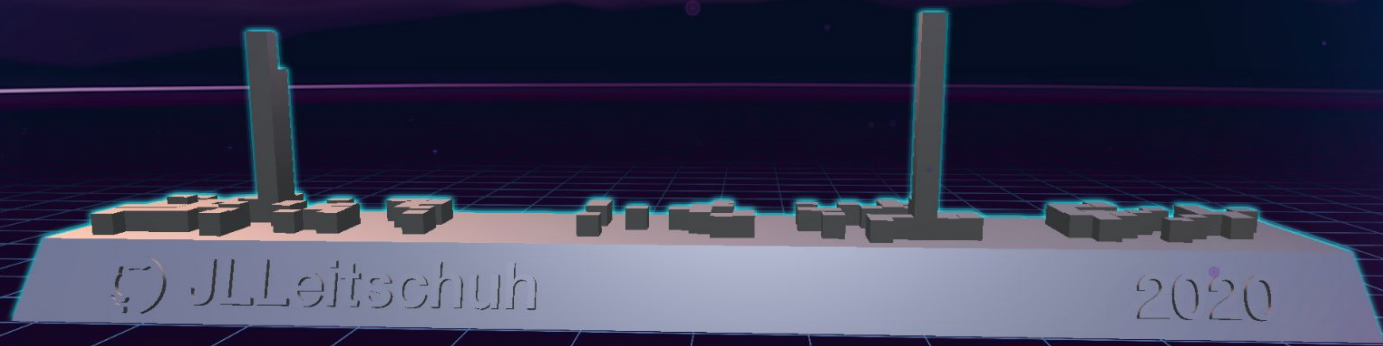
Showing 1 - 20 of 500 files found in 200 milliseconds

<pre> 24 &lt;!--log4jRepository--&gt; 25 &lt;id&gt;boss-snapshots&lt;/repository/id&gt; 26 &lt;name&gt;Boss Snapshot&lt;/repository/name&gt; 27 &lt;url&gt;http://repository.jboss.org/nexus/content/repositories/snapshots/&lt;/url&gt; 28 &lt;/log4jRepository--&gt; 29 30 &lt;/log4jRepository--&gt; </pre>	<p>21 matches Maven POM master</p>
<pre> 31 &lt;!--releng/org.eclipse.xosmantics.repository.pom.xml 32 33 &lt;artifactId&gt;org.eclipse.xosmantics.repository&lt;/artifactId&gt; 34 &lt;packaging&gt;eclipse-repository&lt;/packaging&gt; 35 36 37 38 &lt;!--repository--&gt; 39 &lt;repository&gt; 40 &lt;url&gt;http://download.eclipse.org/webtools/releng/repository&lt;/url&gt; </pre>	<p>21 matches Maven POM master</p>
<pre> 41 &lt;!--repository.pom.xml 42 43 &lt;artifactId&gt;repository&lt;/artifactId&gt; 44 &lt;name&gt;Verte&lt;/repository/name&gt; 45 &lt;packaging&gt;pom&lt;/packaging&gt; 46 &lt;description&gt;Eclipse Verte&lt;/repository&gt; manages Verte Information Models&lt;/description&gt; 47 &lt;url&gt;http://www.eclipse.org/verte/&lt;/url&gt; 48 49 50 </pre>	<p>18 matches Maven POM development</p>
<pre> 51 &lt;!--releng/org.eclipse.gef.runtime.repository.pom.xml 52 53 &lt;!--parent--&gt; 54 &lt;artifactId&gt;org.eclipse.gef.runtime.repository&lt;/artifactId&gt; 55 &lt;version&gt;1.15.0-SNAPSHOT&lt;/version&gt; 56 57 58 &lt;!--Eclipse.org/id--&gt; 59 &lt;url&gt;http://www.eclipse.org/nexus/content/groups/public/&lt;/url&gt; 60 &lt;/!--repository--&gt; </pre>	<p>7 matches Maven POM master</p>
<pre> 61 &lt;!--runtime.pom.xml 62 63 &lt;name&gt;Eclipse&lt;/name&gt; 64 &lt;url&gt;http://repository.jboss.org/licenses/lgpl.txt&lt;/url&gt; 65 &lt;/license&gt; 66 67 68 &lt;name&gt;Boss Public&lt;/repository&gt; Group&lt;/name&gt; 69 &lt;url&gt;http://repository.jboss.org/nexus/content/groups/public/&lt;/url&gt; 70 &lt;layout&gt;default&lt;/layout&gt; </pre>	<p>18 matches Maven POM master</p>
<pre> 71 &lt;!--releng/transaction.repository.pom.xml 72 73 &lt;!--parent--&gt; 74 &lt;artifactId&gt;org.eclipse.eef.transaction.repository&lt;/artifactId&gt; 75 &lt;version&gt;1.12.0-SNAPSHOT&lt;/version&gt; 76 77 78 &lt;!--Eclipse.org/id--&gt; 79 &lt;url&gt;http://www.eclipse.org/nexus/content/groups/public/&lt;/url&gt; 80 &lt;/!--repository--&gt; </pre>	<p>7 matches Maven POM master</p>
<pre> 81 &lt;!--general/org.eclipse.gemec.language-xml-schema-declaration.pom.xml 82 83 &lt;!--repository--&gt; 84 &lt;repository&gt; 85 &lt;id&gt;boss&lt;/id&gt; 86 &lt;url&gt;http://repository.jboss.org/nexus/content/groups/public/&lt;/url&gt; 87 &lt;/repository&gt; 88 &lt;/repository--&gt; 89 &lt;/log4jRepository--&gt; </pre>	<p>8 matches Maven POM master</p>
<pre> 90 &lt;!--releng/ty36.pom.xml 91 92 &lt;url&gt;http://maven.ty36.org/snapshots/&lt;/url&gt; 93 &lt;/repository--&gt; 94 &lt;/repository--&gt; 95 96 97 &lt;url&gt;http://maven.ty36.org/releases/&lt;/url&gt; 98 &lt;/repository--&gt; 99 &lt;/repository--&gt; </pre>	<p>8 matches Maven POM master</p>
<pre> 100 &lt;!--scripts/reqmgr.pom.xml 101 102 &lt;!--parent--&gt; 103 &lt;/parent--&gt; </pre>	<p>8 matches Maven POM develop</p>

More Pull Request Generation  
For this in the Future!



I got hooked on  
Bulk Pull Request Generation



JLLeitschuh

2020

I have a Problem

AD⚡HD

HIGHWAY TO *HEY LOOK A SQUIRRELI*

I was finding too many security vulnerabilities!

<aws2/carbon-mediation>/components/./utils/SynapseArtifactInitUtils.java

```
↑ 1:254
255
256 // If the entry is a file, write the file
257 copyInputStream(cipIn.getInputStream(entry),
    new BufferedOutputStream(new FileOutputStream(destPath + entry.getName())));
258
259 }
260 }
↓ 268-276
```

Unsanitized archive entry, which may contain '!', is used in a file system operation. [Show paths](#)

<apache/druid>/indexing-hadoop/./Indexer/JobHelper.java

```
↑ 1:768
769
770 try (ZipInputStream in = new ZipInputStream(fileSystem.open(cip, 1 << 13)); {
771     for (ZipEntry entry = in.getNextEntry(); entry != null; entry = in.getNextEntry()) {
772         final String fileName = entry.getName();
773
774         final String outputPath = new File(outDir, fileName).getAbsolutePath();
775
776     }
777 }
↓ 774-884
```

Unsanitized archive entry, which may contain '!', is used in a file system operation. [Show paths](#)

<HongZhaokua/jstarcraft-core>/jstarcraft-core-common/./utility/PressUtility.java

```
↑ 1:179
180
181 ArchiveEntry archiveEntry;
182 while (null != (archiveEntry = archiveInputStream.getNextEntry())) {
183     File file = new File(toDirectory, archiveEntry.getName());
184
185     try (FileOutputStream fileOutputStream = new FileOutputStream(file)) {
186         int length = -1;
187
188     }
189
190
191
192 ArchiveEntry archiveEntry;
193 while (null != (archiveEntry = archiveInputStream.getNextEntry())) {
194     File file = new File(toDirectory, archiveEntry.getName());
195
196     try (FileOutputStream fileOutputStream = new FileOutputStream(file)) {
197         int length = -1;
198
199     }
200
201 }
↓ 223-233
```

Unsanitized archive entry, which may contain '!', is used in a file system operation. [Show paths](#)

Unsanitized archive entry, which may contain '!', is used in a file system operation. [Show paths](#)

<deepjava.library>/api/./repository/AbstractRepository.java

```
↑ 1:245
246
247 TarArchiveEntry entry;
248 while ((entry = tis.getNextTarEntry()) != null) {
249     String entryName = entry.getName();
250
251     if (entryName.contains(".")) {
252         throw new IOException("Malicious zip entry: " + entryName);
253     }
254 }
↓ 251-373
```

Unsanitized archive entry, which may contain '!', is used in a [ 2 Values ]. [Show paths](#)

<deepjava.library>/api/./util/ZipUtils.java

```
↑ 1:39
40
41 ZipEntry entry;
42 while ((entry = zis.getNextEntry()) != null) {
43     String name = entry.getName();
44
45     if (name.contains(".")) {
46         throw new IOException("Malicious zip entry: " + name);
47     }
48 }
↓ 45-103
```

Unsanitized archive entry, which may contain '!', is used in a [ 2 Values ]. [Show paths](#)

<cybertaxonomy/cdm1lib>/cdm1lib-ext/./scratchpads/ScratchpadsService.java

```
↑ 1:108
109
110
111 System.out.println("Extracting: " + ze);
112
113 FileOutputStream fos = new FileOutputStream(ze.getName());
114
115 }
↓ 113-183
```

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!



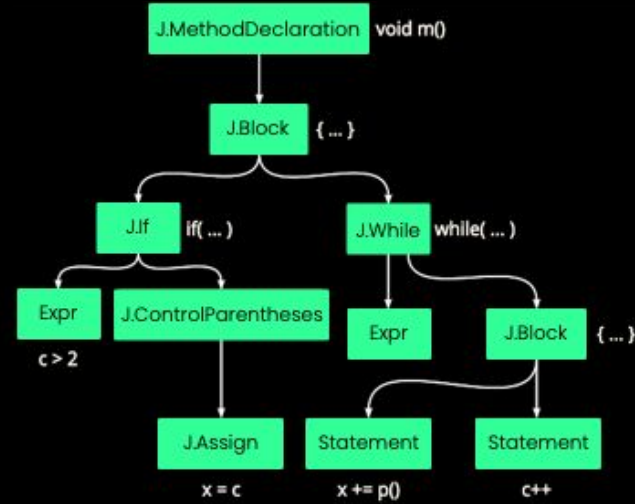
Automated Accurate Transformations  
at a  
Massive Scale

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.

OpenRewrite

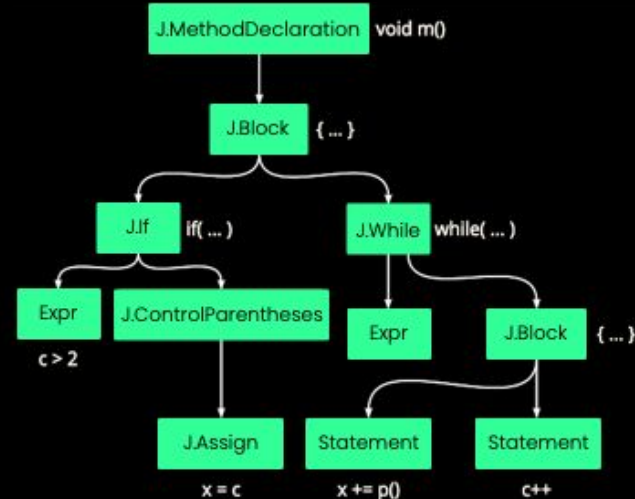
# Abstract Syntax Tree (AST)

```
... /** 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)

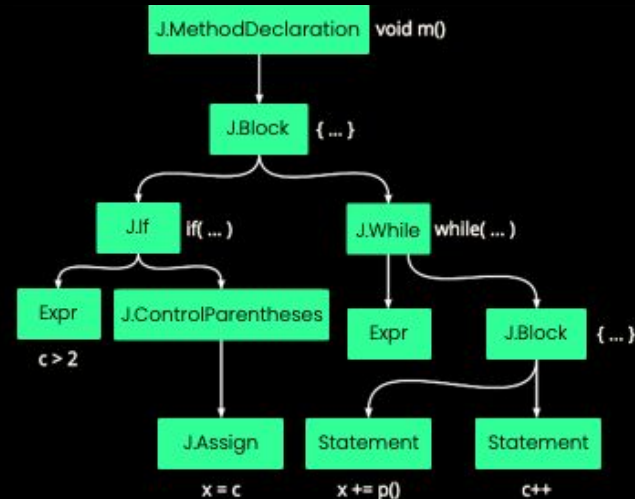
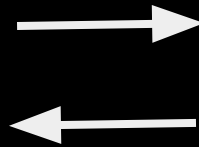
```
... /** myMethod */
... void m(){
...     if (c > 2) {
...         // c is more than 2
...         x = c;
...     }
...     while(c < 10) { // increment x
...         x += p();
...         c++;
...     }
... }
```



```
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

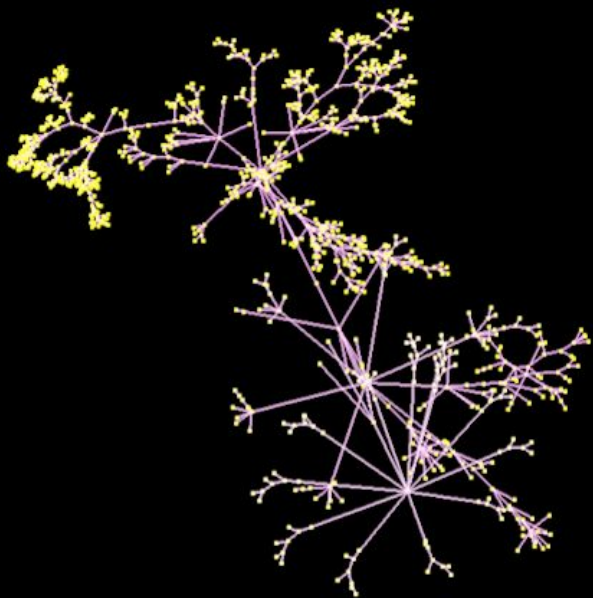
```
String name = entry.getName();
Path path = dir.resolve(name);
if (!path.normalize().startsWith(dir))
{
    ... throw new RuntimeException("Bad zip entry");
}
OutputStream os = Files.newOutputStream(path);
```

# Accurate Transformations Require Fully Type-attributed ASTs

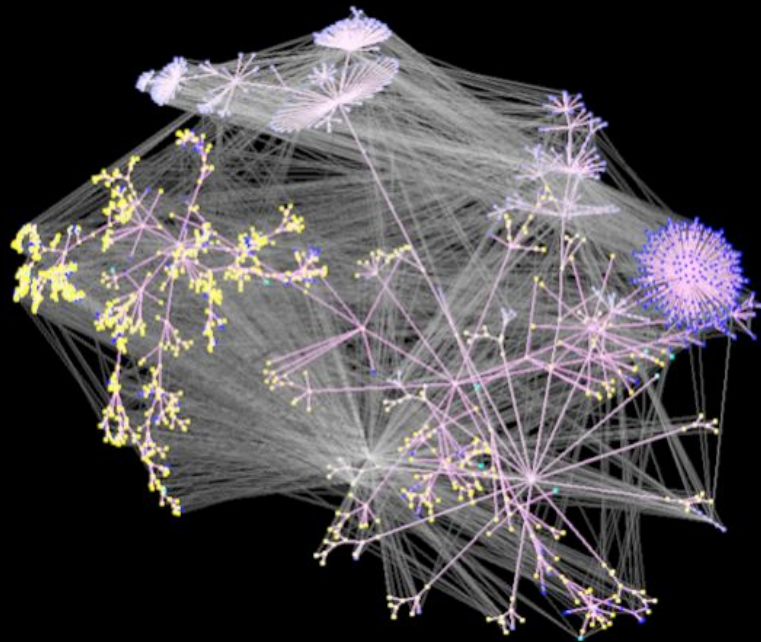
```
log.info("...");
```

Is that log4j, slf4j, LogBack?

# The OpenRewrite AST is both Syntactically and Semantically aware.



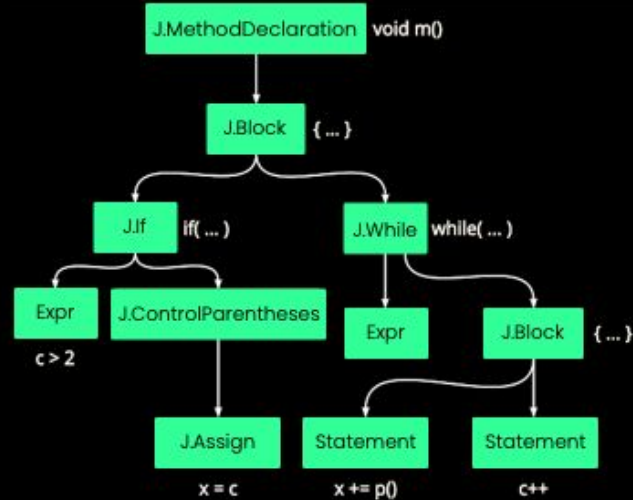
Syntax alone



With type attribution and formatting



# Even simple code produces complex AST



# Even simple code produces complex AST





**ONE DOES NOT SIMPLY**

**FORMAT THE WHOLE  
SOURCE FILE**

With the ability to Transform Code

How can we transform source files while  
preserving the style?

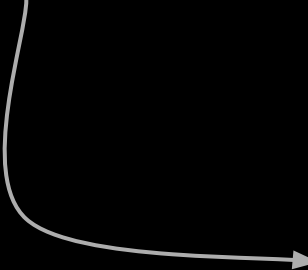
## Open Rewrite

- Automatically Detects the the Code Style during Parsing
- Provides a Templating Engine to add New Source Code
- Auto-format applies the detected style

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

# Transform ASTs using AutoFormat and the JavaTemplate

```
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);  
    }  
}
```

## Easily transform ASTs using AutoFormat and the JavaTemplate

```
String template =  
"if(!#{any(java.nio.file.Path)}.normalize().startsWith(#{any(java.nio.file.Path)})) {throw new  
RuntimeException(\"Bad zip entry\");}"
```

```
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);  
    }  
}
```



## Easily transform ASTs using AutoFormat and the JavaTemplate

```
String template =
    "if(!#{any(java.nio.file.Path)}.normalize().startsWith(#{any(java.nio.file.Path)})) {throw new
    RuntimeException(\"Bad zip entry\");}"

block = maybeAutoFormat(
    block, block.withTemplate(JavaTemplate.builder(this::getCursor, template).build(),
        resolvePathStatement.getCoordinates().after(),
        zipEntryArg, parentDirArg),
    ctx);
```

```
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);
    }
}
```

## Easily transform ASTs using AutoFormat and the JavaTemplate

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

## Easily transform ASTs using AutoFormat and the JavaTemplate

```
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);  
    }  
}
```

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

```
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);  
    }  
}
```

Data and Control flow are new additions to  
rewrite....

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**

# 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;

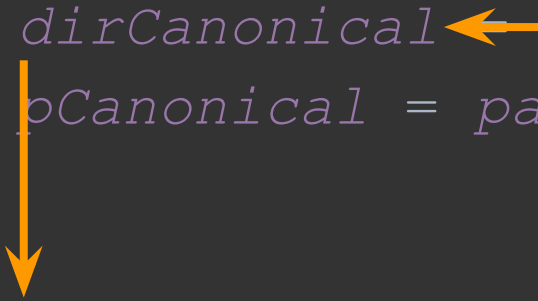
if (!dirCanonical
    .startsWith(pCanonical)) {
    ...
}
```

## Partial Path Traversal - Data Flow

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

# Partial Path Traversal - Data Flow

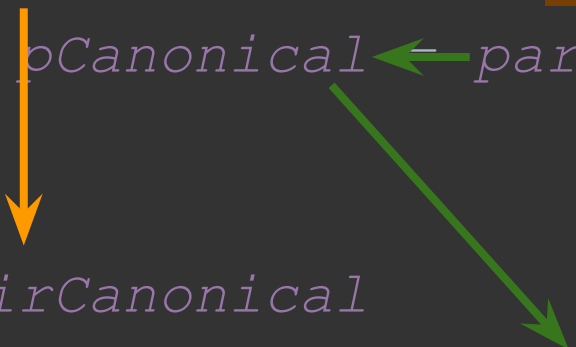
```
String dirCanonical ← dir.getCanonicalPath();  
String pCanonical = parent.getCanonicalPath() +  
                    File.separatorChar;  
  
if (!dirCanonical  
    .startsWith(pCanonical)) {  
    ...  
}
```

A diagram illustrating data flow. A yellow arrow points from the expression `dir.getCanonicalPath()` in the first line to the variable `dirCanonical`. A second yellow arrow points from `dirCanonical` down to the `!dirCanonical.startsWith(pCanonical)` condition in the `if` statement.

# 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;
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?



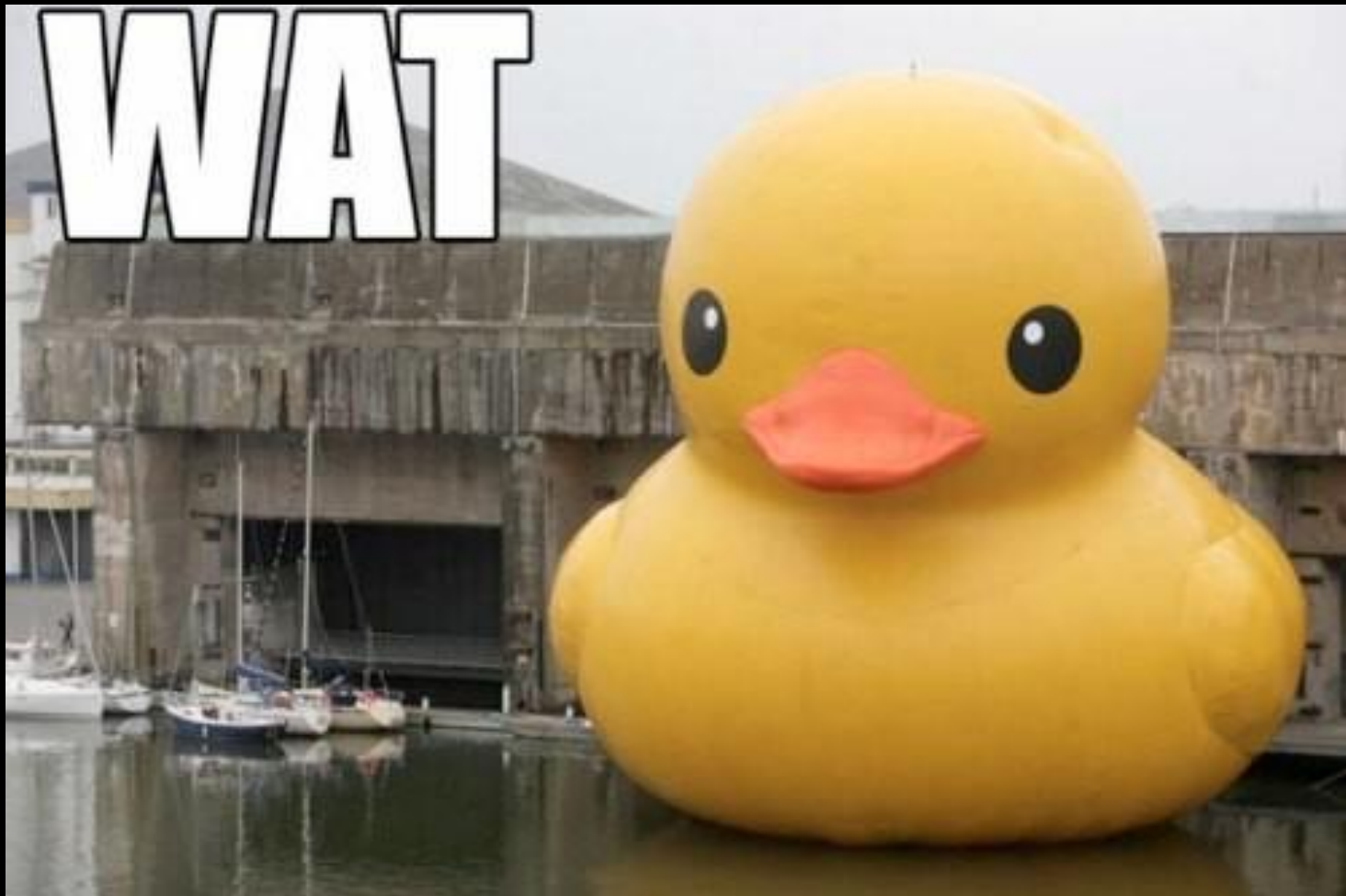
## Aside: Email with AWS Security Team

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

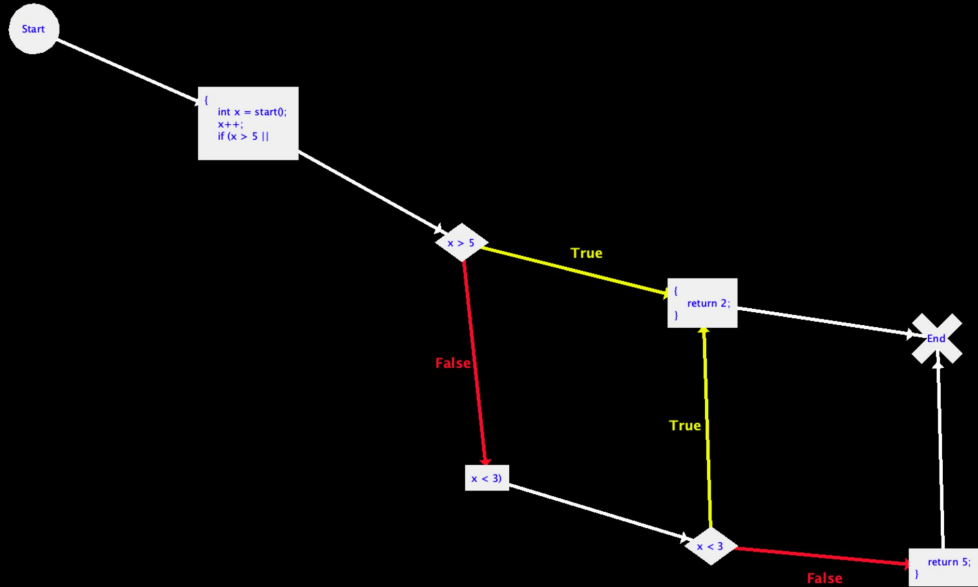
```
File f = new File(destination, e.getName());
IOUtils.copy(
    zip.getInputStream(e),
    new FileOutputStream(f)
);
```

```
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

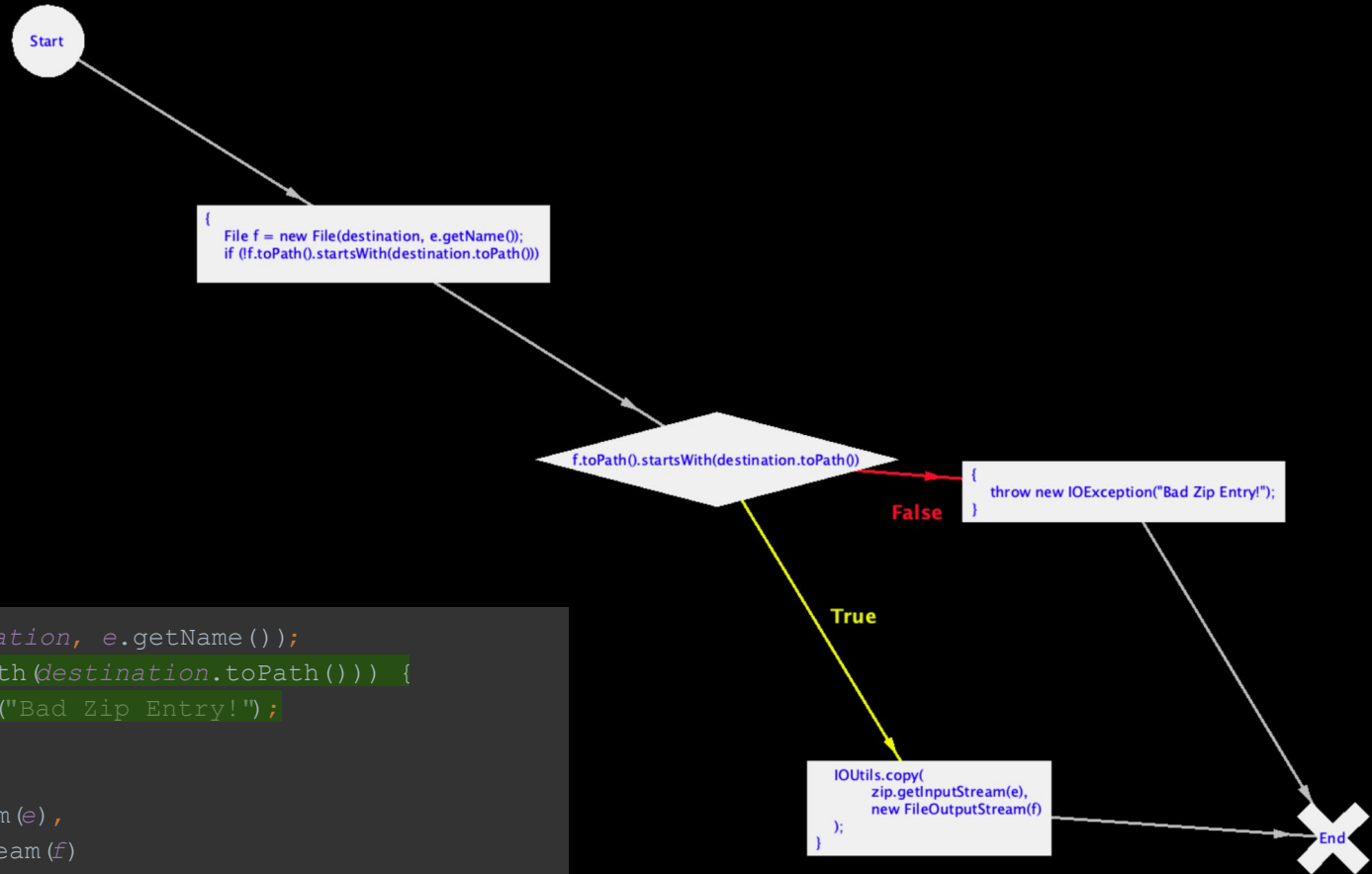
```
abstract class Test {  
    abstract int start();  
    int test() {  
        int x = start();  
        x++;  
        if (x > 5 || x < 3) {  
            return 2;  
        }  
        return 5;  
    }  
}
```



# 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();
```

# Zip Slip - Putting it all together

src/test/java/software/amazon/neptune/csv2rdf/Csv2RdfIntegrationTest.java

@@ -212,8 +212,12 @@ org.openrewrite.java.security.ZipSlip

```
212 212         ZipEntry zipEntry = zis.getNextEntry();
213 213
214 214         while (zipEntry != null) {
215 +             final Path zipEntryPath = outputDirectory.resolve(zipEntry.getName());
216 +             if (!zipEntryPath.normalize().startsWith(outputDirectory)) {
217 +                 throw new RuntimeException("Bad zip entry");
218 +             }
219 219             try (FileOutputStream fos = new FileOutputStream(
220 +                 outputDirectory.resolve(zipEntry.getName()).toFile()); {
221 221                 zipEntryPath.toFile()); {
222 222                 int len;
223 223                 while ((len = zis.read(buffer)) > 0) {
224 224                     fos.write(buffer, 0, len);
225 225                 }
226 226             }
227 227         }
228 228     }
229 229 }
```

Pull Request Generation!

**GOT SECURITY VULNERABILITIES?**



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



# Problems with Pull Request Generation

How fast can we generate  
Pull Requests?

## Pull Request Generation Steps

# 1. Checkout (ie. Download) code Repository

## Pull Request Generation Steps

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

## Pull Request Generation Steps

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

## 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

## 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

## 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



## 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

# 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

## 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

Let's talk about...  
GitHub's API Rate Limiter

# Github Documentation

“If you're making a large number of POST, PATCH, PUT, or DELETE requests for a single user or client ID, wait at least one second between each request.”

# Github Documentation

“When you have been limited, use the `Retry-After` response header to slow down. The value of the `Retry-After` header will always be an integer, representing the number of seconds you should wait before making requests again.”

Something New Appeared in 2022

# Github Documentation

“Requests that create content which triggers notifications, such as issues, comments and pull requests, may be further limited and will not include a `Retry-After` header in the response. Please create this content at a reasonable pace to avoid further limiting.”



# Github Documentation

“Requests that create content which triggers notifications, such as issues, comments and pull requests, may be further limited and will not include a *Retry-After* header in the response. Please create this content at a reasonable pace to avoid further limiting.”

# Github Documentation

“Requests that create content which triggers notifications, such as issues, comments and pull requests, may be further limited and will not include a *Retry-After* header in the response. Please create this content at a reasonable pace to avoid further limiting.”

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

**THAT WOULD BE GREAT**

# 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



## 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.



## Spring Boot 2.x migration from Spring Boot 1.x

Migrates Spring Boot 1.x to 2.x including best practices.



## Migrate deprecated javax packages to jakarta

`org.openrewrite.java.migrate.JavaxMigrationToJakarta` 



## JUnit Jupiter migration from JUnit 4.x

Migrates JUnit 4.x tests to JUnit Jupiter.

It's not just your code that needs to be secure  
It's also the dependencies



# 800+ OpenRewrite Recipes including complete Framework Migrations

The image shows a screenshot of the Moderne OpenRewrite Recipes interface. The interface is organized into several sections, each with a title and a list of recipes. Each recipe is represented by a circular icon and a text box containing the recipe name and a brief description.

**Analyze your code**

- 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**

**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 CVSS5: 8.1 Detection: OpenRewrite

**Reported-by:** Jonathan Leitschuh [Jonathan.Leitschuh@gmail.com](mailto:Jonathan.Leitschuh@gmail.com) **Signed-off-by:** Jonathan Leitschuh [Jonathan.Leitschuh@gmail.com](mailto: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	<a href="#">View commit</a>
Completed	SmartReceipts/SmartReceiptsLibrary	1 day ago	<a href="#">View commit</a>
Completed	jmad/jmad-core	1 day ago	<a href="#">View commit</a>
Completed	sitewhere/sitewhere	1 day ago	<a href="#">View commit</a>
Completed	nining377/UnblockMusicPro_Xposed	1 day ago	<a href="#">View commit</a>
Completed	Mocha-L/QuJing	1 day ago	<a href="#">View commit</a>

The GitHub interface shows two pull requests for the `sonalake/swagger-changelog-gradle-plugin` repository:

- Pull Request #13:** "[SECURITY] Use HTTPS to resolve dependencies in Gradle Build #13". It is a security fix for a high-severity vulnerability (CVSS v3.0 Base Score of 8.1/10) where build files were resolving dependencies over HTTP instead of HTTPS. The fix involves allowing a Man in the Middle (MTM) attacker to execute arbitrary code on the user's computer or CI/CD system.
- Pull Request #982:** "[SECURITY] Use HTTPS to resolve dependencies in Gradle Build #982". This is a security fix for a high-severity vulnerability in the `build.gradle` files. The build files indicate that the project is resolving dependencies over HTTP instead of HTTPS, leaving the build vulnerable to MTM attacks.

Both pull requests include a diagram illustrating the "Man in the Middle" attack. The diagram shows an "Original Connection" between a "Build Tool" and a "WWW" (Artifact Host). A "New Connection" is shown where a "Man in the Middle" (represented by a red heart icon) intercepts the communication between the Build Tool and the WWW.



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

## Commit results

Recipe	Success	Progress	Started
<b>e52VD</b>	<b>80%</b>	<b>100%</b>	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](mailto:Jonathan.Leitschuh@gmail.com) Signed-off-by: Jonathan Leitschuh [Jonathan.Leitschuh@gmail.com](mailto:Jonathan.Leitschuh@gmail.com)

Bug-tracker: <https://github.com/JLLeitschuh/security-research/issues/10>

🔍 Search...

[🔄 RERUN FAILED JOBS](#)

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

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	0ffz/gpr-for-gradle	master	8	gradle
2	0opslab/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	2dxgujun/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



# CodeQL: Partial Path Traversal

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

```
class MethodStringStartsWith extends Method {  
  MethodStringStartsWith() {  
    this.getDeclaringType() instanceof TypeString and  
    this.hasName("startsWith")  
  }  
}  
  
class MethodFileGetCanonicalPath extends Method {  
  MethodFileGetCanonicalPath() {  
    this.getDeclaringType() instanceof TypeFile and  
    this.hasName("getCanonicalPath")  
  }  
}  
  
class MethodAccessFileGetCanonicalPath extends MethodAccess {  
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }  
}  
  
abstract class FileSeparatorExpr extends Expr { }  
  
class SystemPropFileSeparatorExpr extends FileSeparatorExpr {  
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }  
}  
  
class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {  
  StringLiteralFileSeparatorExpr() {  
    this.getValue().matches("/") or this.getValue().matches("%\\")  
  }  
}  
  
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {  
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\ " }  
}  
  
class FileSeparatorAppend extends AddExpr {  
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }  
}  
  
predicate isSafe(Expr expr) {  
  DataFlow::localExprFlow(any(Expr e |  
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr  
  ), expr)  
}  
  
from MethodAccess ma  
where  
  ma.getMethod() instanceof MethodStringStartsWith and  
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and  
  not isSafe(ma.getArgument(0))  
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

With the list of vulnerable projects in hand!

Finally!

Let's generate some  
Open Source Software  
Pull Requests!

searls / jasmine-maven-plugin Public

Code Issues Pull requests 12 Actions Projects Wiki Security Insights

## [SECURITY] Fix Temporary Directory Hijacking or Information Disclosure Vulnerability #530

Edit Code

Open searls:master ← BulkSecurityGeneratorProjectV2:fix/JLL/temporary\_directory\_hijacking\_or\_temporary\_directory\_information\_disclosure

Conversation 3 Commits 1 Checks 0 Files changed 2 -3

JLLeitschuh on Aug 9

### Security Vulnerability Fix

This pull request fixes either 1. Temporary Directory Hijacking or Information Disclosure Vulnerability, which existed in this project.

#### Preamble

The system temporary directory is shared between all users interacting with the system temporary directory must be careful that the correct file permissions are set.

This PR was generated because the following chain of calls is vulnerable.

```
File.createTempFile(..) -> file.delete() -> either file.delete() or file.delete()
```

#### Impact

This vulnerability can have one of two impacts depending upon the configuration of the system.

- Temporary Directory Information Disclosure - Informatic malicious actor co-resident on the same machine to view files in the system temporary directory.

apache / maven-build-cache-extension Public

Code Pull requests 2 Actions Security Insights

## [SECURITY] Fix Zip Slip Vulnerability #23

Merged by gnodet apache:master ← BulkSecurityGeneratorProjectV2:fix/JLL/zip-slip-vulnerability 6 days ago

JLLeitschuh on Jul 29 - edited

### Security Vulnerability Fix

This pull request fixes a Zip Slip vulnerability either due to an insufficient, or missing guard when unzipping a zip file. Even if you deem, as the maintainer of this project, this is not necessarily fixing a security vulnerability, valid security hardening.

#### Preamble

This issue allows a malicious zip file to potentially break out of the expected destination directory, write arbitrary locations on the file system. Overwriting certain files/directories could allow an attacker to achieve remote code execution on a target system exploiting this vulnerability.

#### Why?

The best description of Zip-Slip can be found in the white paper published by Snyk: [Zip Slip Vulnerability](#). But I had a guard in place, why wasn't it sufficient?

If the changes you see are a change to the guard, not the addition of a new guard, this is probably because the existing guard was insufficient to prevent a Zip Slip vulnerability due to a partial path traversal vulnerability.

asciidoctor / asciidoctor-maven-plugin Public

Code Issues 29 Pull requests 1 Actions Projects Security Insights

## [SECURITY] Fix Partial Path Traversal Vulnerability #587

Open asciidoctor:main ← BulkSecurityGeneratorProjectV2:fix/JLL/partial-path-traversal-vulnerability

Conversation 0 Commits 2 Checks 0 Files changed 3 +21 -14

JLLeitschuh on Jul 29 - edited

### Security Vulnerability Fix

This pull request fixes a partial-path traversal vulnerability due to an insufficient path traversal guard.

Even if you deem, as the maintainer of this project, this is not necessarily fixing a security vulnerability, it is still a valid security hardening.

#### Preamble

This issue allows a malicious actor to potentially break out of the expected directory. The impact is limited to sibling directories. For example, `userControlled.getCanonicalPath().startsWith("/usr/out")` will allow an attacker to access a directory with a name like `/usr/outnot`.

#### Impact

This issue allows a malicious actor to potentially break out of the expected directory. The impact is limited to sibling directories. For example, `userControlled.getCanonicalPath().startsWith("/usr/out")` will allow an attacker to access a directory with a name like `/usr/outnot`.

#### Why?

To demonstrate this vulnerability, consider `"/usr/outnot".startsWith("/usr/out")`. The check is bypassed although `/outnot` is not under the `/out` directory. It's important to understand that the terminating slash may be removed when using various `String` representations of the `File` object. For example, on Linux, `println(new File("/var"))` will print `/var`, but `println(new File("/var", ""))` will print `/var/`; however, `println(new File("/var", "").getCanonicalPath())` will print `/var`.

Reviewers -- review now

Still in progress? Convert to draft

Notifications Customize

You're receiving notifications because you authored the thread.

Unsubscribe

2 participants

# 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	25%
Partial Path Traversal	Moderne	50	22%
Zip Slip	Moderne	152	20%

# 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	25%
Partial Path Traversal	Moderne	50	22%
Zip Slip	Moderne	152	20%

**New Pull Requests Generated in 2022: 600+**

# 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	25%
Partial Path Traversal	Moderne	50	22%
Zip Slip	Moderne	152	20%

**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

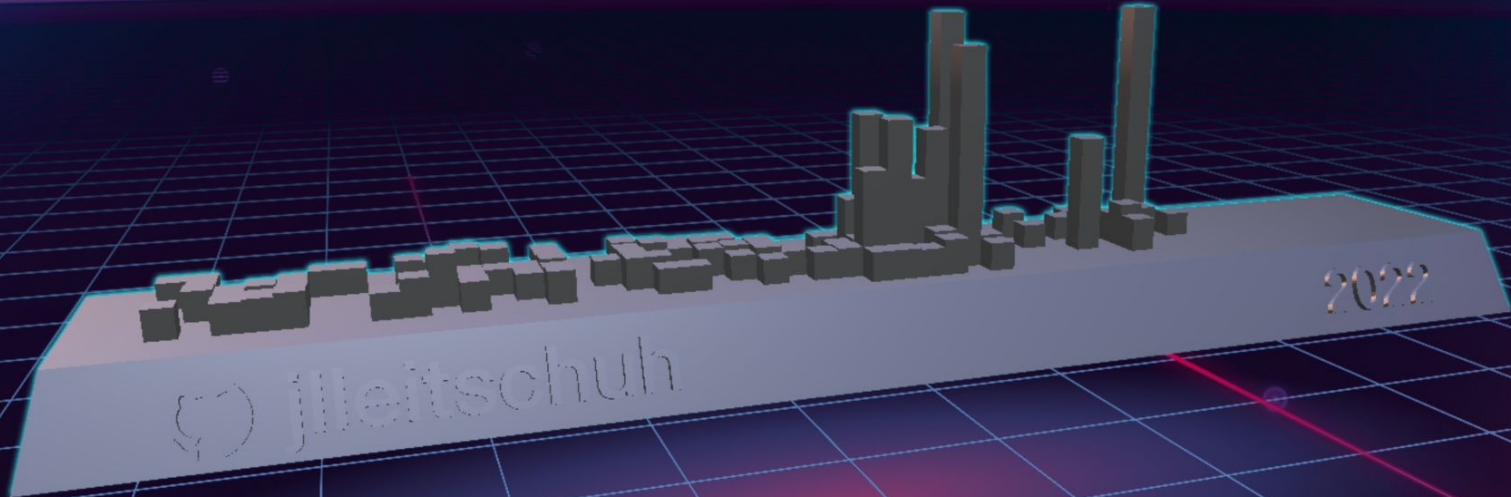
**[SECURITY] Fix Temporary Directory Hijacking or Information Disclosure Vulnerability**

#147 opened 2 days ago by *JLLeitschuh*

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



@jlleitschuh's 2022 GitHub Skyline



# 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**

Remember GitHub's Rate Limit?



Lesson 5

# Coordinate with GitHub

Before Attempting

Reach out to GitHub!

[SecurityLab@github.com](mailto:SecurityLab@github.com)



Lesson 6

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)

- 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!
- Join the Open Source Security Foundation (OSSF)!

Thanks



Lidia Giuliano

Shyam Mehta

@JLLeitschuh

Jonathan.Leitschuh@gmail.com

**WHY!!**



# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType() instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType() instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }

class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }
}

class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {
  StringLiteralFileSeparatorExpr() {
    this.getValue().matches("/") or this.getValue().matches("\\")
  }
}

class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\" }
}

class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }
}

predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}

from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType().instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType().instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod().instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

```
FileSeparatorExpr {
  getSystemProperty("file.separator") }

FileSeparatorExpr, StringLiteral {
  getValue().matches("%\\")
}

class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\ " }
}

class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }
}

predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType() instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType() instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}
```

```
class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() {
    this.getMethod() instanceof MethodFileGetCanonicalPath
  }
}

class FileSeparatorExpr {
  FileSeparatorExpr() {
    this.getSystemProperty("file.separator")
  }
}

class FileSeparatorExpr, StringLiteral {
  FileSeparatorExpr(StringLiteral l) {
    l.getValue().matches("%\\")
  }
}

class FileSeparatorExpr, CharacterLiteral {
  FileSeparatorExpr(CharacterLiteral c) {
    c.getValue() = "/" or this.getValue() = "\\\"
  }
}

class FileSeparatorExpr {
  FileSeparatorExpr() {
    this instanceof FileSeparatorExpr
  }
}
```

```
DataFlow::localExprFlow(any(Expr e)
  e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
), expr)
}
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType() instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType() instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}
```



# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType() instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType() instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }

class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this.getSystemProperty("file.separator") }
}

class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\\" }
}

class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }
}

predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}

from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

```
class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }
}
```

# CodeQL: Partial Path Traversal

```
abstract class FileSeparatorExpr extends Expr { }
```

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType().instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType().instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod().instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }

class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }
}

class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {
  StringLiteralFileSeparatorExpr() {
    this.getValue().matches("%/") or this.getValue().matches("%\\")
  }
}

class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\ " }
}

class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand().instanceof FileSeparatorExpr }
}

predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}

from MethodAccess ma
where
  ma.getMethod().instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType().instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType().instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod().instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }

class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }
}
```

```
class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {
  StringLiteralFileSeparatorExpr() {
    this.getValue().matches("%/") or this.getValue().matches("%\\")
  }
}
```

```
class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }
}
```

```
class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand().instanceof FileSeparatorExpr }
}
```

```
predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}
```

```
from MethodAccess ma
where
  ma.getMethod().instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and
  not isSafe(ma.getArgument(0)))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType().instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType().instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod().instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }

class SystemPropFileSeparatorExpr extends FileSeparatorExpr {
  SystemPropFileSeparatorExpr() { this = getSystemProperty("file.separator") }
}

class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {
  StringLiteralFileSeparatorExpr() {
    this.getValue().matches("%/") or this.getValue().matches("%\\")
  }
}
```

```
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\" }
```

```
class StringLiteralFileSeparatorExpr extends FileSeparatorExpr, StringLiteral {
  StringLiteralFileSeparatorExpr() {
    this.getValue().matches("%/") or this.getValue().matches("%\\")
  }
}
```

```
{
  rand() instanceof FileSeparatorExpr
}

e instanceof FileSeparatorExpr
```

```
from MethodAccess ma
where
  ma.getMethod().instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType() instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType() instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }
```

```
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\" }
}
```

```
FileSeparatorExpr {
  getSystemProperty("file.separator") }

FileSeparatorExpr, StringLiteral {
  getValue().matches("%\\")
}
```

```
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\" }
}
```

```
class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }
}
```

```
predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and
  not isSafe(ma.getArgument(0)))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class FileSeparatorAppend extends AddExpr {  
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }  
}
```

```
class MethodStringStartsWith extends Method {  
  MethodStringStartsWith() {  
    this.getDeclaringType() instanceof TypeString and  
    this.hasName("startsWith")  
  }  
}  
  
class MethodFileGetCanonicalPath extends Method {  
  MethodFileGetCanonicalPath() {  
    this.getDeclaringType() instanceof TypeFile and  
    this.hasName("getCanonicalPath")  
  }  
}  
  
class MethodAccessFileGetCanonicalPath extends MethodAccess {  
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }  
}  
  
abstract class FileSeparatorExpr extends Expr { }
```

```
FileSeparatorExpr {  
  getSystemProperty("file.separator") }  
  
FileSeparatorExpr, StringLiteral {  
  getValue().matches("%\\")  
}  
}
```

```
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {  
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\ " }  
}
```

```
class FileSeparatorAppend extends AddExpr {  
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }  
}
```

```
predicate isSafe(Expr expr) {  
  DataFlow::localExprFlow(any(Expr e |  
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr  
  ), expr)  
}
```

```
from MethodAccess ma  
where  
  ma.getMethod() instanceof MethodStringStartsWith and  
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and  
  not isSafe(ma.getArgument(0))  
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
predicate isSafe(Expr expr) {  
  DataFlow::localExprFlow(any(Expr e |  
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr  
  ), expr)  
}
```

```
class MethodStringStartsWith extends Method {  
  MethodStringStartsWith() {  
    this.getDeclaringType() instanceof TypeString and  
    this.hasName("startsWith")  
  }  
}  
  
class MethodFileGetCanonicalPath extends Method {  
  MethodFileGetCanonicalPath() {  
    this.getDeclaringType() instanceof TypeFile and  
    this.hasName("getCanonicalPath")  
  }  
}  
  
class MethodAccessFileGetCanonicalPath extends MethodAccess {  
  MethodAccessFileGetCanonicalPath() { this.getMethod() instanceof MethodFileGetCanonicalPath }  
}  
  
abstract class FileSeparatorExpr extends Expr { }
```

```
FileSeparatorExpr {  
  getSystemProperty("file.separator") }  
  
FileSeparatorExpr, StringLiteral {  
  getValue().matches("%\\")  
}
```

```
class CharacterLiteralFileSeparatorExpr extends FileSeparatorExpr, CharacterLiteral {  
  CharacterLiteralFileSeparatorExpr() { this.getValue() = "/" or this.getValue() = "\\\" }  
}
```

```
class FileSeparatorAppend extends AddExpr {  
  FileSeparatorAppend() { this.getRightOperand() instanceof FileSeparatorExpr }  
}
```

```
predicate isSafe(Expr expr) {  
  DataFlow::localExprFlow(any(Expr e |  
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr  
  ), expr)  
}
```

```
from MethodAccess ma  
where  
  ma.getMethod() instanceof MethodStringStartsWith and  
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma, ma.getQualifier()) and  
  not isSafe(ma.getArgument(0))  
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

# CodeQL: Partial Path Traversal

```
class MethodStringStartsWith extends Method {
  MethodStringStartsWith() {
    this.getDeclaringType().instanceof TypeString and
    this.hasName("startsWith")
  }
}

class MethodFileGetCanonicalPath extends Method {
  MethodFileGetCanonicalPath() {
    this.getDeclaringType().instanceof TypeFile and
    this.hasName("getCanonicalPath")
  }
}

class MethodAccessFileGetCanonicalPath extends MethodAccess {
  MethodAccessFileGetCanonicalPath() { this.getMethod().instanceof MethodFileGetCanonicalPath }
}

abstract class FileSeparatorExpr extends Expr { }
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```

```
FileSeparatorExpr {
  getSystemProperty("file.separator") }

FileSeparatorExpr, StringLiteral {
  getValue().matches("%\\")
}

FileSeparatorExpr, CharacterLiteral {
  this.getValue() = "/" or this.getValue() = "\\ "
}

class FileSeparatorAppend extends AddExpr {
  FileSeparatorAppend() { this.getRightOperand().instanceof FileSeparatorExpr }
}

predicate isSafe(Expr expr) {
  DataFlow::localExprFlow(any(Expr e |
    e instanceof FileSeparatorAppend or e instanceof FileSeparatorExpr
  ), expr)
}
```

```
from MethodAccess ma
where
  ma.getMethod() instanceof MethodStringStartsWith and
  DataFlow::localExprFlow(any(MethodAccessFileGetCanonicalPath gcpma), ma.getQualifier()) and
  not isSafe(ma.getArgument(0))
select ma, "Partial Path Traversal Vulnerability due to insufficient guard against path traversal"
```