Local privilege escalation through system temporary directory

High **big-guy** published **GHSA-89qm-pxvm-p336** on Apr 9, 2021

Package

Gradle (Java)

Affected versions	Patched versions
<7.0	7.0

Description

Impact

On Unix-like systems, the system temporary directory can be created with open permissions that allow multiple users to create and delete files within it. Gradle builds could be vulnerable to a local privilege escalation from an attacker quickly deleting and recreating files in the system temporary directory.

This vulnerability impacted builds using precompiled script plugins written in Kotlin DSL and tests for Gradle plugins written using `ProjectBuilder` or TestKit.

If you are on Windows or modern versions of macOS, you are not vulnerable.
If you are on a Unix-like operating system with the "sticky" bit set on your system temporary directory, you are not vulnerable.

Precompiled script plugins

When compiling precompiled script plugins written in Kotlin DSL, Gradle uses a temporary directory in the system temporary directory to execute a build-like process.

A local privilege escalation can occur due to a race condition in the creation of this temporary directory. If an attacker is able to write files into the directory created by Gradle, the attacker can execute commands at the same privilege level as the build.

If you do not use precompiled script plugins, you are not vulnerable.

`ProjectBuilder` API in tests

`ProjectBuilder` is most commonly used by plugin authors to write unit tests. With this API, if you do not specify a project directory, Gradle will create one for you in a temporary directory.

A local privilege escalation can occur due to a race condition in the creation of this temporary directory. If an attacker is able to write files into the project directory created for the test, the attacker can execute commands at the same privilege level as the build.

If you do not use `ProjectBuilder` or you always specify a project directory, you are not vulnerable.

Gradle TestKit in tests

TestKit is also used by plugin authors to write integration tests. These tests execute full Gradle builds. TestKit uses a well-known path in the system temporary directory to cache some information between runs, like Gradle distributions and dependency downloads.

A local privilege escalation can occur if an attacker controls the creation of this directory. If an attacker is able to write files into the directory created for TestKit, the attacker can execute commands at the same privilege level as the build.

If you do not use TestKit, you are not vulnerable.

What should you do?

Upgrade to Gradle 7.0

As of Gradle 7.0, `ProjectBuilder`, TestKit and other uses of the system temporary directory have been removed where possible. Instead of relying on the system temporary directory, Gradle now uses a combination of the Gradle User Home and project's build directory.

Workaround for older versions

On Unix-like operating systems, ensure that the "sticky" bit is set. This only allows the original user (or root) to delete a file.

If you are unable to change the permissions of the system temporary directory, you can move the Java temporary directory by setting the System Property `java.io.tmpdir`. The new path needs to limit permissions to the build user only.

References

The vulnerable code:

- gradle/subprojects/core/src/main/java/org/gradle/api/internal/file/DefaultTemporaryFileProvider.java
  Lines 57 to 63 in `ff5dc12`

```
57      try {
58          // TODO: This is not a great paradigm for creating a temporary directory.
59          // See http://guava-libraries.googlecode.com/svn/tags/release08/javadoc/com/google/common/io/Files.html#createTempDir%28%29 for an alternative.
60          File tmpDir = File.createTempFile("gradle", "projectDir", dir);
61          tmpDir.delete();
62          tmpDir.mkdir();
63          return tmpDir;
```

- gradle/subprojects/test-kit/src/main/java/org/gradle/testkit/runner/internal/DefaultGradleRunner.java
  Lines 336 to 350 in `d024846`

```
336     private File createTestKitDir(TestKitDirProvider testKitDirProvider) {
337         File dir = testKitDirProvider.getDir();
338         if (dir.isDirectory()) {
339             if (!dir.canWrite()) {
340                 throw new InvalidRunnerConfigurationException("Unable to write to test kit directory: " + dir.getAbsolutePath());
341             }
342             return dir;
```

```
343         } else if (dir.exists() && !dir.isDirectory()) {
344             throw new InvalidRunnerConfigurationException("Unable to use non-directory as test kit directory: " + dir.getAbsolutePath());
345         } else if (dir.mkdirs() || dir.isDirectory()) {
346             return dir;
```

- #15240
- #15654

These other projects experienced similar vulnerabilities:

- Eclipse Jetty - GHSA-g3wg-6mcf-8jj6
- JUnit 4 - GHSA-269g-pwp5-87pp
- Google Guava - google/guava#4011
- Apache Ant - https://nvd.nist.gov/vuln/detail/CVE-2020-1945
- JetBrains Kotlin Compiler - https://nvd.nist.gov/vuln/detail/CVE-2020-15824

More about this type of vulnerability:

- CWE-378: Creation of Temporary File With Insecure Permissions
- CWE-379: Creation of Temporary File in Directory with Insecure Permissions

Questions?

- For security related issues, please email us at security@gradle.com.
- For non-security related issues, please open an issue on GitHub.

Severity

High 8.8 / 10

CVSS base metrics

Attack vector	Local
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

CVE ID

CVE-2021-29428

Weaknesses

CWE-378 CWE-379

Credits

JLLeitschuh

big-guy