


TemporaryFolder on unix-like systems does not limit access to created files

Low marcphilipp published GHSA-269g-pwp5-87pp on Oct 11, 2020

Package

 **junit:junit** (Maven)

Affected versions

4.7 - 4.13

Patched versions

4.13.1

Description

Vulnerability

The JUnit4 test rule [TemporaryFolder](#) contains a local information disclosure vulnerability.

Example of vulnerable code:

```
public static class HasTempFolder {
    @Rule
    public TemporaryFolder folder = new TemporaryFolder();

    @Test
    public void testUsingTempFolder() throws IOException {
        folder.getRoot(); // Previous file permissions: 'drwxr-xr-x'; After fix: 'drwx-----'
        File createdFile= folder.newFile("myfile.txt"); // unchanged/irrelevant file permissions
        File createdFolder= folder.newFolder("subfolder"); // unchanged/irrelevant file permissions
        // ...
    }
}
```

Impact

On Unix like systems, the system's temporary directory is shared between all users on that system. Because of this, when files and directories are written into this directory they are, by default, readable by other users on that same system.

This vulnerability **does not** allow other users to overwrite the contents of these directories or files. This is purely an information disclosure vulnerability.

When analyzing the impact of this vulnerability, here are the important questions to ask:

- Do the JUnit tests write sensitive information, like API keys or passwords, into the temporary folder?
 - If yes, this vulnerability impacts you, but only if you also answer 'yes' to question 2.
 - If no, this vulnerability does not impact you.
- Do the JUnit tests ever execute in an environment where the OS has other untrusted users.

This may apply in CI/CD environments but normally won't be 'yes' for personal developer machines.

 - If yes, and you answered 'yes' to question 1, this vulnerability impacts you.
 - If no, this vulnerability does not impact you.

Patches

Because certain JDK file system APIs were only added in JDK 1.7, this this fix is dependent upon the version of the JDK you are using.

- Java 1.7 and higher users: this vulnerability is fixed in 4.13.1.
- Java 1.6 and lower users: **no patch is available, you must use the workaround below.**

Workarounds

If you are unable to patch, or are stuck running on Java 1.6, specifying the `java.io.tmpdir` system environment variable to a directory that is exclusively owned by the executing user will fix this vulnerability.

References

- [CWE-200: Exposure of Sensitive Information to an Unauthorized Actor](#)
- Fix commit [610155b](#)

Similar Vulnerabilities

- 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>

For more information

If you have any questions or comments about this advisory, please pen an issue in [junit-team/junit4](#).

Severity

Low

CVE ID

CVE-2020-15250

Weaknesses

No CWEs

Credits

 JLeitschuh