Generator Web Application: Local Privilege Escalation Vulnerability via System Temp Directory

Critical **frantuma** published **GHSA-pc22-3g76-gm6j** on Mar 9, 2021

Package

swagger-codegen (Maven)

Affected versions	Patched versions
< 2.4.19	2.4.19+

Description

Impact

On Unix like systems, the system's temporary directory is shared between all users on that system. A collocated user can observe the process of creating a temporary sub directory in the shared temporary directory and race to complete the creation of the temporary subdirectory.

This vulnerability is local privilege escalation because the contents of the `outputFolder` can be appended to by an attacker. As such, code written to this directory, when executed can be attacker controlled.

Java Code

The method `File.createTempFile` from the JDK is vulnerable to this local information disclosure vulnerability.

swagger-codegen/modules/swagger-generator/src/main/java/io/swagger/generator/online/Generator.java
Lines 174 to 185 in 068b1eb

```
174     protected static File getTmpFolder() {
175         try {
176             File outputFolder = File.createTempFile("codegen-", "-tmp");
177             outputFolder.delete();
178             outputFolder.mkdir();
179             outputFolder.deleteOnExit();
180             return outputFolder;
181         } catch (Exception e) {
182             e.printStackTrace();
183             return null;
184         }
185     }
```

Patches

Fix has been applied to the master branch with:

- 987ea7a

included in release: 2.4.19

References

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

For more information

If you have any questions or comments about this advisory:

- Email us at security@swagger.io

Original vulnerability report

> I'm performing OSS security research under the GitHub Security Lab Bug Bounty program.
> I've been using a custom CodeQL query to find local temporary directory vulnerabilities in OSS with three custom CodeQL queries.
>
> - https://github.com/github/codeql/pull/4388/files#diff-71d36c0f2bd0b08e32866f873f1c906cdc17277e0ad327c0c6cd2c882f30de4f
> - https://github.com/github/codeql/pull/4388/files#diff-1893a18a8bf43c011d61a7889d0139b998a5a78701a30fe7722eddd4c506aaac
> - github/codeql#4473
>
> The code generated by the Swagger Generator contains a local information disclosure vulnerability. The system temporary directory, on unix-like systems is shared between multiple users. Information written to this directory, or directories created under this directory that do not correctly set the posix standard permissions can have these directories read/modified by other users.
>
> This vulnerability exists in the maven plugin.
>
> This vulnerability is distinctly different. This vulnerability is most likely a local privilege escalation vulnerability.

swagger-codegen/modules/swagger-generator/src/main/java/io/swagger/generator/online/Generator.java
Lines 174 to 185 in 068b1eb

```
174     protected static File getTmpFolder() {
175         try {
176             File outputFolder = File.createTempFile("codegen-", "-tmp");
177             outputFolder.delete();
178             outputFolder.mkdir();
179             outputFolder.deleteOnExit();
```

```
            return outputFolder;
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
```

This vulnerability is very similar to this similar vulnerability I disclosed in the Eclipse Jetty project.

GHSA-g3wg-6mcf-8jj6

This is due to a race condition between the call to `delete` and the call to `mkdirs`.

```
// ensure file will always be unique by appending random digits
File outputFolder = File.createTempFile("codegen-", "-tmp"); // Attacker knows the full path of the file that will be generated
// delete the file that was created
outputFolder.delete(); // Attacker sees file is deleted and begins a race to create their own directory before Swagger Code Generator.
// and make a directory of the same name
// SECURITY VULNERABILITY: Race Condition! - Attacker beats Swagger Code Generator and now owns this directory
outputFolder.mkdirs();
```

This vulnerability is local privilege escalation because the contents of the `outputFolder` can be appended to by an attacker. As such, code written to this directory, when executed can be attacker controlled.

The fix here is to switch to the `Files` API for creating temporary directories. Which does not contain this race condition, and appropriately sets the correct file permissions.

Severity

Critical 9.3 / 10

CVSS base metrics

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

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

CVE ID

CVE-2021-21363

Weaknesses

CWE-378 CWE-379

Credits

JLLeitschuh