rules.sonarsource.com

Objective C static code analysis: Setting loose POSIX file permissions is security-sensitive

3 minutes

In Unix, "`others`" class refers to all users except the owner of the file and the members of the group assigned to this file.

Granting permissions to this group can lead to unintended access to files.

Ask Yourself Whether

- The application is designed to be run on a multi-user environment.
- Corresponding files and directories may contain confidential information.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

The most restrictive possible permissions should be assigned to files and directories.

Sensitive Code Example

When creating a file or directory with permissions to "other group":

```
open("myfile.txt", O_CREAT, S_IRWXU | S_IRWXG | S_IRWXO); // Sensitive: the process set 777 permissions to this newly created file

mkdir("myfolder", S_IRWXU | S_IRWXG | S_IRWXO); // Sensitive: the process try to set 777 permissions to this newly created directory
```

When explicitly adding permissions to "other group" with `chmod`, `fchmod` or `filesystem::permissions` functions:

```
chmod("myfile.txt", S_IRWXU | S_IRWXG | S_IRWXO);  // Sensitive: the process set 777 permissions to this file

fchmod(fd, S_IRWXU | S_IRWXG | S_IRWXO); // Sensitive: the process set 777 permissions to this file descriptor
```

When defining the `umask` without read, write and execute permissions for "other group":

```
umask(S_IRWXU | S_IRWXG); // Sensitive: the further files and folders will be created with possible permissions to "other group"
```

Compliant Solution

When creating a file or directory, do not set permissions to "other group":

```
open("myfile.txt", O_CREAT, S_IRWXU | S_IRWXG); // Compliant

mkdir("myfolder", S_IRWXU | S_IRWXG); // Compliant
```

When using `chmod`, `fchmod` or `filesystem::permissions` functions, do not add permissions to "other group":

```
chmod("myfile.txt", S_IRWXU | S_IRWXG);  // Compliant

fchmod(fd, S_IRWXU | S_IRWXG); // Compliant
```

When defining the `umask`, set read, write and execute permissions to other group:

```
umask(S_IRWXO); // Compliant: further created files or directories will not have permissions set for "other group"
```

See

- OWASP Top 10 2021 Category A1 - Broken Access Control
- OWASP Top 10 2021 Category A4 - Insecure Design
- OWASP Top 10 2017 Category A5 - Broken Access Control

- OWASP File Permission
- MITRE, CWE-732 - Incorrect Permission Assignment for Critical Resource
- MITRE, CWE-266 - Incorrect Privilege Assignment
- CERT, FIO01-J. - Create files with appropriate access permissions
- CERT, FIO06-C. - Create files with appropriate access permissions
- SANS Top 25 - Porous Defenses