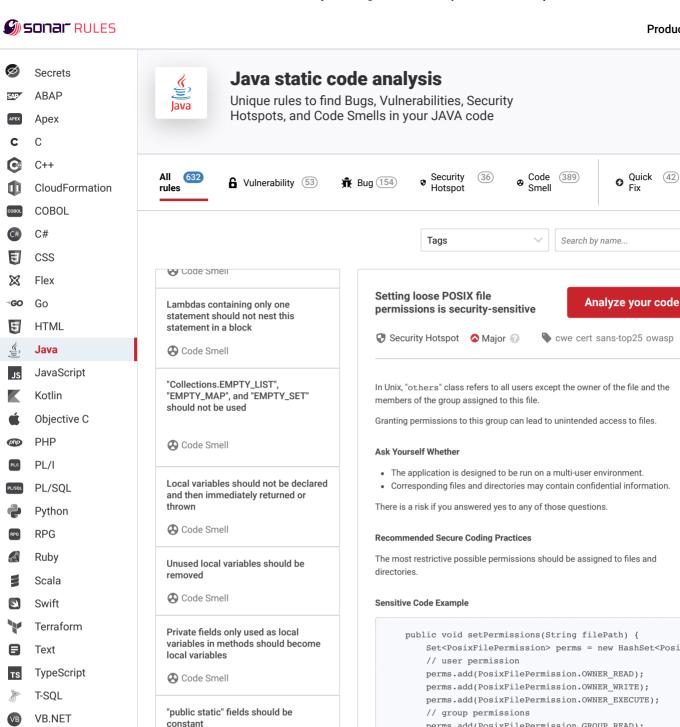
Products ✓

O Quick 42 Fix



public void setPermissions(String filePath) { Set<PosixFilePermission> perms = new HashSet<PosixFi perms.add(PosixFilePermission.OWNER READ); perms.add(PosixFilePermission.OWNER WRITE); perms.add(PosixFilePermission.OWNER EXECUTE); perms.add(PosixFilePermission.GROUP READ); perms.add(PosixFilePermission.GROUP_EXECUTE); // others permissions perms.add(PosixFilePermission.OTHERS_READ); // Sensi perms.add(PosixFilePermission.OTHERS WRITE); // Sens perms.add(PosixFilePermission.OTHERS_EXECUTE); // Se Files.setPosixFilePermissions(Paths.get(filePath), p

public void setPermissionsUsingRuntimeExec(String filePa Runtime.getRuntime().exec("chmod 777 file.json"); //

public void setOthersPermissionsHardCoded(String filePat Files.setPosixFilePermissions(Paths.get(filePath), P

Compliant Solution

On operating systems that implement POSIX standard. This will throw a UnsupportedOperationException on Windows

VB₆

XML

Code Smell

Code Smell

Code Smell

Code Smell

of a "for" loop

Code Smell

least 3 "case" clauses

Loops should not contain more than a single "break" or "continue" statement

Declarations should use Java collection interfaces such as "List" rather than specific implementation

classes such as "LinkedList"

"switch" statements should have at

A "while" loop should be used instead

The default unnamed package should not be used

Code Smell

"equals(Object obj)" should be overridden along with the "compareTo(T obj)" method

A Code Smell

Package names should comply with a naming convention

Code Smell

Nested code blocks should not be used

```
public void setPermissionsSafe(String filePath) throws I
    Set<PosixFilePermission> perms = new HashSet<PosixFi
    // user permission
    perms.add(PosixFilePermission.OWNER_READ);
    perms.add(PosixFilePermission.OWNER_WRITE);
    perms.add(PosixFilePermission.OWNER_EXECUTE);
    // group permissions
    perms.add(PosixFilePermission.GROUP_READ);
    perms.add(PosixFilePermission.GROUP_EXECUTE);
    // others permissions removed
    perms.remove(PosixFilePermission.OTHERS_READ); // Co
    perms.remove(PosixFilePermission.OTHERS_WRITE); // C
    perms.remove(PosixFilePermission.OTHERS_EXECUTE); //
    Files.setPosixFilePermissions(Paths.get(filePath), p
}</pre>
```

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, FI001-J. Create files with appropriate access permissions
- CERT, FI006-C. Create files with appropriate access permissions
- SANS Top 25 Porous Defenses

Available In:

sonarcloud <equation-block> 🛮 sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy