Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

All rules 632 | Vulnerability 53 | Bug 154 | Security Hotspot 36 | Code Smell 389 | Quick Fix 42

Tags | Search by name...

Catches should be combined
Code Smell

Methods of "Random" that return floating point values should not be used in random integer generation
Code Smell

Parsing should be used to convert "Strings" to primitives
Code Smell

Classes should not be empty
Code Smell

Fields in non-serializable classes should not be "transient"
Code Smell

Boolean checks should not be inverted
Code Smell

Redundant casts should not be used
Code Smell

"@Deprecated" code should not be used
Code Smell

"toString()" should never be called on a String object
Code Smell

Annotation repetitions should not be wrapped
Code Smell

Multiple variables should not be declared on the same line
Code Smell

Strings should not be concatenated using '+' in a loop

Using unencrypted files in mobile applications is security-sensitive

Analyze your code

Security Hotspot Major cwe owasp android

Storing files locally is a common task for mobile applications. Files that are stored unencrypted can be read out and modified by an attacker with physical access to the device. Access to sensitive data can be harmful for the user of the application, for example when the device gets stolen.

Ask Yourself Whether

- The file contains sensitive data that could cause harm when leaked.

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

Recommended Secure Coding Practices

It's recommended to password-encrypt local files that contain sensitive information. The class EncryptedFile can be used to easily encrypt files.

Sensitive Code Example

```
Files.write(path, content); // Sensitive

FileOutputStream out = new FileOutputStream(file); // Sensit

FileWriter fw = new FileWriter("outfilename", false); // Sen
```

Compliant Solution

```
String masterKeyAlias = MasterKeys.getOrCreate(MasterKeys.AE

File file = new File(context.getFilesDir(), "secret_data");
EncryptedFile encryptedFile = EncryptedFile.Builder(
    file,
    context,
    masterKeyAlias,
    EncryptedFile.FileEncryptionScheme.AES256_GCM_HKDF_4KB
).build();

// write to the encrypted file
FileOutputStream encryptedOutputStream = encryptedFile.openF
```

See

- OWASP Top 10 2021 Category A4 - Insecure Design
- Mobile AppSec Verification Standard - Data Storage and Privacy Requirements
- OWASP Mobile Top 10 2016 Category M2 - Insecure Data Storage
- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-311 - Missing Encryption of Sensitive Data

Code Smell

Maps with keys that are enum values should be replaced with EnumMap

Code Smell

Lambdas should be replaced with method references

Code Smell

Parentheses should be removed from a single lambda input parameter when its type is inferred

Code Smell

Abstract classes without fields should be converted to interfaces

Available In:
sonarcloud | sonarqube