Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
HTML
Java
JavaScript
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

Kotlin static code analysis

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

All rules 98 | Vulnerability 10 | Bug 17 | Security Hotspot 15 | Code Smell 56

Tags

Search by name...

Hard-coded credentials are security-sensitive
Security Hotspot

Cipher algorithms should be robust
Vulnerability

Encryption algorithms should be used with secure mode and padding scheme
Vulnerability

Server hostnames should be verified during SSL/TLS connections
Vulnerability

Server certificates should be verified during SSL/TLS connections
Vulnerability

Cryptographic keys should be robust
Vulnerability

Weak SSL/TLS protocols should not be used
Vulnerability

"SecureRandom" seeds should not be predictable
Vulnerability

Cipher Block Chaining IVs should be unpredictable
Vulnerability

Hashes should include an unpredictable salt
Vulnerability

Regular expressions should be syntactically valid
Bug

"runFinalizersOnExit" should not be called
Bug

Broadcasting intents is security-sensitive

Analyze your code

Security Hotspot Critical cwe android owasp

In Android applications, broadcasting intents is security-sensitive. For example, it has led in the past to the following vulnerability:

- CVE-2018-9489

By default, broadcasted intents are visible to every application, exposing all sensitive information they contain.

This rule raises an issue when an intent is broadcasted without specifying any "receiver permission".

Ask Yourself Whether

- The intent contains sensitive information.
- Intent reception is not restricted.

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

Recommended Secure Coding Practices

Restrict the access to broadcasted intents. See Android documentation for more information.

Sensitive Code Example

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.UserHandle

public class MyIntentBroadcast {
    fun broadcast(intent: Intent,
                  context: Context,
                  user: UserHandle,
                  resultReceiver: BroadcastReceiver,
                  scheduler: Handler,
                  initialCode: Int,
                  initialData: String,
                  initialExtras: Bundle,
                  broadcastPermission: String) {
        context.sendBroadcast(intent) // Sensitive
        context.sendBroadcastAsUser(intent, user) // Sen

        // Broadcasting intent with "null" for receiverP
        context.sendBroadcast(intent, null) // Sensitive
        context.sendBroadcastAsUser(intent, user, null)
        context.sendOrderedBroadcast(intent, null) // Se
        context.sendOrderedBroadcastAsUser(intent, user,
            scheduler, initialCode, initialData, initial
    }
}
```

Compliant Solution

"ScheduledThreadPoolExecutor" should not have 0 core threads

Bug

Jump statements should not occur in "finally" blocks

Bug

Using clear-text protocols is security-sensitive

Security Hotspot

Accessing Android external storage is security-sensitive

Security Hotspot

Receiving intents is security-sensitive

Security Hotspot

Broadcasting intents is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot

Using pseudorandom number generators (PRNGs) is security-sensitive

Security Hotspot

Empty lines should not be tested with regex MULTILINE flag

Code Smell

Cognitive Complexity of functions should not be too high

Code Smell

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.UserHandle

public class MyIntentBroadcast {
    fun broadcast(intent: Intent,
                  context: Context,
                  user: UserHandle,
                  resultReceiver: BroadcastReceiver,
                  scheduler: Handler,
                  initialCode: Int,
                  initialData: String,
                  initialExtras: Bundle,
                  broadcastPermission: String) {

        context.sendBroadcast(intent, broadcastPermissio
        context.sendBroadcastAsUser(intent, user, broadc
        context.sendOrderedBroadcast(intent, broadcastPe
        context.sendOrderedBroadcastAsUser(intent, user,
            scheduler, initialCode, initialData, initial
    }
}
```

See

- OWASP Top 10 2021 Category A4 - Insecure Design
- Mobile AppSec Verification Standard - Platform Interaction Requirements
- OWASP Mobile Top 10 2016 Category M1 - Improper Platform Usage
- MITRE, CWE-927 - Use of Implicit Intent for Sensitive Communication
- Android documentation - Broadcast Overview - Security considerations and best practices

Available In:

sonarcloud | sonarqube