

Leveraging AppSweep helps you identify and fix security issues in your code and dependencies with actionable recommendations and insights that will help you build more secure mobile apps.

# 7 MASTG Kotlin App [🔗](#)

owasp.mastgkotlin · Analysed Apr 24 2023, 19:05

Version

1.0

App size

10.58 MB

Commit Hash

9e9eb9053834eb9eb79a03a72c201ab74306be2b

App composition

Bytecode size: 10.00 MB

Number of Java classes: 4473

Number of Kotlin classes: 2106

Analysis duration

1m 10s

Obfuscation mapping

✗ Not Provided

Tags

Debug

Issue Summary

3

High Severity Issues

2 internal · 1 in dependencies

6

Medium Severity Issues

4 internal · 2 in dependencies

4

Low Severity Issues

3 internal · 4 in dependencies

46

Dependencies

38 transitive dependencies



High • **Android manifest attribute android:debuggable="true" is set** [↗](#)

The attribute `android:debuggable` is set to `true` in the app's manifest. This means that your app can be debugged using Java Wired Debugging Protocol (JWDP). Using JWDP, it is possible to gain full access to the Java process and execute arbitrary code in the context of a debuggable app.

Releasing an app with this flag set can lead to leakage of sensitive information and leaves the app vulnerable to debugging.

Note that setting `android:debuggable` to `false` is necessary to prevent debugging, but is not sufficient. An adversary can still connect a debugger and use it to reverse-engineer or tamper with the app's behaviour.

Recommendations

Ensure that the flag `android:debuggable` is set to `false` in your `AndroidManifest.xml` when building for release.



Fix with DexGuard

Setting the attribute `android:debuggable` to `"false"` is necessary to prevent debugging, but is not sufficient. An adversary can still connect a debugger and use it to reverse-engineer or tamper with the app's behaviour.

Enable debugging protection in DexGuard using this configuration line:

`-raspchecks debug`

or consult the RASP section in your Dexguard Manual to learn more.

External Links

[OWASP recommendations regarding debuggable flags](#)

1 Finding

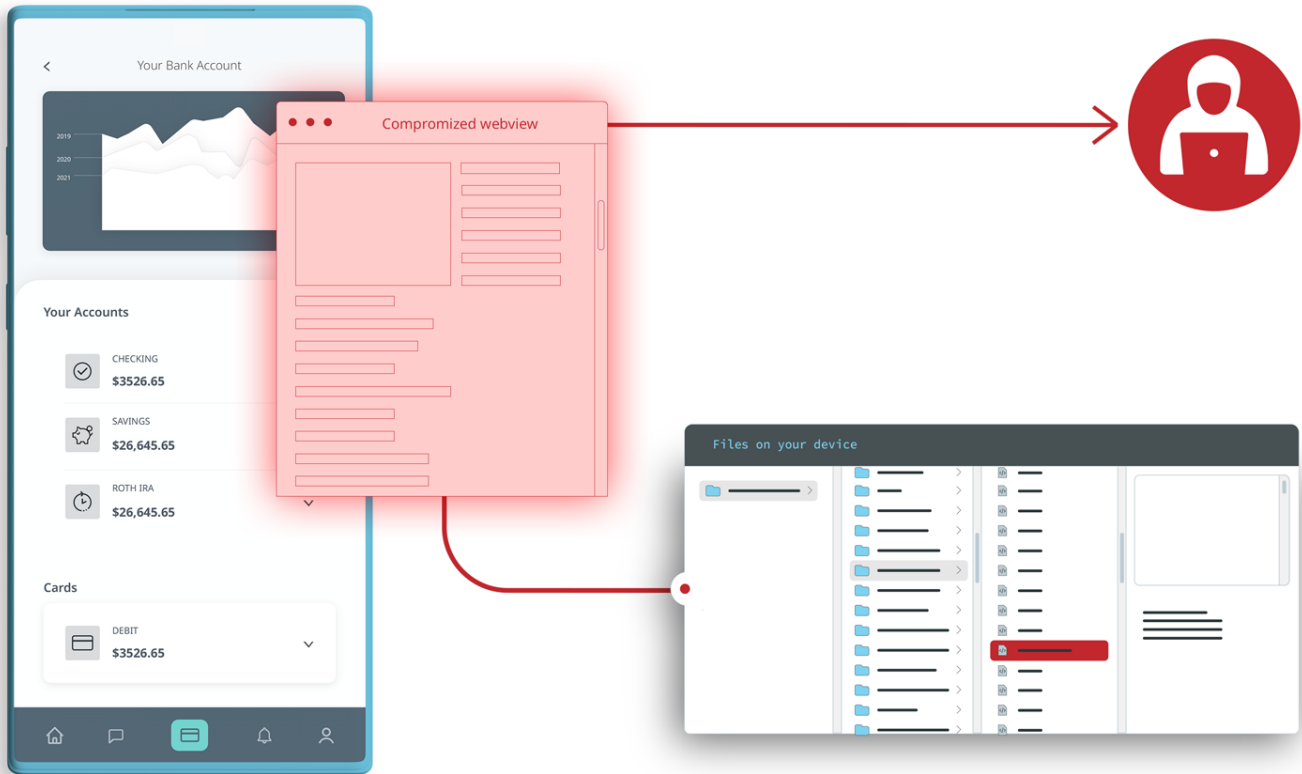
- `AndroidManifest.xml`



High ● The app enables dangerous file access via `setAllowFileAccessFromFileURLs`

Your app uses the API `setAllowFileAccessFromFileURLs`, to enable dangerous file access. This method is deprecated and is not considered secure.

If local file access is enabled in a WebView, an attacker who gains access to that WebView, for example through a man-in-the-middle (MitM) attack, can gain access to user's files on the device through the vulnerable app.



Recommendations

Use `androidx#webkit#WebViewAssetLoader` to load file content securely.

External Links

[WebSettings](#) | [Android Developers](#)

1 Finding

- `setAllowFileAccessFromFileURLs`
`owasp.mastgkotlin.InsecureWebViewActivity:24`



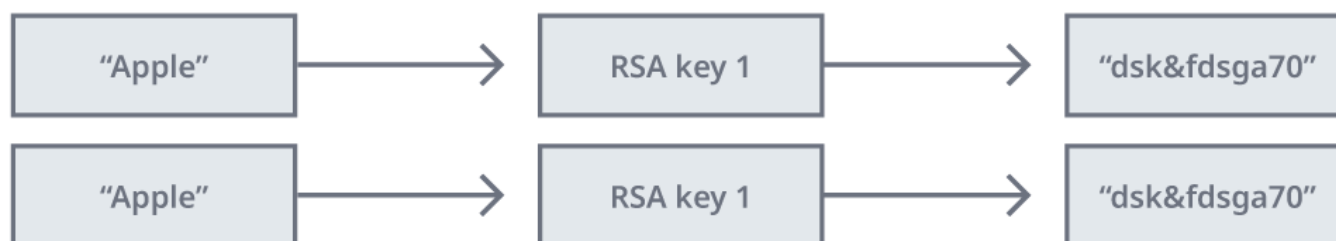
High • Asymmetric cipher with insecure padding used [↗](#)

Asymmetric encryption algorithms should be used with OAEP padding. Secure paddings must be used to protect against a number of attacks against RSA.

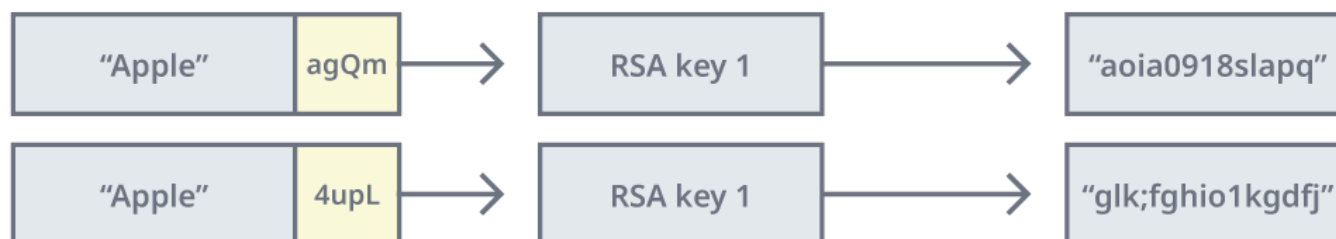
The simplest implementation of RSA is deterministic. This means that for the same message and the same encryption key the result will always be the same. In most cases it is important to not reveal the fact that an identical message is being sent. In case when the same message is sent to multiple recipients encrypted with several public keys, an additional attack can be performed that may retrieve the message without having any of the recipients' keys.

Padding is a way to diversify encryption by adding extra random information. With padding, the above issues do not apply and the encryption is safe to use.

Without padding, the same data generates the same ciphertext for the same keypair.



With padding, even the same data always generates different ciphertext.



Using the RSA algorithm without Optimal Asymmetric Encryption Padding (OAEP) might weaken the encryption.

Recommendations

Use OAEP padding to secure your asymmetric cipher.

External Links

[CWE-780: Use of RSA Algorithm without OAEP](#)

[Attacks against plain RSA without padding](#)

[M5: Insufficient CryptographyTestValidDescription](#)

[OWASP recommendations regarding RSA algorithms not supporting padding](#)

2 Findings

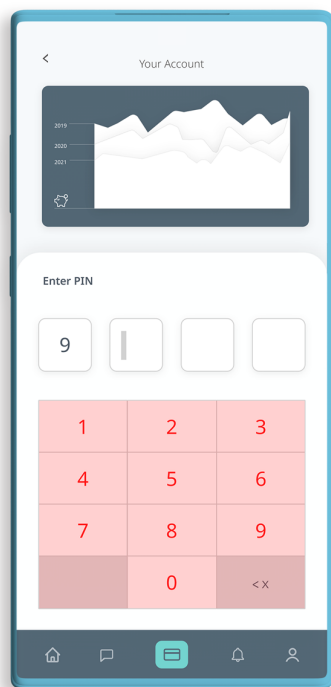
- [de.adorsys.android.securestoragelibrary.a](#)
([de.adorsys.android:securestoragelibrary:0.0.2](#))
- [de.adorsys.android.securestoragelibrary.a](#)
([de.adorsys.android:securestoragelibrary:0.0.2](#))



Medium ● Elements of RegisterActivity are not protected against tapjacking [↗](#)

Tapjacking is a technique that allows an attacker to capture the taps in your app (for example, on a virtual pin-pad), or trick users into making taps without their consent (for example, switching off an important security setting).

Tapjacking protection is especially important for security relevant parts of the app like pin or password entry.



The essence of the attack is that a malicious app places a window over your app.

If the attacker wants to capture user clicks, that window will be transparent. The overlay window gets an opportunity to learn about the taps made in your app without the device user being aware.

If the attacker wants to trick the user into clicking something in your app, the window will be opaque with fake controls lying exactly over the corresponding controls in your app.

For instance, as seen in the image to the left, placing transparent overlays over each button on a pin pad allows an attacker to capture users' pin codes.

Recommendations

Add `filterTouchesWhenObscured="true"` to the relevant view elements in the respective layout files, or set the protection programmatically.

External Links

[Android Developer - View Security](#)

[OWASP recommendations regarding tapjacking](#)

1 Finding

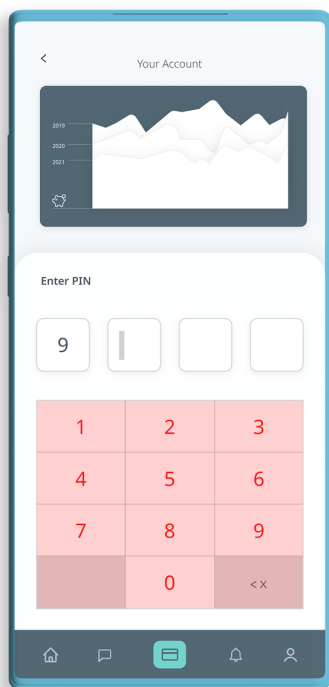
- **Button**
res/layout/activity_main.xml



Medium ● Elements of MainActivity are not protected against tapjacking [🔗](#)

Tapjacking is a technique that allows an attacker to capture the taps in your app (for example, on a virtual pin-pad), or trick users into making taps without their consent (for example, switching off an important security setting).

Tapjacking protection is especially important for security relevant parts of the app like pin or password entry.



The essence of the attack is that a malicious app places a window over your app.

If the attacker wants to capture user clicks, that window will be transparent. The overlay window gets an opportunity to learn about the taps made in your app without the device user being aware.

If the attacker wants to trick the user into clicking something in your app, the window will be opaque with fake controls lying exactly over the corresponding controls in your app.

For instance, as seen in the image to the left, placing transparent overlays over each button on a pin pad allows an attacker to capture users' pin codes.

Recommendations

Add `filterTouchesWhenObscured="true"` to the relevant view elements in the respective layout files, or set the protection programmatically.

External Links

[Android Developer - View Security](#)

[OWASP recommendations regarding tapjacking](#)

1 Finding

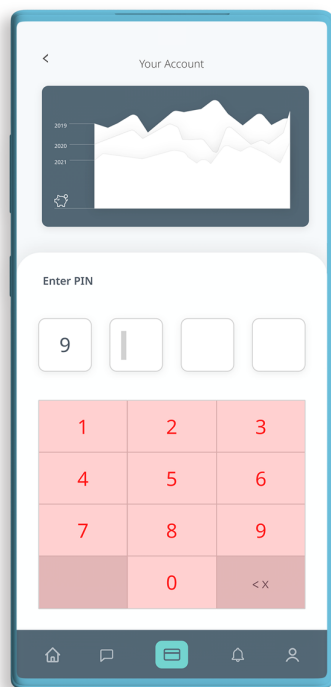
- **Button**
res/layout/activity_main.xml



Medium ● Elements of MenuActivity are not protected against tapjacking [🔗](#)

Tapjacking is a technique that allows an attacker to capture the taps in your app (for example, on a virtual pin-pad), or trick users into making taps without their consent (for example, switching off an important security setting).

Tapjacking protection is especially important for security relevant parts of the app like pin or password entry.



The essence of the attack is that a malicious app places a window over your app.

If the attacker wants to capture user clicks, that window will be transparent. The overlay window gets an opportunity to learn about the taps made in your app without the device user being aware.

If the attacker wants to trick the user into clicking something in your app, the window will be opaque with fake controls lying exactly over the corresponding controls in your app.

For instance, as seen in the image to the left, placing transparent overlays over each button on a pin pad allows an attacker to capture users' pin codes.

Recommendations

Add `filterTouchesWhenObscured="true"` to the relevant view elements in the respective layout files, or set the protection programmatically.

External Links

[Android Developer - View Security](#)

[OWASP recommendations regarding tapjacking](#)

1 Finding

- **Button**
res/layout/activity_menu.xml

Medium ● **Outdated protocol SSL enabled** [↗](#)

Initializing SSLContext with a generic or old TLS version may enable outdated and insecure communication protocols on devices with API level less than 26.

Recommendations

Explicitly initialize SSLContext with TLSv1.2 or TLSv1.3. Additionally, make sure your server configuration only allows TLSv1.2 or newer versions.

External Links

[OWASP recommendations regarding outdated TLS certificates](#)

1 Finding

- `com.github.kittinunf.fuel.core.FuelManager$socketFactory$2:33`
([com.github.kittinunf.fuel:fuel:1.12.1](#))

Medium ● **The app allows cleartext communication** [↗](#)

Cleartext communication should be disabled as it allows attackers to spy on your network traffic. Android allows developers to configure their cleartext communication through the manifest or by importing a network security configuration file. Beware, if you're adding a network configuration file, it will always override the values in the manifest. Even if the flag is not set in the network security file the default value of that flag will override the values set in the manifest. Additionally, below Android 28 cleartext is enabled by default and starting from Android 28 the flag is disabled by default.

Recommendations

Ensure that the flag `cleartextTrafficPermitted` is set to `false` in your `base-config` tag of the security network file. If the app doesn't contain a network security configuration, ensure `android:usesCleartextTraffic` is set to `false` in the `application` tag of the manifest.

External Links

[MASTG: Verifying Data Encryption on the Network](#)

[Android Documentation - security config](#)

1 Finding

- `AndroidManifest.xml`

**Low ● The app logs information** [↗](#)

Logs may give important information to an attacker, in particular, once sensitive data is logged. But even the log messages in the code itself can give a reverse engineer a lot of information what is happening, and can make reverse engineering much easier.


Keeping logging in your app also increases the app size, often unnecessarily.

Recommendations

Remove all logging statements before releasing the app. Using tools, this can be done in an automated way.

**Fix with ProGuard**

The ProGuard configuration can be modified to remove logging by adding

```
-assumenosideeffects class 
    android.util.Log {
    public static int v(...);
    public static int i(...);
    public static int w(...);
    public static int d(...);
    public static int e(...);
}
```

More detailed information can be found in the ProGuard Community.

External Links

[OWASP recommendations regarding using loggers](#)

36 Findings

- 1 occurrences in [androidx.customview.widget](#)
- 13 occurrences in [androidx.core.widget](#)
- 4 occurrences in [de.adorsys.android.securestoragelibrary](#)
- 3 occurrences in [androidx.appcompat.view](#)
- 3 occurrences in [androidx.startup](#)
- 9 occurrences in [androidx.core.util](#)

- 1 occurrences in
`androidx.core.content.pm`
- 2 occurrences in
`androidx.activity.result`
- 7 occurrences in
`androidx.constraintlayout.helper.widget`
- 32 occurrences in
`androidx.appcompat.app`
- 53 occurrences in
`androidx.constraintlayout.widget`
- 2 occurrences in
`owasp.mastgkotlin`
- 1 occurrences in
`androidx.core.view.accessibility`
- 9 occurrences in
`androidx.vectordrawable.graphics.drawable`
- 8 occurrences in
`androidx.core.os`
- 2 occurrences in
`androidx.viewpager.widget`
- 8 occurrences in
`androidx.constraintlayout.motion.utils`
- 8 occurrences in
`androidx.core.content.res`
- 32 occurrences in
`androidx.core.view`
- 15 occurrences in
`org.jetbrains.anko`
- 101 occurrences in
`androidx.fragment.app`

- 58 occurrences in
`androidx.constraintlayout.motion.widget`
- 12 occurrences in
`androidx.loader.app`
- 1 occurrences in
`androidx.tracing`
- 2 occurrences in
`androidx.appcompat.graphics.drawable`
- 1 occurrences in
`androidx.core.view.inputmethod`
- 27 occurrences in
`androidx.core.graphics.drawable`
- 30 occurrences in
`androidx.core.app`
- 2 occurrences in
`androidx.constraintlayout.utils.widget`
- 2 occurrences in
`androidx.appcompat.view.menu`
- 2 occurrences in
`androidx.concurrent.futures`
- 58 occurrences in
`androidx.appcompat.widget`
- 4 occurrences in
`androidx.core.content`
- 1 occurrences in
`androidx.loader.content`
- 7 occurrences in
`androidx.core.text`
- 13 occurrences in
`androidx.core.graphics`



Low ● **Legacy cryptographic classes used** [🔗](#)

Several classes in the `java.security` package have been marked as legacy and must no longer be used. Please refer to the Android SDK documentation for information on how to securely use cryptography in your app.

Recommendations

Only use supported cryptography classes.

External Links

Check the Android SDK documentation to find out which cryptography classes are no longer supported
[OWASP recommendations regarding legacy cryptography class usage](#)

2 Findings

Legacy class `java.security.AccessController` used

- 1 occurrences in
[com.google.gson.internal.bind](#)

Legacy class `java.security.PrivilegedAction` used

- 1 occurrences in
[com.google.gson.internal.bind](#)



Low ● **Classes contain Kotlin Metadata** [↗](#)

Kotlin Metadata may give important information to a reverse engineer, like class names.

Recommendations

Kotlin Metadata should be removed from the app to minimize its size and make reverse-engineering more difficult. This can be done automatically with Guardsquare's ProGuard or DexGuard.



Fix with ProGuard

By default, ProGuard will remove all Kotlin Metadata. If you do not have a specific reason to keep parts of this Metadata, you can use the ProGuard default configuration.

External Links

[Kotlin Metadata Printer](#) | [Guardsquare](#)

98 Findings

- | | |
|---|---|
| <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 6 occurrences in org.jetbrains.anko.custom | <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 91 occurrences in org.jetbrains.anko |
| <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 16 occurrences in kotlin.coroutines.jvm.internal | <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 13 occurrences in androidx.savedstate |
| <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 16 occurrences in androidx.core.graphics | <p>Kotlin Metadata class annotation found</p> <ul style="list-style-type: none">• 7 occurrences in com.github.kittinunf.fuel |

Kotlin Metadata class annotation found

- 37 occurrences in [kotlin.ranges](#)

Kotlin Metadata class annotation found

- 2 occurrences in [com.github.kittinunf.fuel.toolbox](#)

Kotlin Metadata class annotation found

- 7 occurrences in [com.github.kittinunf.fuel.util](#)

Kotlin Metadata class annotation found

- 32 occurrences in [androidx.core.util](#)

Kotlin Metadata class annotation found

- 5 occurrences in [kotlin.math](#)

Kotlin Metadata class annotation found

- 47 occurrences in [kotlinx.coroutines.flow.internal](#)

Kotlin Metadata class annotation found

- 22 occurrences in [kotlinx.coroutines.selects](#)

Kotlin Metadata class annotation found

- 2 occurrences in [kotlin.internal.jdk7](#)

Kotlin Metadata class annotation found

- 27 occurrences in [kotlinx.coroutines.debug.internal](#)

Kotlin Metadata class annotation found

- 15 occurrences in [kotlin.random](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.text.jdk8](#)

Kotlin Metadata class annotation found

- 23 occurrences in [kotlinx.coroutines.sync](#)

Kotlin Metadata class annotation found

- 15 occurrences in [androidx.core.view](#)

Kotlin Metadata class annotation found

- 176 occurrences in [kotlinx.coroutines](#)

Kotlin Metadata class annotation found

- 1 occurrences in [androidx.core.database.sqlite](#)

Kotlin Metadata class annotation found

- 51 occurrences in [kotlin.reflect](#)

Kotlin Metadata class annotation found

- 239 occurrences in [kotlinx.coroutines.flow](#)

Kotlin Metadata class annotation found

- 4 occurrences in [androidx.core.graphics.drawable](#)

Kotlin Metadata class annotation found

- 10 occurrences in [kotlin.coroutines.intrinsics](#)

Kotlin Metadata class annotation found

- 9 occurrences in [com.github.kittinunf.fuel.core.requests](#)

Kotlin Metadata class annotation found

- 2 occurrences in [kotlin.system](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.jvm.optionals](#)

Kotlin Metadata class annotation found

- 8 occurrences in [kotlinx.coroutines.android](#)

Kotlin Metadata class annotation found

- 28 occurrences in [kotlin.time](#)

Kotlin Metadata class annotation found

- 1 occurrences in [com.github.kittinunf.fuel.android.core](#)

Kotlin Metadata class annotation found

- 3 occurrences in [androidx.annotation](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.jvm.internal.unsafe](#)

Kotlin Metadata class annotation found

- 4 occurrences in [androidx.activity](#)

Kotlin Metadata class annotation found

- 113 occurrences in [kotlin](#)

Kotlin Metadata class annotation found

- 14 occurrences in [androidx.core.animation](#)

Kotlin Metadata class annotation found

- 19 occurrences in [kotlin.collections.builders](#)

Kotlin Metadata class annotation found

- 76 occurrences in [kotlinx.coroutines.internal](#)

Kotlin Metadata class annotation found

- 24 occurrences in [kotlin.coroutines](#)

Kotlin Metadata class annotation found

- 8 occurrences in [androidx.core.widget](#)

Kotlin Metadata class annotation found

- 62 occurrences in [com.github.kittinunf.fuel.core](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.collections.jdk8](#)

Kotlin Metadata class annotation found

- 4 occurrences in [kotlinx.android.extensions](#)

Kotlin Metadata class annotation found

- 11 occurrences in [kotlin.collections.unsigned](#)

Kotlin Metadata class annotation found

- 132 occurrences in [kotlin.collections](#)

Kotlin Metadata class annotation found

- 115 occurrences in [kotlinx.coroutines.channels](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.jvm.jdk8](#)

Kotlin Metadata class annotation found

- 1 occurrences in [owasp.mastgkotlin.util](#)

Kotlin Metadata class annotation found

- 5 occurrences in [kotlin.streams.jdk8](#)

Kotlin Metadata class annotation found

- 2 occurrences in [com.github.kittinunf.fuel.android.extension](#)

Kotlin Metadata class annotation found

- 2 occurrences in [com.github.kittinunf.fuel.core.deserializers](#)

Kotlin Metadata class annotation found

- 28 occurrences in [androidx.lifecycle](#)

Kotlin Metadata class annotation found

- 49 occurrences in [kotlin.jvm.internal](#)

Kotlin Metadata class annotation found

- 8 occurrences in [kotlinx.android.parcel](#)

Kotlin Metadata class annotation found

- 76 occurrences in [kotlin.sequences](#)

Kotlin Metadata class annotation found

- 5 occurrences in [org.jetbrains.anko.internals](#)

Kotlin Metadata class annotation found

- 7 occurrences in [owasp.mastgkotlin](#)

Kotlin Metadata class annotation found

- 3 occurrences in [androidx.core.content](#)

Kotlin Metadata class annotation found

- 2 occurrences in [com.github.kittinunf.fuel.android.util](#)

Kotlin Metadata class annotation found

- 8 occurrences in [org.jetbrains.anko.collections](#)

Kotlin Metadata class annotation found

- 68 occurrences in [kotlin.text](#)

Kotlin Metadata class annotation found

- 10 occurrences in [com.github.kittinunf.result](#)

Kotlin Metadata class annotation found

- 3 occurrences in [androidx.activity.result](#)

Kotlin Metadata class annotation found

- 5 occurrences in [kotlin.io.path](#)

Kotlin Metadata class annotation found

- 5 occurrences in [com.github.kittinunf.fuel.rx](#)

Kotlin Metadata class annotation found

- 1 occurrences in [androidx.core.location](#)

Kotlin Metadata class annotation found

- 24 occurrences in [kotlin.jvm.functions](#)

Kotlin Metadata class annotation found

- 20 occurrences in [kotlin.internal](#)

Kotlin Metadata class annotation found

- 13 occurrences in [com.github.kittinunf.fuel.core.interceptors](#)

Kotlin Metadata class annotation found

- 32 occurrences in [androidx.activity.result.contract](#)

Kotlin Metadata class annotation found

- 8 occurrences in [kotlin.properties](#)

Kotlin Metadata class annotation found

- 22 occurrences in [kotlinx.coroutines.scheduling](#)

Kotlin Metadata class annotation found

- 22 occurrences in [kotlin.jvm](#)

Kotlin Metadata class annotation found

- 23 occurrences in [kotlin.comparisons](#)

Kotlin Metadata class annotation found

- 2 occurrences in [androidx.core.content.res](#)

Kotlin Metadata class annotation found

- 7 occurrences in [kotlinx.coroutines.test](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.coroutines.cancellation](#)

Kotlin Metadata class annotation found

- 17 occurrences in [androidx.core.splashscreen](#)

Kotlin Metadata class annotation found

- 5 occurrences in [kotlin.concurrent](#)

Kotlin Metadata class annotation found

- 1 occurrences in [androidx.core.database](#)

Kotlin Metadata class annotation found

- 6 occurrences in [kotlin.annotation](#)

Kotlin Metadata class annotation found

- 12 occurrences in [androidx.core.transition](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.random.jdk8](#)

Kotlin Metadata class annotation found

- 7 occurrences in [androidx.core.text](#)

Kotlin Metadata class annotation found

- 2 occurrences in [kotlinx.coroutines.intrinsics](#)

Kotlin Metadata class annotation found

- 1 occurrences in [androidx.core.net](#)

Kotlin Metadata class annotation found

- 9 occurrences in [androidx.lifecycle.viewmodel](#)

Kotlin Metadata class annotation found

- 2 occurrences in [kotlin.internal.jdk8](#)

Kotlin Metadata class annotation found

- 1 occurrences in [collections](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.time.jdk8](#)

Kotlin Metadata class annotation found

- 38 occurrences in [kotlin.io](#)

Kotlin Metadata class annotation found

- 9 occurrences in [kotlin.jvm.internal.markers](#)

Kotlin Metadata class annotation found

- 12 occurrences in [androidx.core.os](#)

Kotlin Metadata class annotation found

- 11 occurrences in [kotlin.contracts](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.js](#)

Kotlin Metadata class annotation found

- 3 occurrences in [androidx.annotation.experimental](#)

Kotlin Metadata class annotation found

- 2 occurrences in [kotlin.experimental](#)

Kotlin Metadata class annotation found

- 1 occurrences in [kotlin.jdk7](#)

**Low ● Calls to Kotlin assertions leak information**

The class `kotlin.jvm.internal.Intrinsics` contains methods injected by the Kotlin compiler in order to perform checks or assertions on parameters and other elements of the code being null. This is useful for example to guarantee interoperability with Java but results in methods that leak information via their parameters. Most obfuscation techniques hide the name of the parameter but not strings containing the name of the parameter. E.g., a call to `kotlin.jvm.internal.Intrinsics.checkNotNullParameter(secretParameterName, "secretParameterName")` might be obfuscated as `a.z(b, "secretParameterName")`, leaking the actual name of the parameter. This information can help reverse engineers better and easier understand the behavior of your app.

Recommendations

The calls to Kotlin assertions leaking information or the strings they take as argument should be removed in release builds. This can be done automatically with Guardsquare's ProGuard or DexGuard. R8 can only remove the calls to Kotlin assertions but not obfuscate their parameters.

**Fix with ProGuard**

If `-keepkotlinmetadata` is in your ProGuard configuration and `-dontobfuscate` is not specified the input strings of kotlin assertions are removed.

As an alternative you can remove completely the calls to Kotlin assertions adding this to your ProGuard configuration:

`-assumenosideeffects class`

```
kotlin.jvm.internal.Intrinsics {  
    public static void checkNotNull(...);  
    public static void checkExpressionValueIsNotNull(...);  
    public static void checkNotNullExpressionValue(...);  
    public static void checkParameterIsNotNull(...);  
    public static void checkNotNullParameter(...);  
    public static void checkReturnedValueIsNotNull(...);  
    public static void checkFieldIsNotNull(...);  
    public static void throwUninitializedPropertyAccessException(...);  
    public static void throwNpe(...);  
    public static void throwJavaNpe(...);  
    public static void throwAssert(...);  
    public static void throwIllegalArgument(...);  
    public static void throwIllegalState(...);  
}
```

78 Findings

- 1 occurrences in `com.github.kittinunf.fuel.android.core`
- 164 occurrences in `androidx.core.util`
- 6 occurrences in `collections`
- 3 occurrences in `androidx.activity.result`
- 10 occurrences in `kotlin.jvm.optionals`
- 3 occurrences in `com.github.kittinunf.fuel.toolbox`
- 50 occurrences in `kotlin.coroutines`
- 11 occurrences in `kotlin.properties`
- 267 occurrences in `kotlin.io.path`
- 2 occurrences in `androidx.core.location`
- 1 occurrences in `kotlinx.coroutines.flow`
- 57 occurrences in `androidx.core.transition`
- 216 occurrences in `kotlin.io`
- 7 occurrences in `androidx.core.database`
- 3 occurrences in `kotlin.internal.jdk8`
- 12 occurrences in `androidx.activity`
- 26 occurrences in `androidx.core.os`
- 122 occurrences in `kotlin.ranges`
- 15 occurrences in `com.github.kittinunf.fuel.core.requests`
- 14 occurrences in `com.github.kittinunf.fuel.rx`
- 593 occurrences in `org.jetbrains.anko`
- 42 occurrences in `androidx.core.splashscreen`
- 14 occurrences in `com.github.kittinunf.fuel.core.interceptors`
- 1024 occurrences in `kotlin.text`
- 2 occurrences in `owasp.mastgkotlin.util`
- 30 occurrences in `com.github.kittinunf.result`
- 1 occurrences in `kotlin.random.jdk8`
- 8 occurrences in `com.github.kittinunf.fuel.util`
- 96 occurrences in `androidx.core.view`
- 1 occurrences in `kotlin.jdk7`
- 11 occurrences in `kotlin.coroutines.intrinsics`
- 61 occurrences in `com.github.kittinunf.fuel`
- 2 occurrences in `kotlin.text.jdk8`
- 14 occurrences in `androidx.lifecycle.viewmodel`
- 4 occurrences in `kotlin.internal.jdk7`
- 2 occurrences in `kotlin.system`

- 55 occurrences in [androidx.core.animation](#)
- 2 occurrences in [kotlin.collections.jdk8](#)
- 84 occurrences in [androidx.lifecycle](#)
- 19 occurrences in [androidx.core.content.res](#)
- 37 occurrences in [kotlin.concurrent](#)
- 6 occurrences in [com.github.kittinunf.fuel.android.extension](#)
- 24 occurrences in [org.jetbrains.anko.collections](#)
- 51 occurrences in [kotlin.collections.builders](#)
- 1 occurrences in [com.github.kittinunf.fuel.android.util](#)
- 2 occurrences in [kotlin.time.jdk8](#)
- 3 occurrences in [kotlinx.coroutines.debug.internal](#)
- 74 occurrences in [androidx.activity.result.contract](#)
- 169 occurrences in [com.github.kittinunf.fuel.core](#)
- 4491 occurrences in [kotlin.collections](#)
- 18 occurrences in [androidx.core.graphics.drawable](#)
- 29 occurrences in [androidx.savedstate](#)
- 14 occurrences in [androidx.core.widget](#)
- 27 occurrences in [org.jetbrains.anko.custom](#)
- 11 occurrences in [kotlin.jvm](#)
- 1316 occurrences in [kotlin.collections.unsigned](#)
- 37 occurrences in [org.jetbrains.anko.internals](#)

- 53 occurrences in [androidx.core.text](#)
- 432 occurrences in [kotlin.sequences](#)
- 5 occurrences in [kotlinx.coroutines](#)
- 1 occurrences in [kotlin.contracts](#)
- 19 occurrences in [kotlin.streams.jdk8](#)
- 60 occurrences in [owasp.mastgkotlin](#)
- 16 occurrences in [kotlin.reflect](#)
- 58 occurrences in [kotlin.time](#)
- 5 occurrences in [androidx.core.net](#)
- 13 occurrences in [kotlin.internal](#)
- 77 occurrences in [kotlin.comparisons](#)
- 175 occurrences in [kotlin](#)
- 20 occurrences in [androidx.core.content](#)
- 216 occurrences in [androidx.core.graphics](#)
- 58 occurrences in [kotlin.jvm.internal](#)
- 3 occurrences in [com.github.kittinunf.fuel.core.deserializers](#)
- 32 occurrences in [kotlin.random](#)
- 11 occurrences in [kotlin.coroutines.jvm.internal](#)
- 4 occurrences in [androidx.core.database.sqlite](#)
- 1 occurrences in [kotlinx.coroutines.flow.internal](#)
- 3 occurrences in [kotlinx.coroutines.internal](#)

