Comparative Security Assessment Report for Flutter Application (V2.0.0 – V3.27.0)

This document outlines a detailed security testing report performed on five different versions of the application built using Flutter, specifically: **V2.0.0**, **V2.10.0**, **V3.7.0**, **V3.16.6**, and **V3.27.0**. The goal of this assessment is to analyze and compare the security posture of each version against a set of defined attack scenarios and vulnerabilities.

The testing is divided into two primary scopes:

1. **Network Interception Tests** – To evaluate the application's resistance to various network-layer attacks, including:
    - Traffic Interception using Reflutter
    - Traffic Interception using Zygisk Reflutter
    - Traffic Interception using OpenVPN (OVPN)
    - TLS Interception with Frida Bypass
2. **Application Layer Tests** – To verify secure implementation of sensitive data handling at the application level, focusing on:
    - Local Storage Inspection (SharedPreferences)
    - Secure Logging Validation

Each test case was conducted across all five Flutter versions, with the results recorded using the following indicators:

- **V** = Vulnerable / Exploitable
- **X** = Not Vulnerable / Not Exploitable

Network Traffic Interception Test

Subsection: Traffic Interception using Reflutter

Overview

This test evaluates the application's resistance to network traffic interception by using **Reflutter**, a tool designed to decompile and repack Flutter-based Android applications. Reflutter allows deeper inspection and modification of the app's internal logic, which may indirectly facilitate traffic interception. It enables the testing of HTTP and TLS-encrypted traffic interception by generating a modified APK suitable for analysis with tools like **Burp Suite**.

Test Objective

To determine whether network traffic, including HTTPS requests, can be intercepted after rebuilding the Flutter application using Reflutter.

Test Environment

- Interception Tool: **Burp Suite**
- Bypass Tool: **Reflutter (**https://github.com/ptswarm/reFlutter**)**
- Signing Tool: **Uber APK Signer** (https://github.com/patrickfav/uber-apk-signer)
- Platform: Android
- Test Date: **13 January 2025**

Tools and Setup

1. Install Reflutter via Python.
2. Run `reflutter` on the target APK, specifying the device IP address.
3. Re-sign the modified APK using Uber APK Signer.
4. Install the modified APK on the test device.
5. Configure and run Burp Suite to intercept HTTP and HTTPS traffic.

Results Summary

	2.0.0		2.10.0		3.7.0		3.16.6		3.27.0	
	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS
	V	V	V	V	V	V	V	V	X	X

V = Interception successful (Vulnerable)

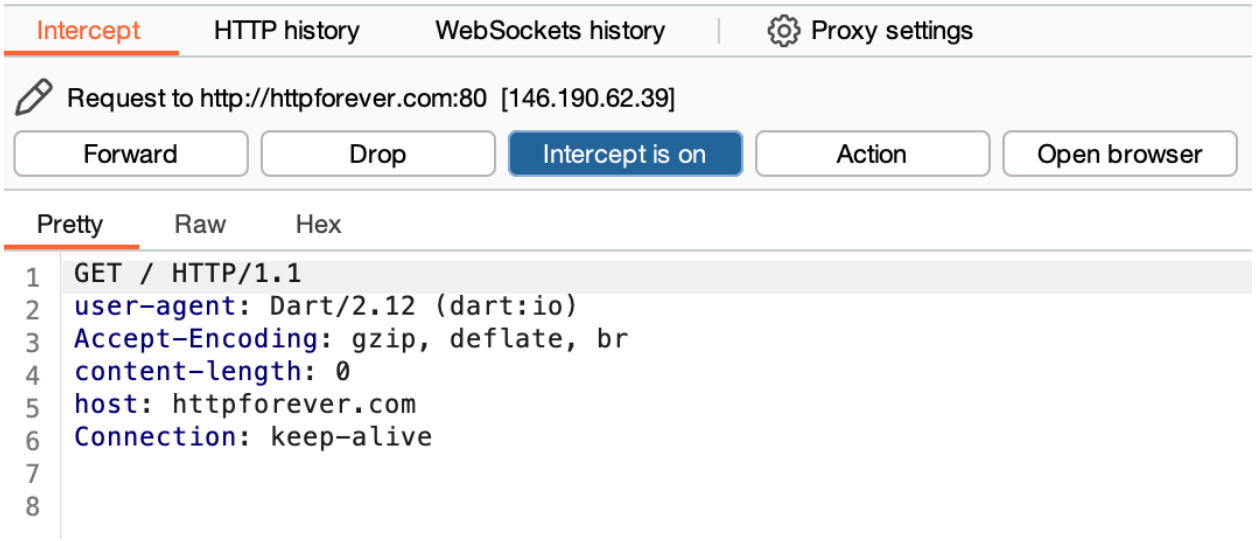
X = Interception blocked (Not Vulnerable)

Screenshots

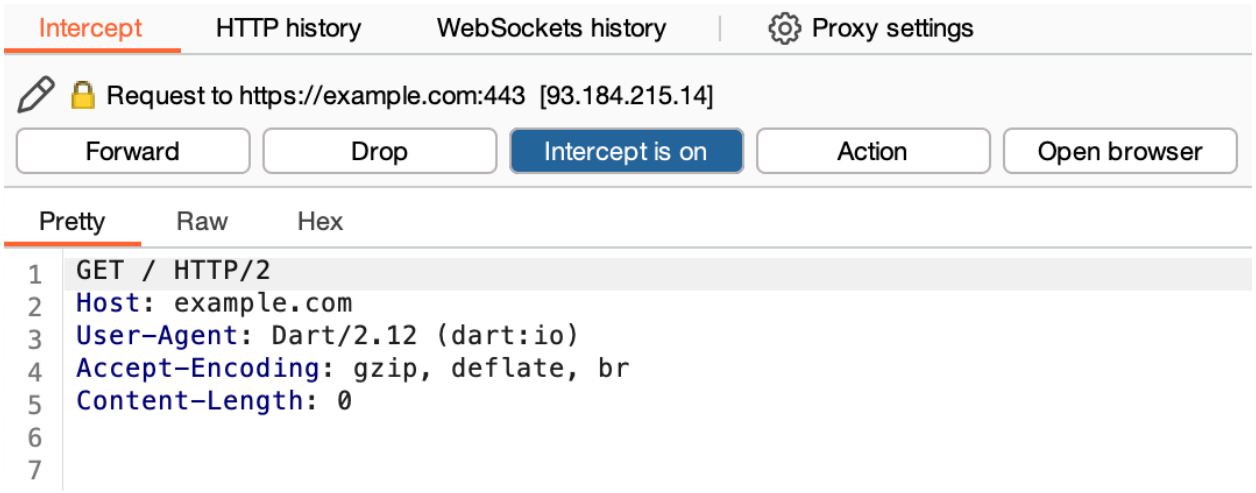
V2.0.0 (Reflutter Process)

```
[redacted] 200 % reflutter 200.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 172.20.10.11

 Wait...


SnapshotHash: 5b97292b25f0a715613b7a28e0734f77
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true

[redacted] 200 % java -jar ../uber-apk-signer-1.3.0.jar --apks release.RE.apk
source:
[redacted]/200
zipalign location: BUILT_IN
[redacted]/mac-zipalign-33_0_21628444798522796018 9.tmp
keystore:
[redacted]/temp_1812304998669923375_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: [redacted]/release.RE.apk (16.36 MiB)
        checksum: cd3ec257d48e3643e96f3dc0f0935857225b2cb8cb18d1f1ed5cda4725bde412 (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: [redacted]/release.RE-aligned-debugSigned.apk (16.37 MiB)
        checksum: aaf1d3fd4c30919273b032b972defff977c800e245d45d62ac237dfd2ba9f6a2 (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                19 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Mon Jan 13 16:49:32 WIB 2025][v1.3.0]
Successfully processed 1 APKs and 0 errors in 1.74 seconds.
```

V2.0.0 (HTTP)



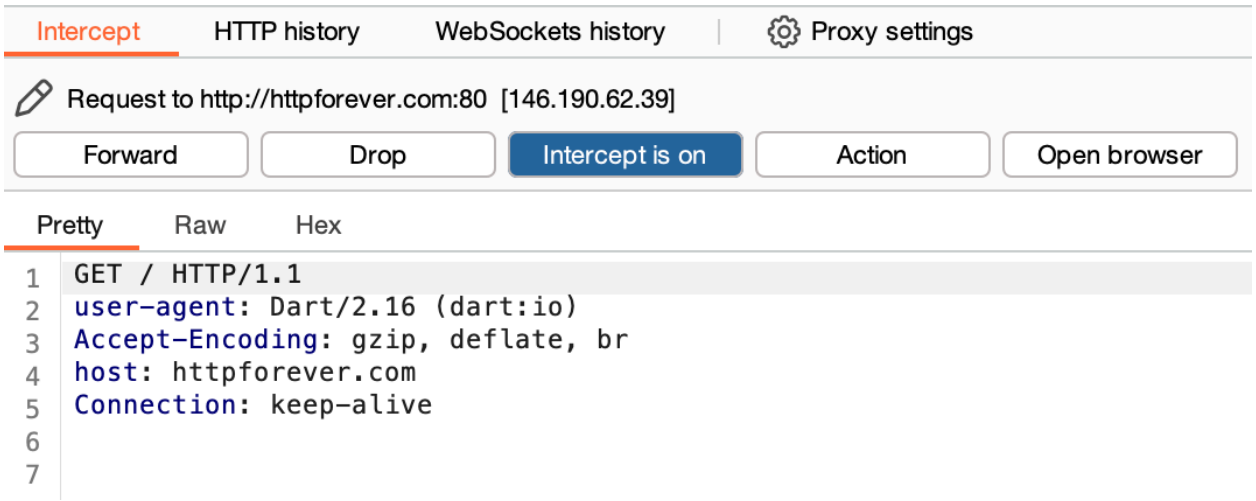
V2.0.0 (HTTPS)



V2.10.0 (Reflutter Process)

```
[redacted] 210 % reflutter 210.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 172.20.10.11

 Wait...


SnapshotHash: d56742caf7b3b3f4bd2df93a9bbb5503
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true

[redacted] 210 % java -jar ../uber-apk-signer-1.3.0.jar --apks release.RE.apk
source:
[redacted]/210
zipalign location: BUILT_IN
[redacted]/mac-zipalign-33_0_24764391054363746199.tmp
keystore:
[redacted]/temp_3934440731600176444_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: [redacted]/release.RE.apk (16.65 MiB)
        checksum: fa2060f832059b6a49fee3b9594f634a84790f9ff81c5f464e321db0dfb2cf8f (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: [redacted]/release.RE-aligned-debugSigned.apk (16.65 MiB)
        checksum: e00f73657644c1c841d018272996a7cd49752bb75d34be5531c106a85a3ebce (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                29 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Mon Jan 13 16:53:01 WIB 2025][v1.3.0]
Successfully processed 1 APKs and 0 errors in 1.26 seconds.
```

V2.10.0 (HTTP)



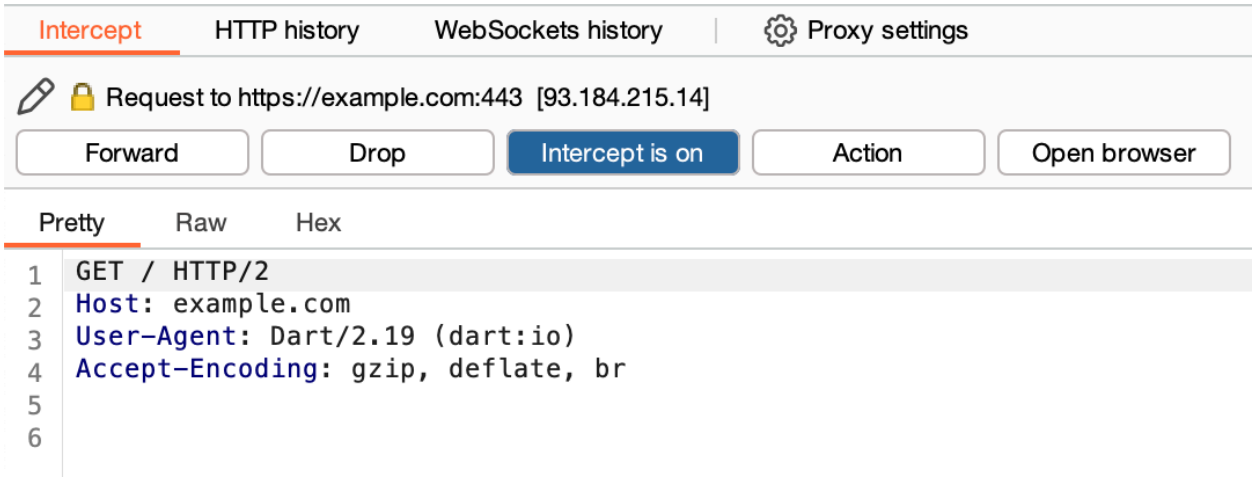
V2.10.0 (HTTPS)

V3.7.0 (Reflutter Process)

```
[redacted] 3700 % reflutter 3700.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 172.20.10.11

 Wait...


SnapshotHash: 501ef5cbd64ca70b6b42672346af6a8a
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true

[redacted] 3700 % java -jar ../uber-apk-signer-1.3.0.jar --apks release.RE.apk
source:
[redacted]/3700
zipalign location: BUILT_IN
[redacted]/mac-zipalign-33_0_29118390049999692280.tmp
keystore:
[redacted]/temp_6176671021533168359_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: [redacted]/release.RE.apk (17.55 MiB)
        checksum: 1582414bc1fc5626fa9c20498d3d8af7fbf9e0f99f3b54348c51e827c3ae7870 (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: [redacted]/release.RE-aligned-debugSigned.apk (17.55 MiB)
        checksum: 4a8e713f2cf1d483ca135dd32b427a86db7d23799e9765a70c32436289634d42 (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                20 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Mon Jan 13 16:56:54 WIB 2025][v1.3.0]
Successfully processed 1 APKs and 0 errors in 1.46 seconds.
```

V3.7.0 (HTTP)

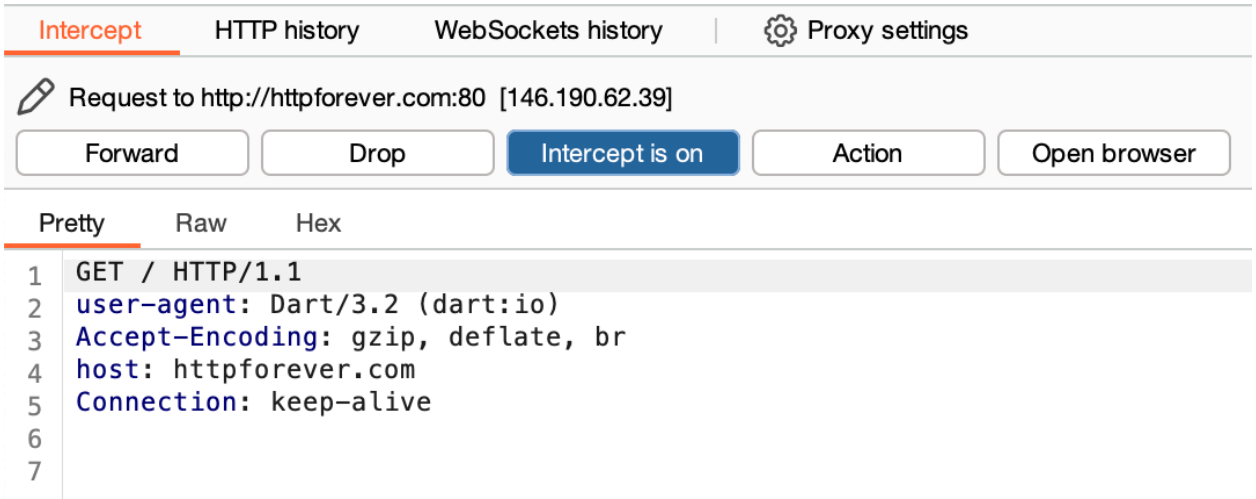
V3.7.0 (HTTPS)



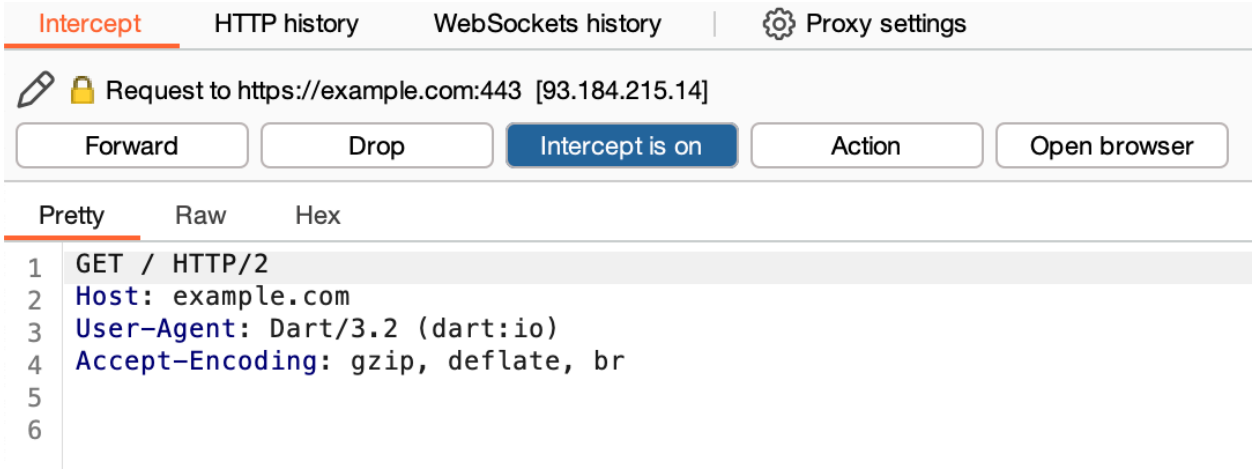
V3.16.6 (Reflutter Process)

```
[redacted] 31660 % reflutter 31660.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 172.20.10.11

 Wait...


SnapshotHash: f71c76320d35b65f1164dbaa6d95fe09
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true

[redacted] 31660 % java -jar ../uber-apk-signer-1.3.0.jar --apks release.RE.apk
source:
[redacted]/31660
zipalign location: BUILT_IN
[redacted]/mac-zipalign-33_0_2465589302545574937 8.tmp
keystore:
[redacted]/temp_16638466647577258198_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: [redacted]/release.RE.apk (18.02 MiB)
        checksum: 6282e1bcf591c5090ea2c6007100708ce1b93dfee5b7f882d66366ee77c12542 (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: [redacted]/release.RE-aligned-debugSigned.apk (18.03 MiB)
        checksum: 67440ab17b5b0a7aa62b66f8b48ab0ada499ecbf5765930586b32a31974f3193 (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                20 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Mon Jan 13 16:59:11 WIB 2025][v1.3.0]
Successfully processed 1 APKs and 0 errors in 1.20 seconds.
```

V3.16.6 (HTTP)



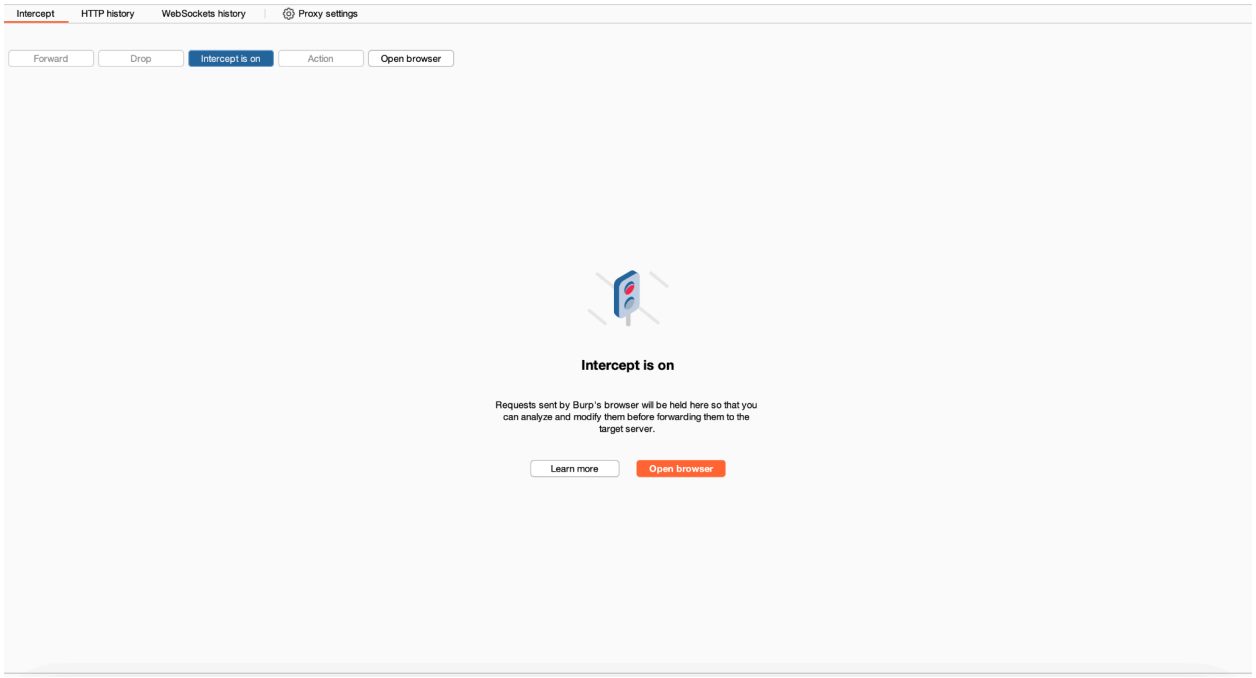
V3.16.6 (HTTPS)



V3.27.0 (Reflutter Process)

```
[redacted] 32700 % reflutter 32700.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 172.20.10.11

 Wait...


SnapshotHash: f956f595844a2f845a55707faaaa51e4
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true

[redacted] 32700 % java -jar ../uber-apk-signer-1.3.0.jar --apks release.RE.apk
source:
[redacted]/32700
zipalign location: BUILT_IN
[redacted]/mac-zipalign-33_0_21234304027505842026.tmp
keystore:
[redacted]/temp_7557593111258408036_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: [redacted]/release.RE.apk (19.17 MiB)
        checksum: b21b23d27e62db98325637d429a85b60418d7d97f825009358f08eb42497007b (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: [redacted]/release.RE-aligned-debugSigned.apk (19.18 MiB)
        checksum: 5ace12672a76874ecf9498a6ca3054d47c80f70123f35716f042bdc65635899b (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                55 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Mon Jan 13 17:00:57 WIB 2025][v1.3.0]
Successfully processed 1 APKs and 0 errors in 1.33 seconds.
```

V3.27.0 (HTTP & HTTPS Not Intercepted)



Subsection: Traffic Interception using Zygisk Reflutter

Overview

This test evaluates the application's vulnerability to network traffic interception using **Zygisk Reflutter**, a module built for **Magisk/Zygisk** that allows interception of Flutter app traffic without modifying the APK. Unlike the original Reflutter tool that modifies app binaries, Zygisk Reflutter **hooks into Flutter's native networking layer at runtime**, enabling HTTP and HTTPS traffic to be redirected through a proxy, such as **Burp Suite**, on a **rooted device**.

The module works by injecting a proxy configuration into the Dart VM within Flutter apps, allowing decrypted traffic capture **without the need to disable SSL pinning manually** — particularly effective when pinning is weak or absent.

Test Objective

To determine whether the application traffic (HTTP and HTTPS) can be intercepted at runtime using Zygisk Reflutter without modifying the application binary.

Test Environment

- Interception Tool: **Burp Suite**
- Bypass Tool: **Zygisk Reflutter v1.0.1 (**https://github.com/yohanes/zygisk-reflutter/releases/tag/v1.0.1**)**
- Device: **Rooted Android with Magisk & Zygisk Enabled**
- Test Date: **13 January 2025**

Tools and Setup

1. Install Magisk with **Zygisk** support on the test device.
2. Install **Zygisk Reflutter** module from GitHub release v1.0.1.
3. Download and configure the provided **proxy library.**.
4. Set environment variables or configurations to redirect Flutter traffic to the desired proxy.

5. Start **Burp Suite** and intercept traffic.

Results Summary

	2.0.0	2.10.0	3.7.0	3.16.6	3.27.0
	X	V	V	V	X

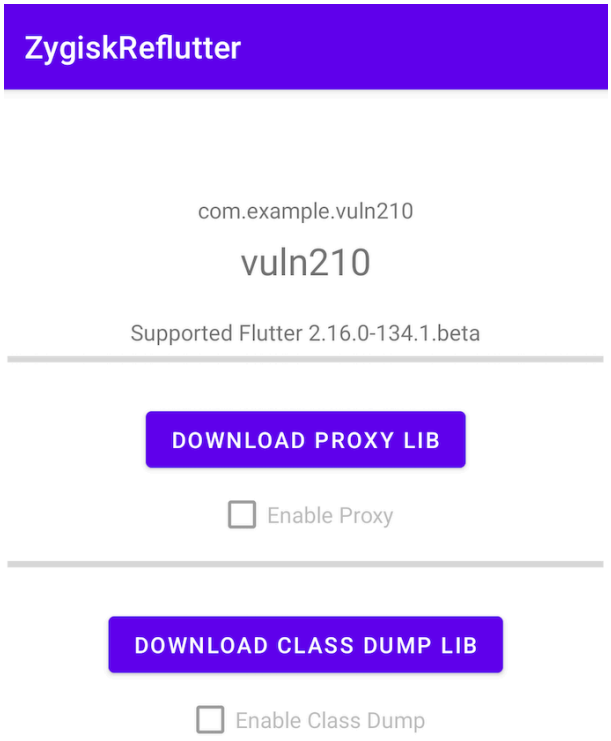
V = Interception successful (Vulnerable)

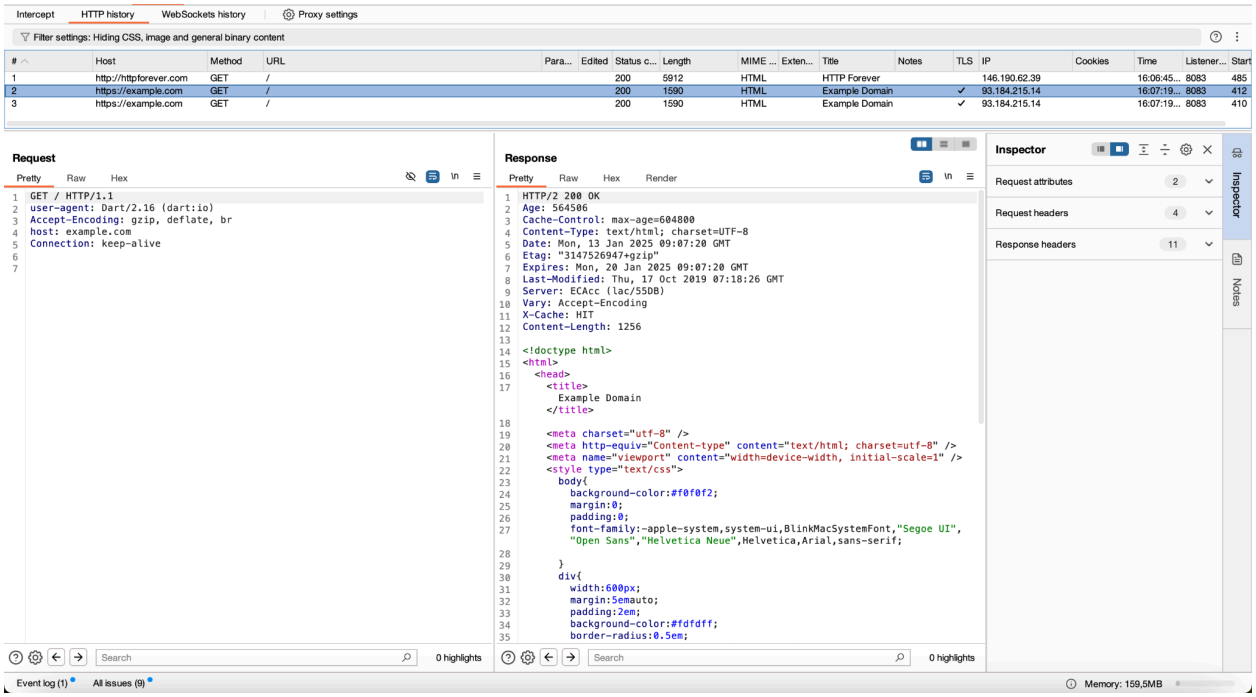
X = Interception blocked (Not Vulnerable)

Screenshots

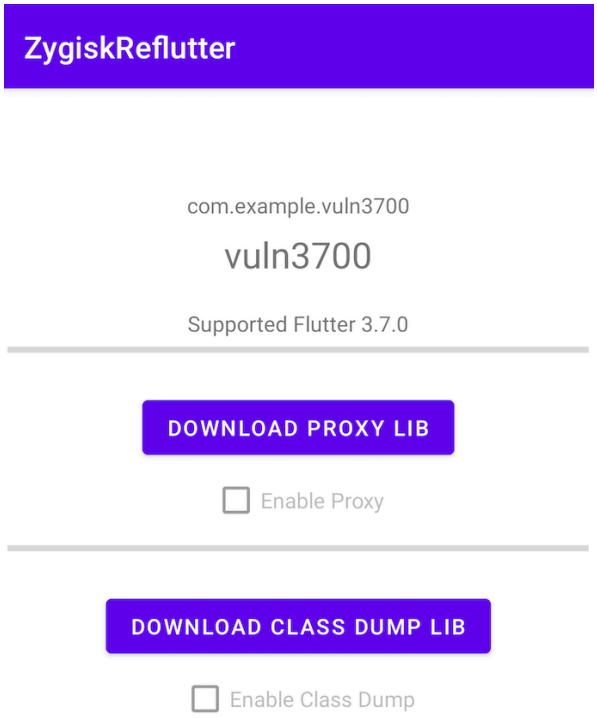
V2.0.0 (Failed)

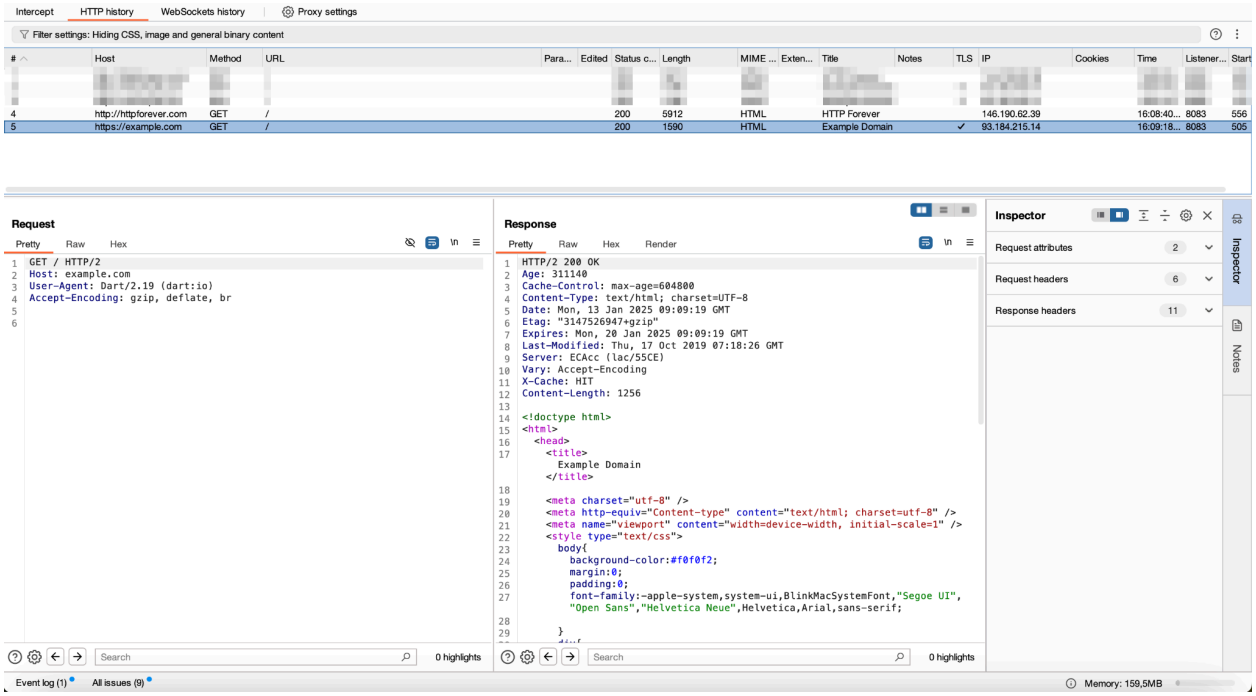
V2.10.0 (Successful)



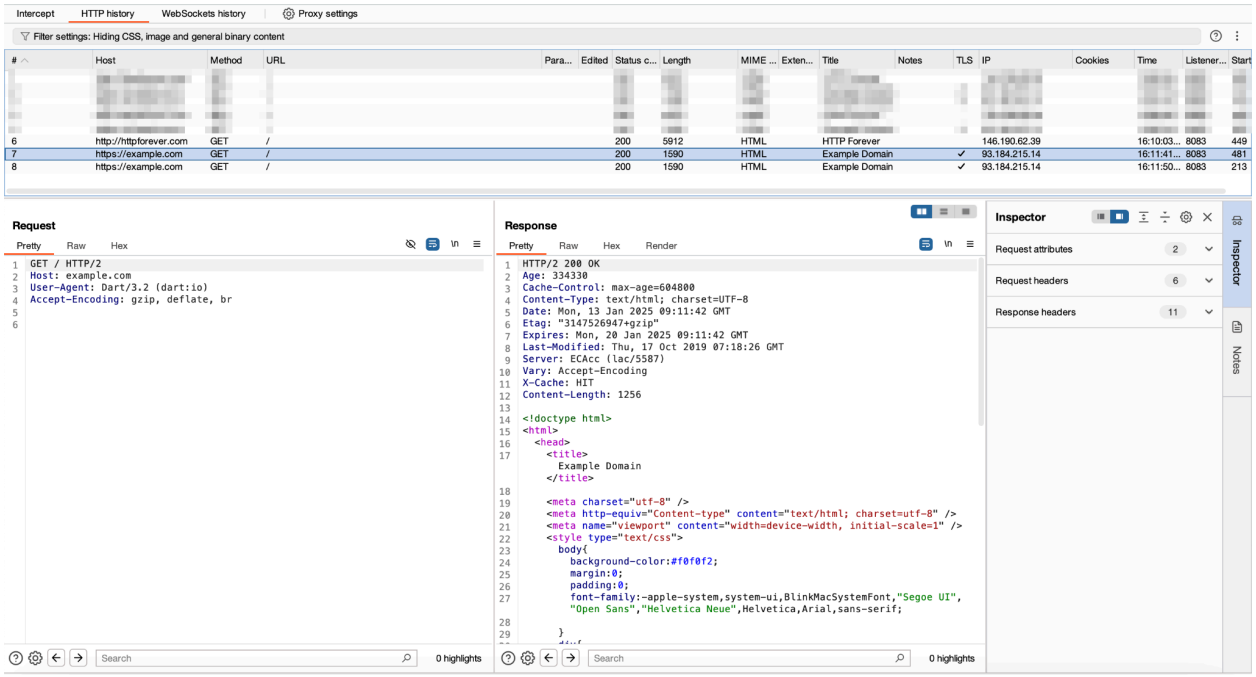


V3.7.0 (Successful)





V3.16.6 (Successful)



V3.27.0 (Failed)

Subsection: Traffic Interception using Frida Script (frida-flutterproxy)

Overview

This test focuses on evaluating the application's resistance to runtime traffic interception using **Frida**, a dynamic instrumentation toolkit widely used for reverse engineering and bypassing security mechanisms such as SSL pinning. The script used in this test is from the public repository `frida-flutterproxy`, which targets Flutter applications by hooking into native methods responsible for network communication.

Unlike Reflutter or Zygisk Reflutter, this approach does **not require APK repacking or Zygisk**, but instead relies on **runtime memory injection** via Frida to manipulate the Flutter networking behavior and force it to route traffic through a specified proxy.

Test Objective

To determine whether the application traffic (HTTP and HTTPS) can be intercepted using **Frida script injection**.

Test Environment

- Interception Tool: **Burp Suite**
- Instrumentation Tool: **Frida with frida-flutterproxy script**
- Frida Script: https://github.com/hackcatml/frida-flutterproxy
- Device: **Rooted Android device with Frida server running**
- Test Date: **13 January 2025**

Tools and Setup

1. Start the **Frida server** on the rooted Android test device.
2. Clone or download the script from frida-flutterproxy.
3. Run the Frida script using a desktop/laptop client (`frida -U -n <app> -l flutter_proxy.js`).
4. Set up **proxy interception settings** (as defined in the script configuration).

5. Use **Burp Suite** to intercept and analyze HTTP/TLS traffic.

Results Summary

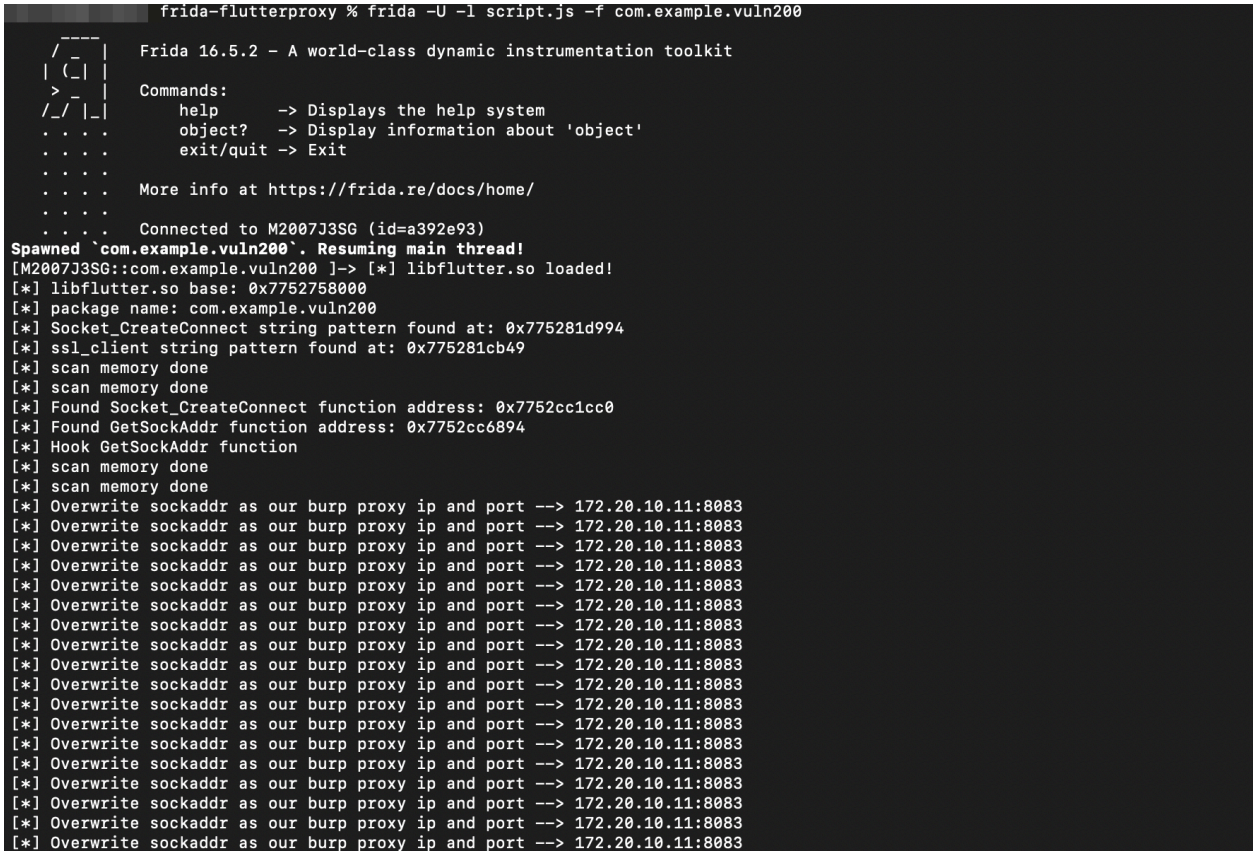
	2.0.0		2.10.0		3.7.0		3.16.6		3.27.0	
	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS
	V	X	V	X	V	V	V	V	V	V

V = Interception successful (Vulnerable)

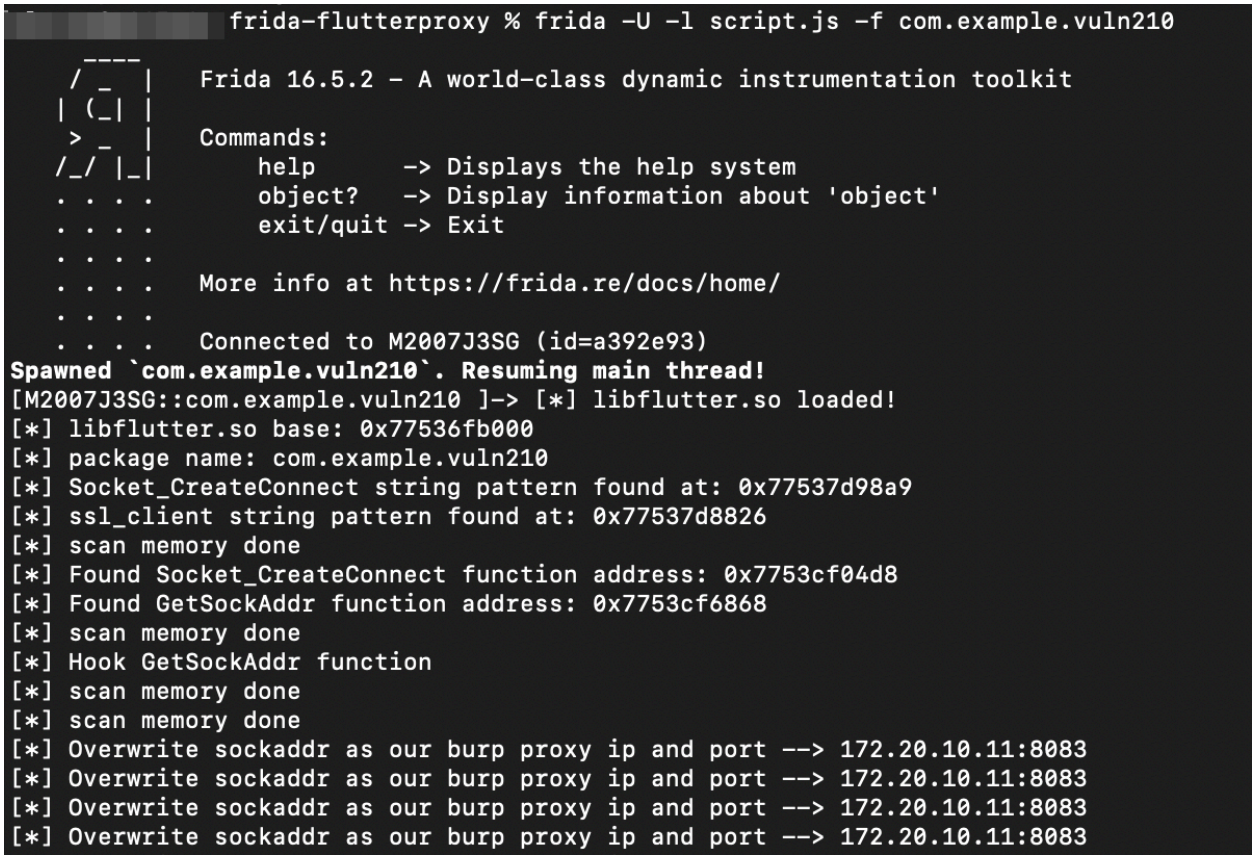
X = Interception blocked (Not Vulnerable)

Screenshots

V2.0.0 (Partially Successful)



V2.10.0 (Partially Successful)



V3.7.0 (Successful)

```
[redacted] frida-flutterproxy % frida -U -l script.js -f com.example.vuln3700
     ____
    / _  |   Frida 16.5.2 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to M2007J3SG (id=a392e93)
Spawned `com.example.vuln3700`. Resuming main thread!
[M2007J3SG::com.example.vuln3700 ]-> [*] libflutter.so loaded!
[*] libflutter.so base: 0x7751e1e000
[*] package name: com.example.vuln3700
[*] ssl_client string pattern found at: 0x7751f0f6ab
[*] Socket_CreateConnect string pattern found at: 0x7751f1086a
[*] scan memory done
[*] scan memory done
[*] Found Socket_CreateConnect function address: 0x7752473820
[*] Found GetSockAddr function address: 0x7752479a20
[*] scan memory done
[*] Hook GetSockAddr function
[*] Found adrp add address: 0x77523b5370
[*] Found verify_cert_chain function address: 0x77523b528c
[*] Hook verify_cert_chain function
[*] scan memory done
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
```

V3.16.6 (Successful)

```
[redacted] frida-flutterproxy % frida -U -l script.js -f com.example.vuln31660
     ____
    / _  |   Frida 16.5.2 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to M2007J3SG (id=a392e93)
Spawned `com.example.vuln31660`. Resuming main thread!
[M2007J3SG::com.example.vuln31660 ]-> [*] libflutter.so loaded!
[*] libflutter.so base: 0x775223c000
[*] package name: com.example.vuln31660
[*] ssl_client string pattern found at: 0x77523347bf
[*] Socket_CreateConnect string pattern found at: 0x7752335aaa
[*] scan memory done
[*] scan memory done
[*] Found Socket_CreateConnect function address: 0x7752953408
[*] Found GetSockAddr function address: 0x77529594b8
[*] Hook GetSockAddr function
[*] scan memory done
[*] Found adrp add address: 0x775283a044
[*] Found verify_cert_chain function address: 0x7752839f60
[*] Hook verify_cert_chain function
[*] scan memory done
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
```

V3.27.0 (Successful)

```
[redacted] frida-flutterproxy % frida -U -l script.js -f com.example.vuln32700
     ____
    / _  |   Frida 16.5.2 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to M2007J3SG (id=a392e93)
Spawned `com.example.vuln32700`. Resuming main thread!
[M2007J3SG::com.example.vuln32700 ]-> [*] libflutter.so loaded!
[*] libflutter.so base: 0x775404e000
[*] package name: com.example.vuln32700
[*] Socket_CreateConnect string pattern found at: 0x77542074ea
[*] ssl_client string pattern found at: 0x7754206225
[*] scan memory done
[*] Found Socket_CreateConnect function address: 0x775485d440
[*] Found GetSockAddr function address: 0x7754863ad0
[*] Hook GetSockAddr function
[*] scan memory done
[*] scan memory done
[*] Found adrp add address: 0x7754738140
[*] Found verify_cert_chain function address: 0x7754738060
[*] Hook verify_cert_chain function
[*] scan memory done
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] Overwrite sockaddr as our burp proxy ip and port --> 172.20.10.11:8083
[*] verify cert bypass
```

Subsection: Traffic Interception using OVPN (OpenVPN Tunnel)

Overview

This test examines whether application network traffic can be intercepted when routed through a controlled **OpenVPN tunnel**. Using a Kali Linux machine configured with a **bridged network adapter**, an OpenVPN server is deployed to establish a secure tunnel, through which the mobile device's traffic is routed. This setup allows **full network traffic interception** on both HTTP and HTTPS protocols via Burp Suite, provided that SSL pinning or other transport-level protections are not enforced.

Unlike Frida or Zygisk-based methods, this approach is **non-invasive**, requiring no binary modification or runtime instrumentation. Its effectiveness relies solely on the app's use of secure transmission techniques such as certificate pinning.

Test Objective

To evaluate whether application traffic can be transparently intercepted by routing all device traffic through a custom VPN tunnel.

Test Environment

- Host: **Kali Linux** with **Bridge Network Adapter**
- VPN Server: **OpenVPN (**https://github.com/Nyr/openvpn-install**)**
- Client: **Android device with imported .ovpn profile**
- Proxy Tool: **Burp Suite**
- Test Date: **13 January 2025**

Tools and Setup

1. Set up **Kali Linux** with a **bridged adapter** to allow traffic visibility.
2. Clone and run `openvpn-install.sh` to deploy an OpenVPN server.
3. Generate a client profile (`.ovpn`) and transfer it to the Android device.

4. Import the profile using any OpenVPN-compatible app (e.g., OpenVPN for Android).
5. Configure **Burp Suite** to listen on port **8080** for interception.
6. Ensure the Android device is set to use the VPN as its default network.

Results Summary

	2.0.0		2.10.0		3.7.0		3.16.6		3.27.0	
	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS	HTTP	TLS
	V	X	V	X	V	X	V	X	V	X

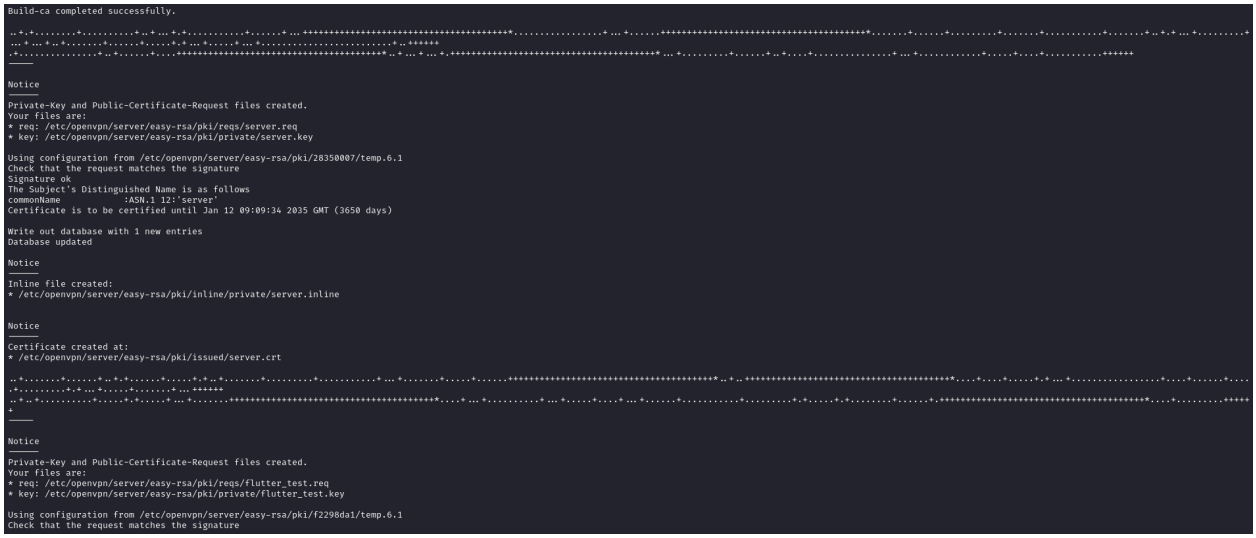
V = Interception successful (Vulnerable)

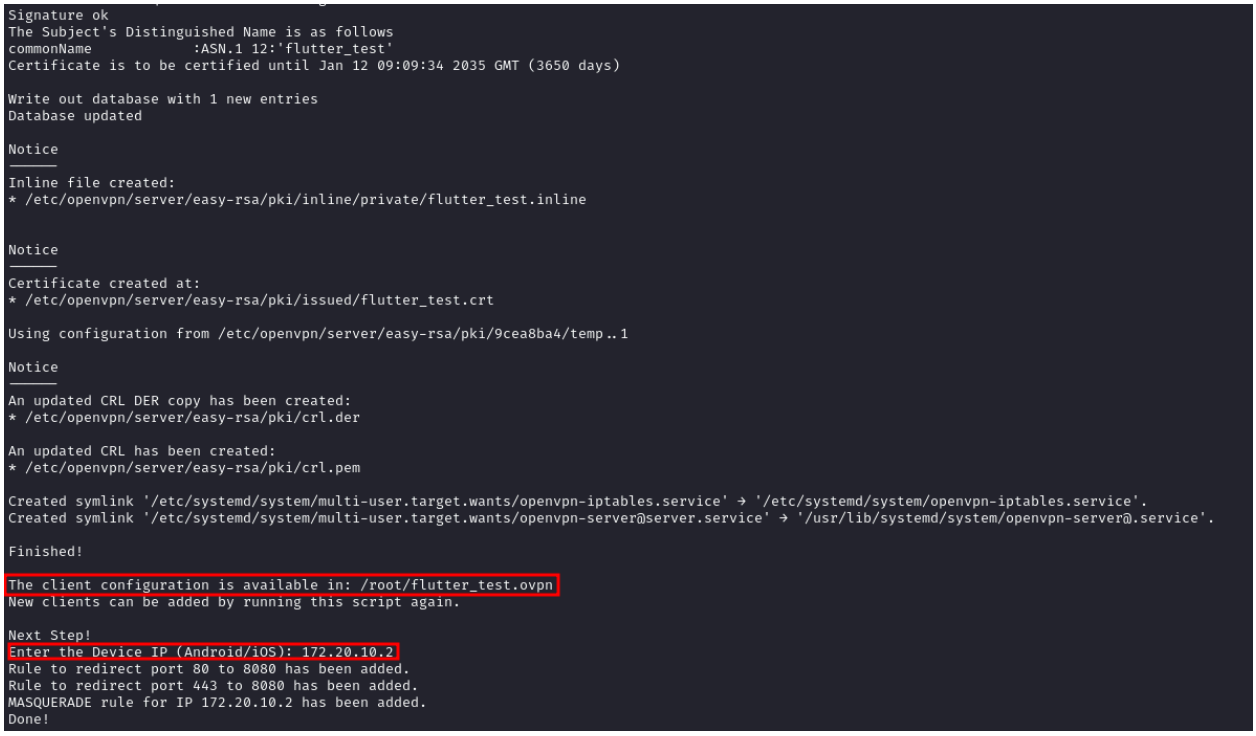
X = Interception blocked (Not Vulnerable)

OVPN Setup Documentation

```
Welcome to this OpenVPN road warrior installer!

This server is behind NAT. What is the local IPv4 address or hostname?
Local IPv4 address [172.20.10.3]:

Which protocol should OpenVPN use?
   1) UDP (recommended)
   2) TCP
Protocol [1]:

What port should OpenVPN listen to?
Port [1194]:

Select a DNS server for the clients:
   1) Current system resolvers
   2) Google
   3) 1.1.1.1
   4) OpenDNS
   5) Quad9
   6) AdGuard
DNS server [1]:

Enter a name for the first client:
Name [client]: flutter_test

OpenVPN installation is ready to begin.
Press any key to continue...
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.3 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.7 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [258 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [195 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [877 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [23.3 kB]
Fetched 70.5 MB in 30s (2,329 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.3.2-2).
Suggested packages:
  resolvconf openvpn-dco-dkms openvpn-systemd-resolved
The following NEW packages will be installed:
  openvpn
The following packages will be upgraded:
  ca-certificates
1 upgraded, 1 newly installed, 0 to remove and 862 not upgraded.
Need to get 164 kB/824 kB of archives.
After this operation, 1,830 kB of additional disk space will be used.
Get:1 http://mirror.primelink.net.id/kali kali-rolling/main amd64 ca-certificates all 20241223 [164 kB]
```

```
Get:1 http://mirror.primelink.net.id/kali kali-rolling/main amd64 ca-certificates all 20241223 [164 kB]
Fetched 164 kB in 1s (153 kB/s)
Preconfiguring packages ...
(Reading database ... 400709 files and directories currently installed.)
Preparing to unpack .../ca-certificates_20241223_all.deb ...
Unpacking ca-certificates (20241223) over (20240203) ...
Selecting previously unselected package openvpn.
Preparing to unpack .../openvpn_2.6.12-1_amd64.deb ...
Unpacking openvpn (2.6.12-1) ...
Setting up openvpn (2.6.12-1) ...
update-rc.d: We have no instructions for the openvpn init script.
update-rc.d: It looks like a network service, we disable it.
openvpn.service is a disabled or a static unit, not starting it.
Setting up ca-certificates (20241223) ...
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt, it does not contain exactly one certificate or CRL
7 added, 1 removed; done.
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2024.4.0) ...
Processing triggers for ca-certificates (20241223) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for ca-certificates-java (20240118) ...
done.

Notice
------
'init-pki' complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /etc/openvpn/server/easy-rsa/pki

Using Easy-RSA configuration:
* undefined
.+..++++++

Notice
------
CA creation complete. Your new CA certificate is at:
* /etc/openvpn/server/easy-rsa/pki/ca.crt

Create an OpenVPN TLS-AUTH|TLS-CRYPT-V1 key now: See 'help gen-tls'

Build-ca completed successfully.
```


Build-ca completed successfully.
Notice
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /etc/openvpn/server/easy-rsa/pki/reqs/server.req
* key: /etc/openvpn/server/easy-rsa/pki/private/server.key
Using configuration from /etc/openvpn/server/easy-rsa/pki/28350007/temp.6.1
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'server'
Certificate is to be certified until Jan 12 09:09:34 2035 GMT (3650 days)
Write out database with 1 new entries
Database updated
Notice
Inline file created:
* /etc/openvpn/server/easy-rsa/pki/inline/private/server.inline
Notice
Certificate created at:
* /etc/openvpn/server/easy-rsa/pki/issued/server.crt
Notice
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /etc/openvpn/server/easy-rsa/pki/reqs/flutter_test.req
* key: /etc/openvpn/server/easy-rsa/pki/private/flutter_test.key
Using configuration from /etc/openvpn/server/easy-rsa/pki/f2298da1/temp.6.1
Check that the request matches the signature


Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'flutter_test'
Certificate is to be certified until Jan 12 09:09:34 2035 GMT (3650 days)
Write out database with 1 new entries
Database updated
Notice
Inline file created:
* /etc/openvpn/server/easy-rsa/pki/inline/private/flutter_test.inline
Notice
Certificate created at:
* /etc/openvpn/server/easy-rsa/pki/issued/flutter_test.crt
Using configuration from /etc/openvpn/server/easy-rsa/pki/9cea8ba4/temp..1
Notice
An updated CRL DER copy has been created:
* /etc/openvpn/server/easy-rsa/pki/crl.der
An updated CRL has been created:
* /etc/openvpn/server/easy-rsa/pki/crl.pem
Created symlink '/etc/systemd/system/multi-user.target.wants/openvpn-iptables.service' → '/etc/systemd/system/openvpn-iptables.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/openvpn-server@server.service' → '/usr/lib/systemd/system/openvpn-server@.service'.
Finished!
The client configuration is available in: /root/flutter_test.ovpn
New clients can be added by running this script again.
Next Step!
Enter the Device IP (Android/iOS): 172.20.10.2
Rule to redirect port 80 to 8080 has been added.
Rule to redirect port 443 to 8080 has been added.
MASQUERADE rule for IP 172.20.10.2 has been added.
Done!

Application Layer Test

Subsection: Local Storage Inspection (SharedPreferences)

Overview

This test investigates how the application stores data locally, specifically within the **`SharedPreferences`** directory used by Flutter apps. In Flutter, this data is typically stored in an XML file named `FlutterSharedPreferences.xml`, located in the app's internal storage directory. The goal is to identify whether any **sensitive information** is stored in plaintext or in an insecure manner.

This type of inspection is crucial in assessing whether the application complies with secure storage best practices and platform guidelines for data confidentiality.

Test Objective

To determine if sensitive data is stored insecurely in the `SharedPreferences` of the application, and to evaluate the effectiveness of the app's local data protection across versions.

Test Environment

- Device: **Rooted Android device**
- Access Method: **ADB Shell with root privileges**
- File Target: **`/data/data/<package_name>/shared_prefs/FlutterSharedPreferences.xml`**
- Test Date: **13 January 2025**

Tools and Setup

1. Install the target APK on a **rooted Android device**.

Use **ADB shell** to gain root access:

```
adb shell
su
cd /data/data/<package_name>
```

2. Within the app UI, click the **"Log and Save"** button to trigger preference saving.

   Navigate to:

   ```
   cd shared_prefs
   cat FlutterSharedPreferences.xml
   ```

3. Analyze the contents of the file for any **plaintext data**.

Results Summary

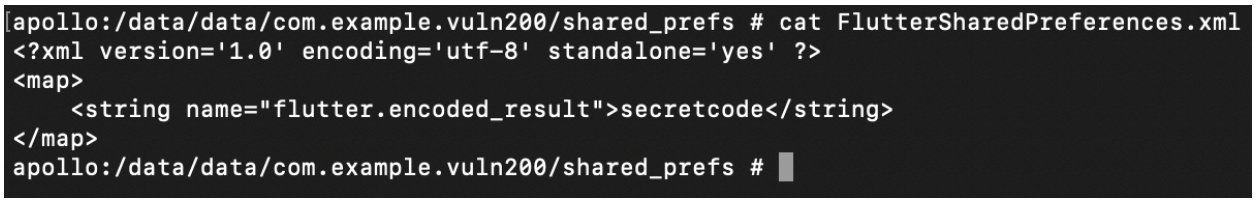
	2.0.0	2.10.0	3.7.0	3.16.6	3.27.0
	V	V	V	V	V

V = Data stored in local storage (Vulnerable)

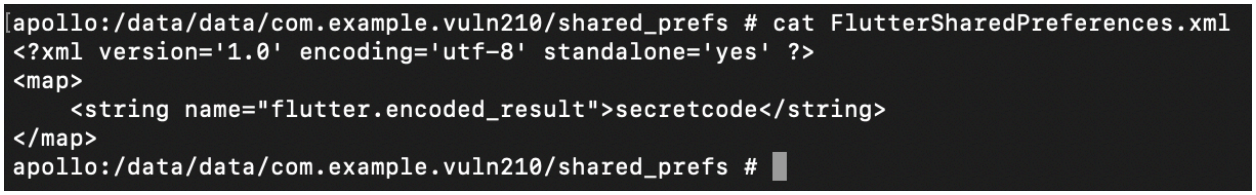
X = Data not stored in local storage (Not Vulnerable)

Screenshots

V2.0.0 (Data Stored)



V2.10.0 (Data Stored)



V3.7.0 (Data Stored)

```
apollo:/data/data/com.example.vuln3700/shared_prefs # cat FlutterSharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="flutter.encoded_result">secretcode</string>
</map>
apollo:/data/data/com.example.vuln3700/shared_prefs #
```

V3.16.6 (Data Stored)

```
apollo:/data/data/com.example.vuln31660/shared_prefs # cat FlutterSharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="flutter.encoded_result">secretcode</string>
</map>
apollo:/data/data/com.example.vuln31660/shared_prefs #
```

V3.27.0 (Data Stored)

```
apollo:/data/data/com.example.vuln32700/shared_prefs # cat FlutterSharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="flutter.encoded_result">secretcode</string>
</map>
apollo:/data/data/com.example.vuln32700/shared_prefs #
```

Subsection: Secure Log Inspection

Overview

This test evaluates the application's **logging behavior** to determine whether sensitive information is inadvertently written to system logs during runtime. Improper logging practices—such as logging authentication tokens, user input, or internal states—can pose significant security risks, especially on rooted devices or in environments where logs can be accessed by third-party apps.

This test focuses on capturing logs during critical interactions in the application (e.g., pressing the **“Log and Save”**button) to observe whether any sensitive data is exposed.

Test Objective

To identify whether the application writes sensitive data into system logs and to evaluate the application’s adherence to secure logging practices across multiple versions.

Test Environment

- Device: **Rooted Android or Emulator**
- Access Method: **ADB Logcat with process filtering**
- Tool: **Android Debug Bridge (adb)**
- Test Date: **13 January 2025**

Tools and Setup

1. Install the target APK on a test device or emulator.
2. Launch the application and identify its **Process ID (PID)**:
   `adb shell pidof <package_name>`
3. Start log monitoring for the app-specific process:
   `adb logcat --pid=<PID>`
4. In the application UI, press the **"Log and Save"** button to trigger logging.
5. Observe and record any log entries that may include sensitive or internal data.

Results Summary

	2.0.0	2.10.0	3.7.0	3.16.6	3.27.0
	V	V	V	V	V

V = Information successfully logged (Vulnerable)

X = Information unsuccessfully logged (Not Vulnerable)

Screenshots

V2.0.0 (Logged)

```
01-13 17:28:44.533 25649 25686 I flutter : Saved encoded_result: secretcode
01-13 17:28:44.999 25649 25686 I flutter : Saved encoded_result: secretcode
01-13 17:28:45.397 25649 25686 I flutter : Saved encoded_result: secretcode
```

V2.10.0 (Logged)

```
01-13 17:30:32.443 25832 25869 I flutter : Saved encoded_result: secretcode
01-13 17:30:32.633 25832 25869 I flutter : Saved encoded_result: secretcode
01-13 17:30:32.816 25832 25869 I flutter : Saved encoded_result: secretcode
```

V3.7.0 (Logged)

```
01-13 17:31:10.545 25947 25983 I flutter : Saved encoded_result: secretcode
01-13 17:31:10.720 25947 25983 I flutter : Saved encoded_result: secretcode
01-13 17:31:10.885 25947 25983 I flutter : Saved encoded_result: secretcode
```

V3.16.6 (Logged)

```
01-13 17:31:44.534 26126 26162 I flutter : Saved encoded_result: secretcode
01-13 17:31:44.728 26126 26162 I flutter : Saved encoded_result: secretcode
01-13 17:31:44.909 26126 26162 I flutter : Saved encoded_result: secretcode
```

V3.27.0 (Logged)

```
01-13 17:32:17.056 26248 26292 I flutter : Saved encoded_result: secretcode
01-13 17:32:17.229 26248 26292 I flutter : Saved encoded_result: secretcode
01-13 17:32:17.397 26248 26292 I flutter : Saved encoded_result: secretcode
```