

Contents

I.	Tools/Software used:	4
1.	MOBSF:	4
2.	Jadx-Gui:	4
3.	Ghidra:	4
4.	Androwarn:	4
5.	ApkTool:	4
6.	Genymotion:	5
8.	AAPT:	5
9.	Jarsigner:	5
10.	Adb:	5
11.	Burp suite:	5
12.	OWASP:	6
13.	Mitmproxy:	6
14.	Frida:	6
15.	Radare2:	6
II.	Application to analyze:	6
	FacilePay	6
III.	Static analysis:	6
1.	Installation of androwarn:	6
2.	Installation of MOBSF:	11
a)	Getting started:	11
b)	Running the tool:	13
c)	Result in the Ui interface:	14
d)	Signer Certificate:	15
e)	Application Permissions:	16
f)	Android API:	17
g)	Classes that use encryption methods:	18
h)	Browsable activity:	20
i)	Certificate analysis:	20
j)	Activity analysis:	21
k)	Code analysis:	21

I)	Shared library binary analysis:.....	29
m)	Niap analysis:.....	34
n)	File analysis:.....	35
o)	Server locations:.....	37
p)	Firebase database:	37
q)	Trackers:	37
r)	Activities:	38
s)	Services:.....	38
t)	The receivers and the providers:.....	38
u)	The libraries and files:	39
IV.	Dynamic analysis:	39
1.	Dynamic analysis with MobSF:	39
2.	Dynamic analysis with burp suite:.....	41
3.	Dynamic analysis with MITMproxy:	44
4.	Dynamic analysis with Frida:	49
5.	Dynamic analysis with radare2:.....	55
V.	Android Code Quality and Build Settings:	61
1.	Making Sure That the App is Properly Signed (MSTG-CODE-1):.....	61
i.	Static Analysis:.....	61
ii.	Dynamic Analysis:.....	63
2.	Testing Whether the App is Debuggable (MSTG-CODE-2):.....	63
i.	Static Analysis:.....	63
ii.	Dynamic Analysis:.....	63
3.	Testing for Debugging Code and Verbose Error Logging (MSTG-CODE-4):.....	64
i.	Static Analysis:.....	64
ii.	Dynamic Analysis	66
VI.	Graphing the apk dependencies:	66

I. Tools/Software used:

1. MOBSF:



Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis. MobSF supports mobile app binaries (APK, XAPK, IPA & APPX) along with zipped source code and provides REST APIs for seamless integration with your CI/CD or DevSecOps pipeline. The Dynamic Analyzer helps you to perform runtime security assessment and interactive instrumented testing

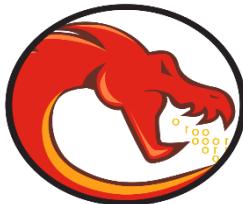
2. Jadx-Gui:



it's a Dex to Java decompiler.

Command line and GUI tools for producing Java source code from Android Dex and Apk files.

3. Ghidra:



Ghidra is a free and open-source reverse engineering tool developed by the National Security Agency of the United States. The binaries were released at RSA Conference in March 2019; the sources were published one month later on GitHub. Ghidra is seen by many security researchers as a competitor to IDA Pro.

4. Androwarn:

Androwarn is a tool whose main aim is to detect and warn the user about potential malicious behaviors developed by an Android application.

5. ApkTool:



A tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications. It also makes working with an app easier because of the project like file structure and automation of some repetitive tasks like building apk, etc. It is **NOT** intended for piracy and other non-legal uses. It could be used for localizing, adding some features or support for custom platforms, analyzing applications and much more.

6. Genymotion:



Genymotion Desktop is an Android emulator which includes a complete set of sensors and features in order to interact with a virtual Android environment. With Genymotion Desktop, you can test your Android applications on a wide range of virtual devices for development, test and demonstration purposes.

7. ApkSigner:



The apkigner tool, available in revision 24.0.3 and higher of the Android SDK Build Tools, lets you sign APKs and confirm that an APK's signature will be verified successfully on all versions of the Android platform supported by that APK

8. AAPT:

aapt stands for Android Asset Packaging Tool and is included in the tools/ directory of the SDK. This tool allows you to view, create, and update Zip-compatible archives (zip, jar, apk). It can also compile resources into binary assets.

9. Jarsigner:

Signs and verifies Java Archive (JAR) files.

10. Adb:



The Android Debug Bridge is a programming tool used for the debugging of Android-based devices. The daemon on the Android device connects with the server on the host PC over USB or TCP, which connects to the client that is used by the end-user over TCP.

11. Burp suite:



vulnerabilities.

Burp Suite is an integrated platform/ graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security

12.OWASP:



The OWASP Foundation

The Open Web Application Security Project is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security. The OWASP provides free and open resources. It is led by a non-profit called

13.Mitmproxy:



Sockets, or any other SSL/TLS-protected protocols

mitmproxy is your Swiss-army knife for debugging, testing, privacy measurements, and penetration testing. It can be used to intercept, inspect, modify and replay web traffic such as HTTP/1, HTTP/2, Web

14.Frida:



Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers.

15.Radare2:



Radare2 is a complete framework for reverse-engineering and analyzing binaries; composed of a set of small utilities that can be used together or independently from the command line

II. Application to analyze:

FacilePay



FacilePay - Payment for Stripe makes it easy for tradespeople and other small business owners to collect credit/debit card payment right on the spot with their smartphone!

Website that contains information about the app:

<https://www.facilepay.ca/>

III. Static analysis:

1. Installation of androwarn:

i. *Cloning from git:*

```
File Edit View Search Terminal Help
root@kali:~# git clone https://github.com/maaaaz/androwarn.git
Cloning into 'androwarn'...
```

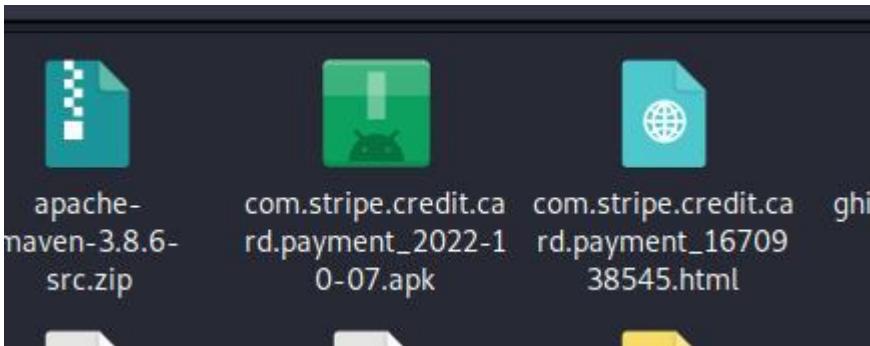
ii. Using it to analyze the apk:

```
(joxavy㉿kali)-[~/Downloads]
$ androwarn -i com.stripe.credit.card.payment_2022-10-07.apk
[+] Androwarn version 1.6

[+] Loading the APK file...
Requested API level 30 is larger than maximum we have, returning API level 28 instead.
[+] Analysis successfully completed and HTML file report available '/home/joxavy/Downloads/com.stripe.credit.card.payment_1670938545.html'

(joxavy㉿kali)-[~/Downloads]
```

The output of the analysis is saved inside an HTML file:



when clicked it displays the following result:

A screenshot of the Androwarn report interface. The top navigation bar shows tabs for WhatsApp, Static Analysis, Running MobSF, and the current tab, 'Androwarn report'. Below the tabs is a toolbar with various Kali Linux links. The main content area is titled 'Androwarn Report' and shows 'com.stripe.credit.card.payment'. On the left, a sidebar lists sections: APPLICATION INFORMATION, ANALYSIS RESULTS, APK FILE, and ANDROIDMANIFEST.XML. The right panel displays the 'Application Name' section, which shows 'FacilePay'.

As we can see, this tool displays multiple information about the apk given, including the following details:

APPLICATION INFORMATION	
Application Name	
Application Version	
Package Name	
Description	
ANALYSIS RESULTS	
Telephony Identifiers Leakage	
Device Settings Harvesting	
Connection Interfaces Exfiltration	
Telephony Services Abuse	
Suspicious Connection Establishment	
Pim Data Leakage	
Code Execution	
APK FILE	
File Name	
Fingerprint	
ANDROIDMANIFEST.XML	
Permissions	

We will now check each result one by one.

Telephony identifiers leakage:

Which are physical slots indexes or IDs: unique indexes referring to a physical SIM slot

APPLICATION INFORMATION	
Application Name	
Application Version	
Package Name	
Description	
ANALYSIS RESULTS	
Telephony Identifiers Leakage	
Device Settings Harvesting	
Connection Interfaces Exfiltration	
Telephony Services Abuse	
Suspicious Connection Establishment	
Pim Data Leakage	
Code Execution	
APK FILE	
File Name	
Fingerprint	
ANDROIDMANIFEST.XML	
Permissions	

Telephony Identifiers Leakage

This application reads the ISO country code equivalent for the SIM provider's country code
This application reads the ISO country code equivalent of the current registered operator's MCC (Mobile Country Code)
This application reads the MCC+MNC of the provider of the SIM
This application reads the device phone type value
This application reads the numeric name (MCC+MNC) of current registered operator
This application reads the operator name
This application reads the radio technology (network type) currently in use on the device for data transmission

This application leaks the ISO country code, operator name, etc.

Device settings harvesting:

Device Settings Harvesting

```
This application logs the message 'nUseNeon(); Lio:card/payment/CardsScanner;->nUseNeon(jZ)' under the tag 'card.io'  
This application logs the message ' ##### init' under the tag 'Constraints'  
This application logs the message ' Points are coincident' under the tag 'PathParser'  
This application logs the message ' no motionScene' under the tag 'MotionLayout'  
This application logs the message '' under the tag 'Destroying: LoaderManager'  
This application logs the message '' under the tag 'Resetting: LoaderManager'  
This application logs the message '' under the tag 'Starting: LoaderManager'  
This application logs the message '' under the tag 'Stopping: LoaderManager'  
This application logs the message '' under the tag 'onLoadFinished in : 3'  
This application logs the message '' under the tag 'KeyPath position '' outside of range MotionController'  
This application logs the message '' under the tag ''  
This application logs the message '' under the tag '1'  
This application logs the message '' under the tag '3'  
This application logs the message '' under the tag '4'  
This application logs the message '' under the tag 'Descriptor changed, descriptor=> MediaRouteProviderProxy'  
This application logs the message '' under the tag 'DynamicRouteDescriptors changed, descriptors=> MediaRouteProviderProxy'  
This application logs the message '' under the tag 'Service connection died MediaRouteProviderProxy'  
This application logs the message '' under the tag 'Service disconnected MediaRouteProviderProxy'  
This application logs the message '' under the tag 'Starting MediaRouteProviderProxy'  
This application logs the message '' under the tag 'Stopping MediaRouteProviderProxy'  
This application logs the message '' under the tag '***'  
This application logs the message '' under the tag 'Authentication error: ***'  
This application logs the message '' under the tag 'BasePendingResult'  
This application logs the message '' under the tag 'Clearing non-config state for FragmentManager'  
This application logs the message '' under the tag 'Commit: FragmentManager'  
This application logs the message '' under the tag 'DittoImageHeaderParser'  
This application logs the message '' under the tag 'Downsampler 1'  
This application logs the message '' under the tag 'DynamiteModule'  
This application logs the message '' under the tag 'FirebaseApp'  
This application logs the message '' under the tag 'Fragment received the following in startIntentSenderForResult() requestCode: IntentSender: fillInIntent: options: FragmentManager'  
This application logs the message '' under the tag 'FragmentManager'
```

It logs a lot of info.

Telephony services Abuse:

It can place phone calls.

Telephony Services Abuse

This application makes phone calls

Connection Interfaces Exfiltration:

Connection Interfaces Exfiltration

This application reads details about the currently active data network

This application tries to find out if the currently active data network is metered

Suspicious connection establishment:

Suspicious Connection Establishment

This application opens a Socket and connects it to the remote address ' returned no addresses for ; port is out of range' on the 'N/A' port

This application opens a Socket and connects it to the remote address " on the 'N/A' port

This application opens a Socket and connects it to the remote address 'Ljava/lang/StringBuilder;.>toString()Ljava/lang/String;' on the 'N/A' port

This application opens a Socket and connects it to the remote address 'Ljava/net/Proxy;.>type()Ljava/net/Proxy\$Type;' on the 'N/A' port

This application opens a Socket and connects it to the remote address 'timeout' on the 'N/A' port

Pim data leakage:

Pim Data Leakage

This application accesses data stored in the clipboard

Code execution:

Code Execution

This application loads a native library

This application loads a native library: 'androidndkgif'

Fingerprint:

Fingerprint

MD5: 6eea256a97972a541f42ece2b727c51d

SHA-1: 54ec3a59e2cd5f46f042bc8d1ced47beaa124291

SHA-256: f7d57fde88ad404a1480d21c519c37d2d218e41f255066cf1293e7f3b6c4c6da

Permissions:

Permissions

```
Asked: ['android.permission.ACCESS_COARSE_LOCATION',
'android.permission.ACCESS_FINE_LOCATION',
'android.permission.ACCESS_NETWORK_STATE',
'android.permission.ACCESS_WIFI_STATE',
'android.permission.BLUETOOTH',
'android.permission.CALL_PHONE',
'android.permission.CAMERA',
'android.permission.FOREGROUND_SERVICE',
'android.permission.GET_ACCOUNTS',
'android.permission.INTERNET',
'android.permission.NFC',
'android.permission.READ_CONTACTS',
'android.permission.READ_EXTERNAL_STORAGE',
'android.permission.READ_PHONE_STATE',
'android.permission.RECEIVE_BOOT_COMPLETED',
'android.permission.USE_BIOMETRIC',
'android.permission.USE_FINGERPRINT',
'android.permission.VIBRATE',
'android.permission.WAKE_LOCK',
'android.permission.WRITE_EXTERNAL_STORAGE',
'com.android.vending.BILLING',
'com.google.android.c2dm.permission.RECEIVE',
'com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE',
'com.google.android.gms.permission.AD_ID']
Implied: []
Declared: []
```

Filename:

File Name

[com.stripe.credit.card.payment_2022-10-07.apk](#)

All of this can be checked by clicking on the following file:



[com.stripe.credit.card.payment_1670938545.html](#)

2. Installation of MOBSF:

a) Getting started:

Cloning it from GitHub

Starting it up with ./setup.sh

```
(joxavy㉿kali)-[~]
└─$ git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
Cloning into 'Mobile-Security-Framework-MobSF' ...
remote: Enumerating objects: 18751, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
Receiving objects: 11% (2121/18751), 20.92 MiB | 61.00 KiB/s
^C

(joxavy㉿kali)-[~]
└─$ git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
Cloning into 'Mobile-Security-Framework-MobSF' ...
remote: Enumerating objects: 18751, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 18751 (delta 0), reused 0 (delta 0), pack-reused 18749
Receiving objects: 100% (18751/18751), 1.23 GiB | 3.05 MiB/s, done.
Resolving deltas: 100% (9321/9321), done.
Updating files: 100% (458/458), done.

(joxavy㉿kali)-[~]
└─$ cd Mobile-Security-Framework-MobSF
(joxavy㉿kali)-[~/Mobile-Security-Framework-MobSF]
└─$ ./setup.sh
[INSTALL] Found Python 3.10.8
pip 22.3 from /usr/lib/python3/dist-packages/pip (python 3.10)
[INSTALL] Found pip
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (22.3)
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  WARNING: The scripts pip, pip3 and pip3.10 are installed in '/home/joxavy/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-22.3.1
[INSTALL] Using python virtualenv
[INSTALL] Activating virtualenv
Requirement already satisfied: pip in ./venv/lib/python3.10/site-packages (22.3)
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3
    Uninstalling pip-22.3:
      Successfully uninstalled pip-22.3
Successfully installed pip-22.3.1
[INSTALL] Installing Requirements
Collecting wheel
  Downloading wheel-0.38.4-py3-none-any.whl (36 kB)
Installing collected packages: wheel
Successfully installed wheel-0.38.4
Ignoring waitress: markers 'platform_system == "Windows"' don't match your environment
Collecting Django≥3.1.5
  Downloading Django-4.1.4-py3-none-any.whl (8.1 MB)
Collecting lxml≥4.6.2
  Downloading lxml-4.9.1-cp310-cp310-manylinux_2_17_x86_64_manylinux2014_x86_64_manylinux_2_24_x86_64.whl (6.9 MB)
)
Collecting rsa≥4.7

```

Finished installation:

```
[INFO] 13/Dec/2022 13:21:58 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
No changes detected in app 'StaticAnalyzer'.
[INFO] 13/Dec/2022 13:21:58 - Checking for Update.
[INFO] 13/Dec/2022 13:21:58 - No updates available.
[INFO] 13/Dec/2022 13:22:00 -
[INFO] 13/Dec/2022 13:22:00 - Mobile Security Framework v3.6.2 Beta
REST API Key: 3260d526ec7ba2a061485447cb06959198191bd21d808eca8ebb9a7b6001ebc5
[INFO] 13/Dec/2022 13:22:00 - OS: Linux
[INFO] 13/Dec/2022 13:22:00 - Platform: Linux-5.14.0-kali4-amd64-x86_64-with-glibc2.36
[INFO] 13/Dec/2022 13:22:00 - Dist: kali 2021.4 kali-rolling
[INFO] 13/Dec/2022 13:22:00 - MobSF Basic Environment Check
[WARNING] 13/Dec/2022 13:22:00 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
Operations to perform:
  Apply all migrations: StaticAnalyzer, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
[INFO] 13/Dec/2022 13:22:01 - Checking for Update.
[INFO] 13/Dec/2022 13:22:01 - No updates available.
wkhtmltopdf 0.12.6
[INSTALL] Installation Complete

(joxavy㉿kali)-[~/Mobile-Security-Framework-MobSF]
└─$
```

b) Running the tool:

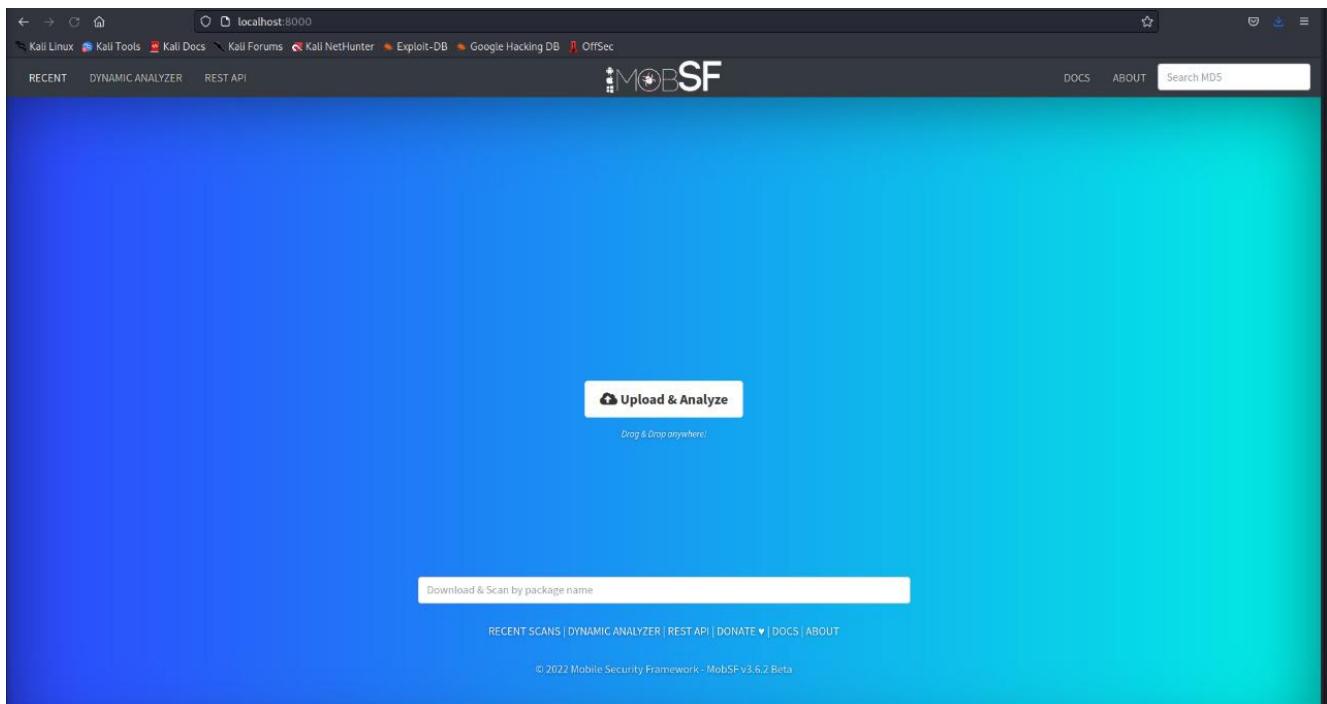
```
[joxavy@kali:~/Mobile-Security-Framework-MobSF]$ ./run.sh 127.0.0.1:8000
[2022-12-13 08:24:50 -0500] [30323] [INFO] Starting gunicorn 20.1.0
[2022-12-13 08:24:50 -0500] [30323] [INFO] Listening at: http://127.0.0.1:8000 (30)
[2022-12-13 08:24:50 -0500] [30323] [INFO] Using worker: gthread
[2022-12-13 08:24:50 -0500] [30324] [INFO] Booting worker with pid: 30324
[INFO] 13/Dec/2022 13:25:51 -
```



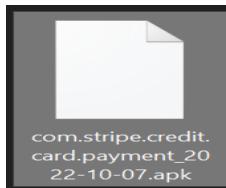
```
[INFO] 13/Dec/2022 13:25:51 - Mobile Security Framework v3.6.2 Beta
REST API Key: 3260d526ec7ba2a061485447cb0695919819bd21d808eca8eb9a7b6001ebc5
[INFO] 13/Dec/2022 13:25:51 - OS: Linux
[INFO] 13/Dec/2022 13:25:51 - Platform: Linux-5.14.0-kali4-amd64-x86_64-with-glibc
[INFO] 13/Dec/2022 13:25:51 - Dist: kali 2021.4 kali-rolling
[INFO] 13/Dec/2022 13:25:51 - MobSF Basic Environment Check
[WARNING] 13/Dec/2022 13:25:51 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing analysis.
[INFO] 13/Dec/2022 13:25:52 - Checking for Update.
[INFO] 13/Dec/2022 13:26:03 - No updates available.
[INFO] 13/Dec/2022 13:26:24 - MIME Type: application/vnd.android.package-archive File
[INFO] 13/Dec/2022 13:26:24 - Performing Static Analysis of Android APK
[INFO] 13/Dec/2022 13:26:24 - Scan Hash: 6eaa256a97972a541f42ee2b727e51d
[INFO] 13/Dec/2022 13:26:24 - Starting Analysis on: com.stripe.credit.card.payment
[INFO] 13/Dec/2022 13:26:24 - Generating Hashes
[INFO] 13/Dec/2022 13:26:24 - Unzipping
[INFO] 13/Dec/2022 13:26:25 - APK Extracted
[INFO] 13/Dec/2022 13:26:25 - Getting Hardcoded Certificates/Keystores
[INFO] 13/Dec/2022 13:26:25 - Getting AndroidManifest.xml from APK
[INFO] 13/Dec/2022 13:26:25 - Converting AXML to XML
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] 13/Dec/2022 13:26:35 - Parsing AndroidManifest.xml
[INFO] 13/Dec/2022 13:26:36 - Fetching icon path
[INFO] 13/Dec/2022 13:26:37 - Extracting Manifest Data
[INFO] 13/Dec/2022 13:26:37 - Fetching Details from Play Store: com.stripe.credit...
[INFO] 13/Dec/2022 13:26:44 - Manifest Analysis Started
[INFO] 13/Dec/2022 13:26:44 - Binary Analysis Started
[INFO] 13/Dec/2022 13:26:44 - Analyzing lib/x86_64/libcardioRecognizer_tegra2.so
[INFO] 13/Dec/2022 13:26:44 - Analyzing lib/x86_64/libcardioRecognizer.so
```

Launching the UI interface with localhost:8000 on any web browser:

We get the following interface:



After dropping the apk, that is, the following apk:



We wait a few minutes then get the following result:

As we can see on the command line while we wait, the apk file is being translated to java, dex to smali, etc.

```
[INFO] Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] 13/Dec/2022 13:27:20 - APK → JAVA
[INFO] 13/Dec/2022 13:27:20 - Decompiling to Java with jadx
[INFO] 13/Dec/2022 13:28:45 - DEX → SMALI
[INFO] 13/Dec/2022 13:28:45 - Converting classes2.dex to Smali Code
[INFO] 13/Dec/2022 13:28:45 - Converting classes3.dex to Smali Code
[INFO] 13/Dec/2022 13:28:45 - Converting classes.dex to Smali Code
[INFO] 13/Dec/2022 13:28:45 - Code Analysis Started on - java_source
[INFO] 13/Dec/2022 13:32:44 - Running NIAP Analyzer
[INFO] 13/Dec/2022 13:34:20 - Finished Code Analysis, Email and URL Extraction
[INFO] 13/Dec/2022 13:34:20 - Extracting Strings from APK
[INFO] 13/Dec/2022 13:34:21 - Detecting Firebase URL(s)
[INFO] 13/Dec/2022 13:34:26 - Performing Malware Check on extracted Domains
[INFO] 13/Dec/2022 13:34:46 - Maltrail Database is outdated!
[INFO] 13/Dec/2022 13:34:47 - Updating Maltrail Database
[INFO] 13/Dec/2022 13:35:15 - Connecting to Database
[INFO] 13/Dec/2022 13:35:15 - Saving to Database
[INFO] 13/Dec/2022 13:35:16 - Scan Hash: 6eea256a97972a541f42ece2b727c51d
[INFO] 13/Dec/2022 13:35:16 - Starting Analysis on: com.stripe.credit.card.payment_2022-10-07.apk
[INFO] 13/Dec/2022 13:35:16 - Analysis is already Done. Fetching data from the DB ... [REDACTED]
```

c) Result in the Ui interface:

The screenshot shows the MobSF static analyzer interface. On the left, there's a sidebar with navigation links like 'Information', 'Scan Options', 'Signer Certificate', 'Permissions', 'Android API', 'Browsable Activities', 'Security Analysis', 'Malware Analysis', 'Reconnaissance', 'Components', 'PDF Report', 'Print Report', and 'Start Dynamic Analysis'. The main content area is divided into several sections: 'APP SCORES' (Security Score: 48/100, Trackers: 2/21), 'FILE INFORMATION' (File Name: com.stripe.credit.card.payment_2022-10-07.apk, Size: 32.89MB, MD5: 6eea256a97972a541f42ece2b727c51d, SHA1: 54ee3a59e2cd5f46042bcd1ced47beaa124291, SHA256: f7d57fdde88ad404a1480d21c519c37d2d218e41255066cd1293e7f3b6c4c6da), 'PLAYSTORE INFORMATION' (Title: Stripe Payments App: FacilePay, Score: 4.5384617, Installs: 100,000+, Price: 0, Category: Finance, Play Store URL: com.stripe.credit.card.payment, Developer: Vbridge Technologies Inc., Developer ID: Vbridge+Technologies+Inc, Developer Address: 2 County Court Blvd., Suite 400, Office Number: 429 Brampton, L6W 3W8, Ontario, Canada, Developer Website: https://www.facilepay.ca/, Developer Email: support@facilepay.ca, Release Date: Feb 16, 2018, Privacy Policy: Privacy link, Description: FacilePay - Payment for Stripe makes it easy for tradespeople and other small business owners to collect credit/debit card payment right on the spot with their smartphone! Connect a Stripe account to your mobile device and get started charging customers in less than 30 seconds. No monthly fees. No minimums. No need for any additional work to get up and running. Use it as a standalone or supplemental tool. In an increasingly cash-less world, FacilePay - Payment for Stripe is a great resource for those who are keeping up with the mobile payment advancements in today's society. Credit card processing on mobile has never been easier.), 'APP INFORMATION' (App Name: FacilePay, Package Name: com.stripe.credit.card.payment, Main Activity: com.easypay.activity.SplashActivity, Target SDK: 30, Min SDK: 21, Max SDK: 30, Android Version Name: 7.7.16, Android Version Code: 108), and 'INSTANT PAYMENT' (with a 'TRUSTED' badge: Businesses from all around the world trust FacilePay to handle millions of dollars in transactions).

First thing we get is general information about the app, the encryption methods, package name, android version, etc.

On the second half, we get play store information, app title, installs, release date, etc.

d) Signer Certificate:

Checking the apk signature:

```
● SIGNER CERTIFICATE

APK is signed
v1 signature: True
v2 signature: True
v3 signature: True
Found 1 unique certificates
Subject: C=US, ST=California, L=Mountain View, O=Google Inc., OU=Android, CN=Android
Signature Algorithm: rsassa_pks1v15
Valid From: 2018-02-16 09:41:47+00:00
Valid To: 2048-02-16 09:41:47+00:00
Issuer: C=US, ST=California, L=Mountain View, O=Google Inc., OU=Android, CN=Android
Serial Number: 0x6c3469aab12528075a7d18f47a724f2598269e1b
Hash Algorithm: sha256
md5: 2597b95812e29712dd13bia628f15c3c
sha1: 21617741684ecbeea474eaa49f93949fd99aeab8f
sha256: 308f388289982fbf0cb349ccbf6c3fb2a4075488b2c3cc99fb39ea18e0b9ce5
sha512: 89e16bb1ca557a6125ec04f4d1aa6f8c4b3614dfc002aec8460b41b852fb8711ciae32b81fcb3d79d03f14eb8444d2c9629911cd402f8b396642767bd70a73e29
PublicKey Algorithm: rsa
Bit Size: 4096
Fingerprint: a64d8fbccca81c23193a4f4b7911a2ea9c6a72fadcf088067566ffff57ea7e0d
```

≡ APPLICATION PERMISSIONS

As we can see, the algorithm used to sign the apk is “**rsassa_pks1v15**” which is a signature algorithm is the most widely used digital signature scheme in practice. Its two main strengths are its extreme simplicity, which makes it very easy to implement, and that verification of signatures is significantly faster than for DSA or ECDSA.

The hash algorithm used is: SHA256

SHA-256, which stands for secure hash algorithm 256, is a cryptographic hashing algorithm (or function) that's used for message, file, and data integrity verification

e) Application Permissions:

APPLICATION PERMISSIONS				Search: <input type="text"/>
PERMISSION	STATUS	INFO	DESCRIPTION	
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.	
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.	
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.	
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.	
android.permission.BLUETOOTH	normal	create Bluetooth connections	Allows applications to connect to paired bluetooth devices.	
android.permission.CALL_PHONE	dangerous	directly call phone numbers	Allows the application to call phone numbers without your intervention. Malicious applications may cause unexpected calls on your phone bill. Note that this does not allow the application to call emergency numbers.	
android.permission.CAMERA	dangerous	take pictures and videos	Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.	
android.permission.FOREGROUND_SERVICE	normal		Allows a regular application to use Service.startForeground.	
android.permission.GET_ACCOUNTS	dangerous	list accounts	Allows access to the list of accounts in the Accounts Service.	
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.	

Showing 1 to 10 of 24 entries

Previous 1 2 3 Next

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.NFC	normal	control Near-Field Communication	Allows a reader.
android.permission.READ_CONTACTS	dangerous	read contact data	Allows a application
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows a
android.permission.READ_PHONE_STATE	dangerous	read phone state and identity	Allows t permission, t
android.permission.RECEIVE_BOOT_COMPLETED	normal	automatically start at boot	Allows a longer t running
android.permission.USE_BIOMETRIC	normal		Allows a
android.permission.USE_FINGERPRINT	normal	allow use of fingerprint	This cor
android.permission.VIBRATE	normal	control vibrator	Allows t
android.permission.WAKE_LOCK	normal	prevent phone from sleeping	Allows a
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows a

PERMISSION	STATUS	IN
com.android.vending.BILLING	unknown	Un
com.google.android.c2dm.permission.RECEIVE	signature	C2
com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE	unknown	Un
com.google.android.gms.permission.AD_ID	unknown	Un

As we can see in this screen, the tool shows the different permissions included in the app, some of which are dangerous permissions such as the one that gives access to the coarse location, because once it is allowed, it will let the app determine the phone location, even if its approximate, and also the camera permission that allows the application to take pictures and videos, that the hackers can later use against you if it's a malicious app.

f) Android API:

API	FILES
Android Notifications	com/easypay/stripe/notification/a.java
Base64 Decode	com/airbnb/lottie/t/b.java com/bumptech/glide/load/n/e.java com/easypay/stripe/util/g.java com/stripe/android/stripe3ds2/security/PublicKeyFactory.java com/stripe/android/stripe3ds2/transactions/ChallengeResponseData.java j/b/b/a/i/a0/j/r0.java k/c.java
Base64 Encode	com/easypay/stripe/util/g.java h/i/l/d.java j/b/b/a/i/a0/j/r0.java j/b/b/a/i/p.java
Certificate Handling	io/grpc/k1/e.java io/grpc/k1/h.java io/grpc/k1/m.java
Crypto	com/easypay/stripe/activity/LoginFingerPrintActivity.java com/easypay/stripe/activity/WebConnectActivity.java com/easypay/stripe/m/a.java com/easypay/stripe/n/a.java com/easypay/stripe/setup/b/f2.java com/easypay/stripe/util/g.java com/easypay/stripe/util/r.java com/nimbusds/jose/jwk/k.java com/nimbusds/jose/u/a.java

It uses base4 decoding, that is used to encode binary data as printable text. This allows you to transport binary over protocols or mediums that cannot handle binary data formats and require simple text.

```

public Bitmap a(String str) {
    g gVar = this.d.get(str);
    if (gVar == null) {
        return null;
    }
    Bitmap a = gVar.a();
    if (a != null) {
        return a;
    }
    com.airbnb.lottie.b bVar = this.c;
    if (bVar != null) {
        Bitmap a2 = bVar.gVar();
        if (a2 != null) {
            c(str, a2);
        }
    }
    return a2;
}
String b = gVar.b();
BitmapOptions options = options == null ? BitmapFactory.Options() : options;
options.inScaled = true;
options.inDensity = 160;
if (b.startsWith("data:") && b.indexOf("base64") > 0) {
    try {
        byte[] decode = Base64.decode(b.substring(b.indexOf("44") + 1), 0);
        Bitmap decodeFromByteArray = BitmapFactory.decodeByteArray(decode, 0, decode.length, options);
        c(str, decodeFromByteArray);
        return decodeFromByteArray;
    } catch (IllegalArgumentException e2) {
        d.c("data URL did not have correct base64 format.", e2);
        return null;
    }
}
try {
    if (!TextUtils.isEmpty(this.b)) {
        AssetManager assets = this.a.getAssets();
        Bitmap k2 = h(BitmapFactory.decodeStream(assets.open(this.b + b), null, options), gVar.e(), gVar.c());
        c(str, k2);
        return k2;
    }
} catch (IllegalStateException e3) {
    d.c("You must set an images folder before loading an image. Set it with LottieComposition.setImagesFolder or LottieDrawable#setImagesFolder");
} catch (IOException e3) {
    d.c("Unable to open asset.", e3);
}

```

As we can see here in this class, the base64 is used.

g) Classes that use encryption methods:

Crypto

```
com/easypay/stripe/activity/LoginFingerPrintActivity.java  
com/easypay/stripe/activity/WebConnectActivity.java  
com/easypay/stripe/m/a.java  
com/easypay/stripe/n/a.java  
com/easypay/stripe/setup/b/f2.java  
com/easypay/stripe/util/g.java  
com/easypay/stripe/util/r.java  
com/nimbusds/jose/jwk/k.java  
com/nimbusds/jose/u/a.java  
com/nimbusds/jose/u/b.java  
com/nimbusds/jose/u/d.java  
com/nimbusds/jose/u/e.java
```

For example, the LoginFingerPrintActivity:

LoginFingerPrintActivity.java

```
1. package com.easypay.stripe.activity;
2.
3. import android.app.KeyguardManager;
4. import android.hardware.fingerprint.FingerprintManager;
5. import android.os.Bundle;
6. import android.security.keystore.KeyGenParameterSpec;
7. import android.security.keystore.KeyPermanentlyInvalidatedException;
8. import com.stripe.credit.card.payment.R;
9. import com.stripe.data.database.ActivityLoginFingerPrintBinding;
10. import java.io.IOException;
11. import java.security.InvalidAlgorithmParameterException;
12. import java.security.KeyStoreException;
13. import java.security.KeyStore;
14. import java.security.KeyStoreException;
15. import java.security.NoSuchAlgorithmException;
16. import java.security.NoSuchProviderException;
17. import java.security.UnrecoverableKeyException;
18. import java.security.cert.CertificateException;
19. import javax.crypto.Cipher;
20. import javax.crypto.KeyGenerator;
21. import javax.crypto.NoSuchPaddingException;
22. import javax.crypto.SecretKey;
23. /* loaded from: classes.dex */
24. public class LoginFingerPrintActivity extends com.easypay.stripe.g {
25.
26.     /* renamed from: r reason: collision with root package name */
27.     private KeyStore f1931j;
28.
29.     /* renamed from: k reason: collision with root package name */
30.     private Cipher f1932k;
31.
32.     public boolean l() {
33.         try {
34.             this.f1932k = Cipher.getInstance("AES/CBC/PKCS5Padding");
35.             try {
36.                 this.f1931j.load(null);
37.                 this.f1932k.init(1, (SecretKey) this.f1931j.getKey("EasyPay", null));
38.                 return true;
39.             } catch (KeyPermanentlyInvalidatedException unused) {
40.                 return false;
41.             } catch (IOException e) {
42.                 e = e;
43.             }
44.         } catch (NoSuchAlgorithmException unused2) {
45.             return false;
46.         }
47.     }
48. }
```

The methods `crypto.cipher`, `crypto.keyGenerator`, `crypto.SecretKey`, are used.

```

public class LoginFingerPrintActivity extends com.easypay.stripe.g {
    /* renamed from: L reason: collision with root package name */
    private KeyStore f1931j;
    /* renamed from: L reason: collision with root package name */
    private Cipher f1932k;

    public boolean H() {
        try {
            this.f1932k = Cipher.getInstance("AES/CBC/PKCS7Padding");
            try {
                this.f1932k.load(null);
                this.f1932k.init(1, (SecretKey) this.f1931j.getKey("EasyPay", null));
                return true;
            } catch (KeyPermanentlyInvalidatedException unused) {
                return false;
            } catch (IOException e) {
                e = e;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (InvalidKeyException e2) {
                e = e2;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (KeyStoreException e3) {
                e = e3;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (NoSuchAlgorithmException e4) {
                e = e4;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (UnrecoverableKeyException e5) {
                e = e5;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (CertificateException e6) {
                e = e6;
                throw new RuntimeException("Failed to init Cipher", e);
            }
        } catch (NoSuchAlgorithmException | NoSuchPaddingException e7) {
            throw new RuntimeException("Failed to get Cipher", e7);
        }
    }

    protected void T() {
}

```

They called cipher named “f1932k” and used the encryption methods AES/CBC/PCKS7PADDING to encrypt it, using the try-catch statement that consists of a try block followed by one or more catch clauses, which specify handlers for different exceptions. When an exception is thrown such as KeyPermanentlyinvalidated exception, nosuchalgorithm exception, etc. and the common language runtime (CLR) looks for the catch statement that handles this exception.

m.java

```

1. package io.grpc.k1;
2.
3. import java.net.Socket;
4. import java.util.Arrays;
5. import java.util.Collections;
6. import java.util.List;
7. import javax.net.ssl.HostnameVerifier;
8. import javax.net.ssl.SSLPeerUnverifiedException;
9. import javax.net.ssl.SSLSocket;
10. import javax.net.ssl.SSLSocketFactory;
11. /* compiled from: OkHttpTlsUpgrader.java */
12. /* loaded from: Classes3.dex */
13. final class {
14.     static final List<io.grpc.k1.r.g> a = Collections.unmodifiableList(Arrays.asList(io.grpc.k1.r.g.HTTP_2));
15.
16.     static String a(String str) {
17.         return (str.startsWith("(") && str.endsWith(")")) ? str.substring(1, str.length() - 1) : str;
18.     }
19.
20.     public static SSLSocket b(SSLSocketFactory sSSLSocketFactory, HostnameVerifier hostnameVerifier, Socket socket, String str, int i2, io.grpc.k1.r.b bVar) {
21.         com.google.common.base.n.o<SSLSocket> sSSLSocketFactory;
22.         com.google.common.base.n.o<Socket> "socket";
23.         com.google.common.base.n.o<bVar> "spec";
24.         SSLSocket sSSLSocket = (SSLSocket) sSSLSocketFactory.createSocket(socket, str, i2, true);
25.         bVar.c(sSSLSocket, false);
26.         String str2 = sSSLSocket.getpeername();
27.         boolean contains = str2.contains(io.grpc.k1.r.g.h2);
28.         com.google.common.base.n.w<contains> "Only " + a + " are supported, but negotiated protocol is " + str, h2);
29.         if (hostnameVerifier == null) {
30.             hostnameVerifier = io.grpc.k1.r.d.a;
31.         }
32.         if (hostnameVerifier.verify(a(str), sSSLSocket.getSession())) {
33.             return sSSLSocket;
34.         }
35.         throw new SSLPeerUnverifiedException("Cannot verify hostname: " + str);
36.     }
37. }

```

A different class is using another security method that is SSLSocketFactory, it acts as a factory for creating secure sockets.

This class is an abstract subclass of javax. net. SocketFactory.

Secure socket factories encapsulate the details of creating and initially configuring secure sockets.

[Https/https connections used in other classes included in the app:](https://www.google.com)

HTTP Connection	<code>com/airbnb/lottie/v/c.java</code> <code>com/bumptech/glide/load/m/j.java</code> <code>com/stripe/android/stripe3ds2/transaction/StripeHttpClient\$doGetRequest\$2.java</code> <code>com/stripe/android/stripe3ds2/transaction/StripeHttpClient.java</code> <code>com/stripe/net/HttpURLConnectionClient.java</code> <code>com/stripe/net/LiveStripeResponseGetter.java</code> <code>j/b/b/b/a/d.java</code>
HTTPS Connection	<code>com/stripe/android/networking/ConnectionFactory.java</code> <code>com/stripe/android/networking/StripeConnection.java</code> <code>com/stripe/android/stripe3ds2/observability/DefaultErrorReporter.java</code>
Inter Process Communication	<code>com/bumptech/glide/m/e.java</code> <code>com/easypay/stripe/activity/AddCustomerActivity.java</code> <code>com/easypay/stripe/activity/AddDiscountActivity.java</code> <code>com/easypay/stripe/activity/AddInvoiceItemActivity.java</code> <code>com/easypay/stripe/activity/AddTaxActivity.java</code>

h) browsable activity:

BROWSABLE ACTIVITIES	
ACTIVITY	INTENT
<code>com.easypay.stripe.activity.MainActivity</code>	<code>Schemes: http://,</code> <code>Hosts: payment_intent_return,</code>
<code>com.easypay.stripe.activity.RecurringPaymentCreateActivity</code>	<code>Schemes: http://,</code> <code>Hosts: payment_intent_return,</code>
<code>com.easypay.stripe.activity.SplashActivity</code>	<code>Schemes: facilepay://,</code> <code>Hosts: navigation,</code>
<code>com.stripe.android.payments.StripeBrowserLauncherActivity</code>	<code>Schemes: stripesdk://,</code> <code>Hosts: payment_return_url,</code> <code>Paths: /com.stripe.credit.card.payment,</code>

Browsable activities are required in order for the intent filter to be accessible from a web browser. Without it, clicking a link in a browser cannot resolve to the app.

i) Certificate analysis:

CERTIFICATE ANALYSIS		
TITLE	SEVERITY	DESCRIPTION
Application vulnerable to Janus Vulnerability	warning	Application is signed with v1 signature scheme, making it vulnerable to Janus vulnerability on Android 5.0-8.0, if signed only with v1 signature scheme. Applications running on Android 5.0-7.0 signed with v1, and v2/v3 scheme is also vulnerable.
Signed Application	info	Application is signed with a code signing certificate

Application is signed with v1 signature scheme which makes its vulnerable to Janus vulnerability which comes from the possibility to add extra bytes to APK files and to DEX files.

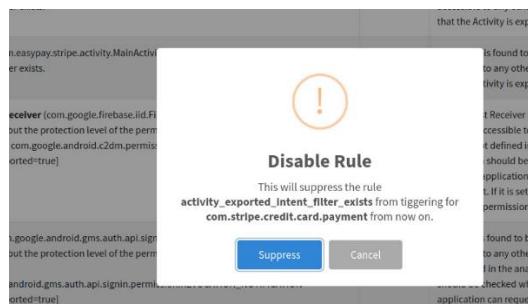
j) Activity analysis:

NO	ISSUE	SEVERITY	DESCRIPTION	OPTIONS
1	Clear text traffic is Enabled For App [android:usesCleartextTraffic=true]	high	The app intends to use cleartext network traffic, such as cleartext HTTP, FTP stacks, DownloadManager, and MediaPlayer. The default value for apps that target API level 27 or lower is "true". Apps that target API level 28 or higher default to "false". The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protections against tampering: a network attacker can eavesdrop on transmitted data and also modify it without being detected.	
2	Activity (com.easypay.stripe.activity.RecurringPaymentCreateActivity) is not Protected. An intent-filter exists.	warning	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.	
3	Activity (com.easypay.stripe.activity.MainActivity) is not Protected. An intent-filter exists.	warning	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.	
4	Broadcast Receiver (com.google.firebaseio.iid.FirebaseInstanceIdReceiver) is Protected by a permission, but the protection level of the permission should be checked. Permissions: com.google.android.c2dm.permission.SEND [android:exported=true]	warning	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analysed application. As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission.	
5	Service (com.google.android.gms.auth.api.signin.RevocationBoundService) is Protected by a permission, but the protection level of the permission should be checked. Permissions: com.google.android.gms.auth.api.signin.permission.REVOCATION_NOTIFICATION [android:exported=true]	warning	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analysed application. As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission.	

Application uses cleartext network traffic for Http, ftp stacks which poses danger to the app, as well as having many activities not completely protected, therefore many warnings are shown.

2	Activity (com.easypay.stripe.activity.RecurringPaymentCreateActivity) is not Protected. An intent-filter exists.	warning	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Activity is explicitly exported.	
---	--	---------	---	--

As we can see here, there is an option that lets you see what rule needs to be deleted from the activity in order to eliminate this threat.



When you click on it, you can disable that rule and fix it without owning the source code and editing directly from android studio.

k) Code analysis:

CODE ANALYSIS					Search: <input type="text"/>
NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	The App logs information. Sensitive information should never be logged.	info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3	com/airbnb/lottie/LottieAnimationView.java com/airbnb/lottie/x/c.java com/bumptech/glide/b.java com/bumptech/glide/l/d.java com/bumptech/glide/l/e.java com/bumptech/glide/load/engine/GlideException.java com/bumptech/glide/load/engine/a0/e.java com/bumptech/glide/load/engine/a0/i.java com/bumptech/glide/load/engine/b0/a.java com/bumptech/glide/load/engine/b0/b.java com/bumptech/glide/load/engine/h.java com/bumptech/glide/load/engine/i.java com/bumptech/glide/load/engine/k.java com/bumptech/glide/load/engine/y.java	

The application logs important information in many classes, as shown in the files part.

As we can see in this class:

```
1. public void p(String str, String str2) {
2.     o(com.airbnb.lottie.w.k0.c.o(r.p.k(new ByteArrayInputStream(str.getBytes())))), str2);
3. }
4.
5. public void setAnimation(int i2) {
6.     this.f1400h = i2;
7.     this.f1399g = null;
8.     setCompositionTask(e.k(getContext(), i2));
9. }
10.
11. @Deprecated
12. public void setAnimationFromJson(String str) {
13.     p(str, null);
14. }
15.
16. public void setAnimationFromUrl(String str) {
17.     setCompositionTask(e.m(getContext(), str));
18. }
19.
20. public void setComposition(d dVar) {
21.     if (com.airbnb.lottie.c.a) {
22.         String str = t;
23.         Log.v(str, "Set Composition \n" + dVar);
24.     }
25.     this.e.setCallback(this);
26.     this.f1407q = dVar;
27.     boolean H = this.e.H(dVar);
28.     i();
29.     if (getDrawable() != this.e || H) {
30.         setImageDrawable(null);
31.         setImageDrawable(this.e);
32.         onVisibilityChanged(this, getVisibility());
33.         requestLayout();
34.         for (j jVar : this.f1405n) {
35.             jVar.adVar();
36.         }
37.     }
38. }
39.
40. public void a(j jVar) {
41.     if (jVar != null) {
42.         jVar.adVar();
43.     }
44. }
45.
46. public void a(j jVar) {
47.     if (jVar != null) {
48.         jVar.adVar();
49.     }
50. }
51.
52. public void a(j jVar) {
53.     if (jVar != null) {
54.         jVar.adVar();
55.     }
56. }
57.
58. public void a(j jVar) {
59.     if (jVar != null) {
60.         jVar.adVar();
61.     }
62. }
63.
64. public void a(j jVar) {
65.     if (jVar != null) {
66.         jVar.adVar();
67.     }
68. }
69.
70. public void a(j jVar) {
71.     if (jVar != null) {
72.         jVar.adVar();
73.     }
74. }
75.
76. public void a(j jVar) {
77.     if (jVar != null) {
78.         jVar.adVar();
79.     }
80. }
81.
82. public void a(j jVar) {
83.     if (jVar != null) {
84.         jVar.adVar();
85.     }
86. }
87.
88. public void a(j jVar) {
89.     if (jVar != null) {
90.         jVar.adVar();
91.     }
92. }
93.
94. public void a(j jVar) {
95.     if (jVar != null) {
96.         jVar.adVar();
97.     }
98. }
99.
100. public void a(j jVar) {
101.     if (jVar != null) {
102.         jVar.adVar();
103.     }
104. }
105.
106. public void a(j jVar) {
107.     if (jVar != null) {
108.         jVar.adVar();
109.     }
110. }
111.
112. public void a(j jVar) {
113.     if (jVar != null) {
114.         jVar.adVar();
115.     }
116. }
117.
118. public void a(j jVar) {
119.     if (jVar != null) {
120.         jVar.adVar();
121.     }
122. }
123.
124. public void a(j jVar) {
125.     if (jVar != null) {
126.         jVar.adVar();
127.     }
128. }
129.
130. public void a(j jVar) {
131.     if (jVar != null) {
132.         jVar.adVar();
133.     }
134. }
135.
136. public void a(j jVar) {
137.     if (jVar != null) {
138.         jVar.adVar();
139.     }
140. }
141.
142. public void a(j jVar) {
143.     if (jVar != null) {
144.         jVar.adVar();
145.     }
146. }
147.
148. public void a(j jVar) {
149.     if (jVar != null) {
150.         jVar.adVar();
151.     }
152. }
153.
154. public void a(j jVar) {
155.     if (jVar != null) {
156.         jVar.adVar();
157.     }
158. }
159.
160. public void a(j jVar) {
161.     if (jVar != null) {
162.         jVar.adVar();
163.     }
164. }
165.
166. public void a(j jVar) {
167.     if (jVar != null) {
168.         jVar.adVar();
169.     }
170. }
171.
172. public void a(j jVar) {
173.     if (jVar != null) {
174.         jVar.adVar();
175.     }
176. }
177.
178. public void a(j jVar) {
179.     if (jVar != null) {
180.         jVar.adVar();
181.     }
182. }
183.
184. public void a(j jVar) {
185.     if (jVar != null) {
186.         jVar.adVar();
187.     }
188. }
189.
190. public void a(j jVar) {
191.     if (jVar != null) {
192.         jVar.adVar();
193.     }
194. }
195.
196. public void a(j jVar) {
197.     if (jVar != null) {
198.         jVar.adVar();
199.     }
200. }
201.
202. public void a(j jVar) {
203.     if (jVar != null) {
204.         jVar.adVar();
205.     }
206. }
207.
208. public void a(j jVar) {
209.     if (jVar != null) {
210.         jVar.adVar();
211.     }
212. }
213.
214. public void a(j jVar) {
215.     if (jVar != null) {
216.         jVar.adVar();
217.     }
218. }
219.
220. public void a(j jVar) {
221.     if (jVar != null) {
222.         jVar.adVar();
223.     }
224. }
225.
226. public void a(j jVar) {
227.     if (jVar != null) {
228.         jVar.adVar();
229.     }
230. }
231.
232. public void a(j jVar) {
233.     if (jVar != null) {
234.         jVar.adVar();
235.     }
236. }
237.
238. public void a(j jVar) {
239.     if (jVar != null) {
240.         jVar.adVar();
241.     }
242. }
243.
244. public void a(j jVar) {
245.     if (jVar != null) {
246.         jVar.adVar();
247.     }
248. }
249.
250. public void a(j jVar) {
251.     if (jVar != null) {
252.         jVar.adVar();
253.     }
254. }
255.
256. public void a(j jVar) {
257.     if (jVar != null) {
258.         jVar.adVar();
259.     }
260. }
261.
262. public void a(j jVar) {
263.     if (jVar != null) {
264.         jVar.adVar();
265.     }
266. }
267.
268. public void a(j jVar) {
269.     if (jVar != null) {
270.         jVar.adVar();
271.     }
272. }
273.
274. public void a(j jVar) {
275.     if (jVar != null) {
276.         jVar.adVar();
277.     }
278. }
279.
280. public void a(j jVar) {
281.     if (jVar != null) {
282.         jVar.adVar();
283.     }
284. }
285.
286. public void a(j jVar) {
287.     if (jVar != null) {
288.         jVar.adVar();
289.     }
290. }
291.
292. public void a(j jVar) {
293.     if (jVar != null) {
294.         jVar.adVar();
295.     }
296. }
297.
298. public void a(j jVar) {
299.     if (jVar != null) {
300.         jVar.adVar();
301.     }
302. }
303.
304. public void a(j jVar) {
305.     if (jVar != null) {
306.         jVar.adVar();
307.     }
308. }
309.
310. public void a(j jVar) {
311.     if (jVar != null) {
312.         jVar.adVar();
313.     }
314. }
315.
316. public void a(j jVar) {
317.     if (jVar != null) {
318.         jVar.adVar();
319.     }
320. }
321.
322. public void a(j jVar) {
323.     if (jVar != null) {
324.         jVar.adVar();
325.     }
326. }
327.
328. public void a(j jVar) {
329.     if (jVar != null) {
330.         jVar.adVar();
331.     }
332. }
333.
334. public void a(j jVar) {
335.     if (jVar != null) {
336.         jVar.adVar();
337.     }
338. }
339.
340. public void a(j jVar) {
341.     if (jVar != null) {
342.         jVar.adVar();
343.     }
344. }
345.
346. public void a(j jVar) {
347.     if (jVar != null) {
348.         jVar.adVar();
349.     }
350. }
351.
352. public void a(j jVar) {
353.     if (jVar != null) {
354.         jVar.adVar();
355.     }
356. }
357.
358. public void a(j jVar) {
359.     if (jVar != null) {
360.         jVar.adVar();
361.     }
362. }
363.
364. public void a(j jVar) {
365.     if (jVar != null) {
366.         jVar.adVar();
367.     }
368. }
369.
370. public void a(j jVar) {
371.     if (jVar != null) {
372.         jVar.adVar();
373.     }
374. }
375.
376. public void a(j jVar) {
377.     if (jVar != null) {
378.         jVar.adVar();
379.     }
380. }
381.
382. public void a(j jVar) {
383.     if (jVar != null) {
384.         jVar.adVar();
385.     }
386. }
387.
388. public void a(j jVar) {
389.     if (jVar != null) {
390.         jVar.adVar();
391.     }
392. }
393.
394. public void a(j jVar) {
395.     if (jVar != null) {
396.         jVar.adVar();
397.     }
398. }
399.
399. }
```

It logs the composition value here.

c.java

```
1. package com.airbnb.lottie.x;
2.
3. import android.util.Log;
4. import com.airbnb.lottie.i;
5. import java.util.HashSet;
6. import java.util.Set;
7. /* compiled from: LogcatLogger.java */
8. /* loaded from: classes.dex */
9. public class c implements i {
10.     private static final Set<String> a = new HashSet();
11.
12.     @Override // com.airbnb.lottie.i
13.     public void a(String str, Throwable th) {
14.         if (a.contains(str)) {
15.             return;
16.         }
17.         Log.w("LOTTIE", str, th);
18.         a.add(str);
19.     }
20.
21.     public void b(String str, Throwable th) {
22.         if (com.airbnb.lottie.c.a) {
23.             Log.d("LOTTIE", str, th);
24.         }
25.     }
26.
27.     @Override // com.airbnb.lottie.i
28.     public void debug(String str) {
29.         b(str, null);
30.     }
31.
32.     @Override // com.airbnb.lottie.i
33.     public void warning(String str) {
34.         a(str, null);
35.     }
36. }
```

It also logs more information about Lottie in here.

2	Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	warning	CWE: CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	com/bumptech/glide/load/engine/d.java com/bumptech/glide/load/engine/p.java com/bumptech/glide/load/engine/w.java com/bumptech/glide/load/g.java com/stripe/android/EphemeralKey.java com/stripe/android/PaymentConfiguration.java com/stripe/android/auth/PaymentBrowserAuthContract.java com/stripe/android/googlepaylauncher/GooglePayLauncherContract.java com/stripe/android/googlepaylauncher/GooglePayPaymentMethodLauncherContract.java com/stripe/android/googlepaylauncher/GooglePayPaymentMethodLauncherViewModel.java
---	--	----------------	--	--

The application code also contains hardcoded sensitive information that aren't supposed to be hardcoded but soft coded instead, inside its classes.

Hard coding sensitive information exposes that information to attackers. The severity of this rule can vary depending on the kind of information that is disclosed. Frequently, the information disclosed is password or key information, which can lead to remote exploitation.

d.java

```

1. package com.bumptech.glide.load.engine;
2.
3. import java.security.MessageDigest;
4. /* compiled from: DataCacheKey.java */
5. /* loaded from: classes.dex */
6. final class d implements com.bumptech.glide.load.f {
7.     private final com.bumptech.glide.load.f b;
8.     private final com.bumptech.glide.load.f c;
9.
10.    /* JADX INFO: Access modifiers changed from: package-private */
11.    public d(com.bumptech.glide.load.f fVar, com.bumptech.glide.load.f fVar2) {
12.        this.b = fVar;
13.        this.c = fVar2;
14.    }
15.
16.    @Override // com.bumptech.glide.load.f
17.    public void a(MessageDigest messageDigest) {
18.        this.b.a(messageDigest);
19.        this.c.a(messageDigest);
20.    }
21.
22.    @Override // com.bumptech.glide.load.f
23.    public boolean equals(Object obj) {
24.        if (obj instanceof d) {
25.            d dVar = (d) obj;
26.            return this.b.equals(dVar.b) && this.c.equals(dVar.c);
27.        }
28.        return false;
29.    }
30.
31.    @Override // com.bumptech.glide.load.f
32.    public int hashCode() {
33.        return (this.b.hashCode() * 31) + this.c.hashCode();
34.    }
35.
36.
37.    public String toString() {
38.        return "DataCacheKey[sourceKey=" + this.b + ", signature=" + this.c + ']';
39.    }
}

```



This MessageDigest class provides applications the functionality of a message digest algorithm, such as SHA-1 or SHA-256. Message digests are secure one-way hash functions that take arbitrary-sized data and output a fixed-length hash value.

This class returns the value of this message digest class into the result, therefore exposing, as seen in the picture.

On the other hand, it also exposes information about the glide functionality, Glide supports fetching, decoding, and displaying video stills, images, and animated GIFs. Glide includes a flexible API that allows developers to plug in to almost any network stack.

It is implemented into the build Gradle file by adding the following dependencies:

```
repositories {
    google()
    mavenCentral()
}

dependencies {
    implementation 'com.github.bumptech.glide:glide:4.14.2'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.14.2'
}
```

As we can see here, it also exposes the hardcoded information about the Ephemeral Key as shown below:

```
public final boolean isLiveMode$payments_core_release() {
    return this.isLiveMode;
}

public String toString() {
    return "EphemeralKey{objectId=" + this.objectId + ", created=" + this.created + ", expires=" + this.expires + ", id=" + this.id + ", isLiveMode=" + this.isLiveMode + ", objectType=" + this.objectType + ", secret=" + this.secret + ", type=" + thi
}

@Override // android.os.Parcelable
public void writeToParcel(Parcel parcel, int i2) {
    super.writeToParcel(parcel, i2);
    parcel.writeString(this.toString());
}
```

It is imported from stripe android sdk:

EphemeralKey.java

```
1. package com.stripe.android;
2.
3. import android.os.Parcel;
4. import android.os.Parcelable;
5. import com.stripe.android.model.StripeModel;
6. import com.stripe.android.stripe3ds2.transactions.MessageExtension;
7. import kotlin.h0.d;
8. /* compiled from: EphemeralKey.kt */
9. /* loaded from: classes2.dex */
10. public final class EphemeralKey implements StripeModel {
11.     private final long created;
12.     private final long expires;
13.     private final String id;
14.     private final boolean isLiveMode;
15.     private final String objectId;
16.     private final String objectType;
17.     private final String secret;
18.     private final String type;
19.     public static final Parcelable.Creator<EphemeralKey> CREATOR = new Creator();
20.     public static final int $stable = 8;
21. }
```

Which makes it quick and easy to build an excellent payment experience in your Android app. They provide powerful and customizable UI elements that can be used out-of-the-box to collect the users' payment details. they also expose the low-level APIs that power those UIs so that you can build fully

custom experiences.				
3	App can read/write to External Storage. Any App can read data written to External Storage.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	com/github/mikephil/charting/charts/Chart.java com/github/mikephil/charting/utils/FileUtils.java

The application can also read/write to external storage, as seen here:

```
    }

    public boolean saveToPath(String str, String str2) {
        Bitmap chartBitmap = getChartBitmap();
        try {
            FileOutputStream fileOutputStream = new FileOutputStream(Environment.getExternalStorageDirectory().getPath() + str2 + "/" + str + ".png");
            chartBitmap.compress(Bitmap.CompressFormat.PNG, 40, fileOutputStream);
            fileOutputStream.close();
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

A file output stream is an output stream for writing data to a File, in this case, that file is written into external storage with the “`getExternalStorageDirectory`” function.

```
1. public static List<Entry> loadEntriesFromFile(String str) {
2.     File file = new File(Environment.getExternalStorageDirectory(), str);
3.     ArrayList arraylist = new ArrayList();
4.     try {
5.         BufferedReader bufferedReader = new BufferedReader(new FileReader(file));
6.         while (true) {
7.             String readline = bufferedReader.readLine();
8.             if (readline == null) {
9.                 break;
10.            }
11.            String[] split = readline.split("#");
12.            if (split.length <= 2) {
13.                arraylist.add(new Entry(Float.parseFloat(split[0]), Integer.parseInt(split[1])));
14.            } else {
15.                float[] fArr = new float[split.length - 1];
16.                for (int i2 = 0; i2 < length; i2++) {
17.                    fArr[i2] = Float.parseFloat(split[i2]);
18.                }
19.                arraylist.add(new BarEntry(Integer.parseInt(split[split.length - 1]), fArr));
20.            }
21.        }
22.    } catch (IOException e) {
23.        Log.e(LOG, e.toString());
24.    }
25.    return arraylist;
26. }
27.
28. public static void saveToSdCard(List<Entry> list, String str) {
29.     File file = new File(Environment.getExternalStorageDirectory(), str);
30.     if (!file.exists()) {
31.         try {
32.             file.createNewFile();
33.         } catch (IOException e) {
34.             Log.e(LOG, e.toString());
35.         }
36.     }
37.     try {
38.         package com.j256.ormlite.android.compar;
39.         import android.database.Cursor;
40.         import android.database.sqlite.SQLiteDatabase;
41.         import android.os.CancellationSignal;
42.         import android.support.v4.util.Compat;
43.         import android.support.v4.util.Compat.ApiCompatibility;
44.         /* loaded from: classes.dex */
45.         public class JellyBeanApiCompatibility extends BasicApiCompatibility {
46.             /* loaded from: classes.dex */
47.             protected void jellyBeanCancellationHook implements ApiCompatibility.CancellationHook {
48.                 private final CancellationSignal cancellationSignal = new CancellationSignal();
49.                 @Override // from com.j256.ormlite.android.compat.ApiCompatibility.CancellationHook
50.                 public void cancel() {
51.                     this.cancellationSignal.cancel();
52.                 }
53.             }
54.
55.             @Override // from com.j256.ormlite.android.compat.BasicApiCompatibility, com.j256.ormlite.android.compat.ApiCompatibility
56.             public ApiCompatibilityHook createCancellationHook() {
57.                 return new JellyBeanCancellationHook();
58.             }
59.
60.             /* loaded from: classes.dex */
61.             public Cursor query(SQLiteDatabase db, String str, String[] strArr, String str2, ApiCompatibility.CancellationHook cancellationHook) {
62.                 return null;
63.             }
64.         }
65.     }
66. }
```

They also write information from this class to external storage.

4	<p>App uses SQLite Database and execute raw SQL query. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.</p>	 warning	<p>CWE: CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')</p> <p>OWASP Top 10: M7: Client Code Quality</p>	<p>com/j256/ormlite/android/AndroidCompiledStatement.java com/j256/ormlite/android/AndroidDatabaseConnection.java com/j256/ormlite/android/compat/ApiCompatibility.java com/j256/ormlite/android/compat/BasicApiCompatibility.java com/j256/ormlite/android/compat/JellyBeanApiCompatibility.java h/v/a/g/.java j/b/b/a//a0/.java j/b/b/a//a0//t0.java j/b/b/a//a0/j/t0.java</p>	
---	---	---	---	--	---

Application uses SQL database which is not trusted.

Example of usage in some of the classes' code:

the application uses insecure random number generator which can cause danger to the functionalities it is used with, especially if it's for encrypting personal data of users.

```

48. import java.util.Iterator;
49. import java.util.Locale;
50. import java.util.Random; highlighted
51. import java.util.TimeZone;
52. import java.util.regex.Pattern;
53. import javax.crypto.IllegalBlockSizeException;
54. import javax.crypto.NoSuchPaddingException;
55. import org.json.JSONException;
56. import org.json.JSONObject;
57. /* compiled from: Util.java */
58. /* loaded from: classes.dex */
59. public class r {
60.     public static p b;

```

5 The App uses an insecure Random Number Generator.

warning

CWE: CWE-330: Use of Insecure Random Values
OWASP Top 10: M5: Insufficient Cryptography
OWASP MASVS: MSTG-CRYPTO-01

As seen below, it imports the util.random functionality:

```

7.
8.
9.
10. public static int q(Context context) {
11.     int[] intArray = context.getResources().getIntArray(R.array.androidcolors);
12.     int nextInt = new Random().nextInt(intArray.length);
13.     if (nextInt >= intArray.length) {
14.         nextInt++;
15.     }
16.     return intArray[nextInt];
17.

```

AndroidCompiledStatement.java

```

1. package com.j256.ormlite.android;
2.
3. import android.database.Cursor;
4. import android.database.SQLException;
5. import android.database.sqlite.SQLiteDatabase;
6. import android.database.sqlite.SQLiteStatement;
7. import com.j256.ormlite.android.compat.ApiCompatibility;
8. import com.j256.ormlite.android.compat.ApiCompatibilityUtils;
9. import com.j256.ormlite.dao.ObjectCache;
10. import com.j256.ormlite.field.SqlType;
11. import com.j256.ormlite.logger.Logger;

```

Public class random function: An instance of this class is used to generate a stream of pseudorandom numbers. The class uses a 48-bit seed, which is modified using a linear congruential formula.

If two instances of Random are created with the same seed, and the same sequence of method calls is made for each, they will generate and return identical sequences of numbers. Which means it doesn't provide semantic security or confidential security, it is very much advised to use a different method.

The application uses certificate pinning to detect Man in The Middle attacks

6 This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.

secure

OWASP MASVS: MSTG-NETWORK-4

[com/easypay/stripe/n/a.java](#)

What is certificate pinning?

okhttp3
Class CertificatePinner
java.lang.Object
 okhttp3.CertificatePinner

```
public final class CertificatePinner
extends Object
```

Constrains which certificates are trusted. Pinning certificates defends against attacks on certificate authorities. It also prevents connections through man-in-the-middle certificate authorities either known or unknown to the application's user.

This class currently pins a certificate's Subject Public Key Info as described on Adam Langley's Weblog. Pins are either base64 SHA-256 hashes as in HTTP Public Key Pinning (HPKP) or SHA-1 base64 hashes as in Chromium's static certificates.

As shown inside the class, this is how it's used:

```
t(300L, TimeUnit.SECONDS).certificatePinner(builder.add("www.facilepayforstripe.ca", "sha256/" + str).build()).addInterceptor(httpLoggingInterceptor).addInterceptor(statisticsInterceptor)
```

They use the SHA256 algorithm, which is very secure, and almost impossible to be cracked because it takes

There will also be around $36^{64} / 2^{256}$ or 34,600,000,000,000,000,000,000 collisions found. Note that the possible combinations of the string are greater than the number of possible hashes.

7	App creates temp file. Sensitive information should never be written into a temp file.	 warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	h/s/b.java
---	--	---	--	----------------------------

The application creates temporary file that contain sensitive information in the class called h/s/b.java

```
110.     private static void b(Closeable closeable) {
111.         try {
112.             closeable.close();
113.         } catch (IOException e) {
114.             Log.w("MultiDex", "Failed to close resource", e);
115.         }
116.     }
117.
118.     private static void c(ZipFile zipFile, ZipEntry zipEntry, File file, String str) {
119.         InputStream inputStream = zipFile.getInputStream(zipEntry);
120.         File createTempFile = File.createTempFile("tmp-" + str, ".zip", file.getParentFile());
121.         Log.i("MultiDex", "Extracting " + createTempFile.getPath());
122.         try {
123.             ZipOutputStream zipOutputStream = new ZipOutputStream(new BufferedOutputStream(new FileOutputStream(createTempFile)));
124.             ZipEntry zipEntry2 = new ZipEntry("classes.dex");
125.             zipEntry2.setTime(zipEntry.getTime());
126.             zipOutputStream.putNextEntry(zipEntry2);
127.             byte[] bArr = new byte[Http2.INITIAL_MAX_FRAME_SIZE];

```

The application uses bad encryption modes that are vulnerable to attacks, such as the PKCS5/PKCS7 paddings.

8	<p>The App uses the encryption mode CBC with PKCS5/PKCS7 padding. This configuration is vulnerable to padding oracle attacks.</p>	high	<p>CWE: CWE-649: Reliance on Obfuscation or Encryption of Security-Relevant Inputs without Integrity Checking OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-3</p>	com/easypay/stripe/activity/LoginFingerPrintActivity.java com/easypay/stripe/util/g.java
---	---	------	---	---

As shown previously in the class that is responsible for encrypting the fingerprint:

```

    public boolean H() {
        try {
            this.f1932k = Cipher.getInstance("AES/CBC/PKCS7Padding");
            try {
                this.f1931j.load(null);
                this.f1932k.init(1, (SecretKey) this.f1931j.getKey("EasyPay", null));
                return true;
            } catch (KeyPermanentlyInvalidatedException unused) {
                return false;
            } catch (IOException e) {
                e = e;
                throw new RuntimeException("Failed to init Cipher", e);
            } catch (InvalidKeyException e2) {

```

As seen here, the finger print has been encrypted with it.

```

/* JADX INFO: Access modifiers changed from: protected */
@Override // com.easypay.stripe.g, androidx.fragment.app.d, androidx.activity.ComponentActivity, androidx.core.app.g, android.app.Activity
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    ActivityLoginFingerPrintBinding activityLoginFingerPrintBinding = (ActivityLoginFingerPrintBinding) androidx.databinding.g.g(this, R.layout.activity_login_finger_print);
    KeyguardManager keyguardManager = (KeyguardManager) getSystemService("keyguard");
    FingerprintManager fingerprintManager = (FingerprintManager) getSystemService("fingerprint");
    if (!fingerprintManager.isHardwareDetected()) {
        activityLoginFingerPrintBinding.tvError.setText(getString(R.string.msg_no_fingerprint_sensor));
    } else if (androidx.core.content.a.a(this, "android.permission.USE_FINGERPRINT") != 0) {
        activityLoginFingerPrintBinding.tvError.setText(getString(R.string.msg_fingerprint_permission_not_enable));
    } else if (!fingerprintManager.hasEnrolledFingerprints()) {
        activityLoginFingerPrintBinding.tvError.setText(getString(R.string.msg_register_atlist_one_fingerprint));
    } else if (!keyguardManager.isKeyguardSecure()) {
        activityLoginFingerPrintBinding.tvError.setText(getString(R.string.msg_lock_screen_security_not_enable_in_setting));
    } else {
        I();
        if (H()) {
            new com.easypay.stripe.util.j(this).a(fingerprintManager, new FingerprintManager.CryptoObject(this.f1932k)); ↴
        }
    }
}

```

Also used here:

```

/* loaded from: classes.dex */
public class g {
    Cipher a = Cipher.getInstance("AES/CBC/PKCS5Padding");
    byte[] b = new byte[32];
    byte[] c = new byte[16];

    /* JADX INFO: Access modifiers changed from: private */
    /* compiled from: CriptoHelper.java */
    /* loaded from: classes.dex */
    public enum a {
        ENCRYPT,
        DECRYPT
    }
}

```

Application copies sensitive information to clipboard which can put it at risk of being used by other applications for malicious reasons.

10	This App copies data to clipboard. Sensitive data should not be copied to clipboard as other applications can access it.	Info	OWASP MASVS: MSTG-STORAGE-10	com/easypay/stripe/activity/PaymentLinkSendActivity.java
----	--	----------------------	-------------------------------------	--

As shown below:

```

}

/* JADX INFO: Access modifiers changed from: private */
public void P() {
    ((ClipboardManager) getSystemService("clipboard")).setPrimaryClip(ClipData.newPlainText(AnnotatedPrivateKey.LABEL, this.f1962k));
    com.easypay.stripe.util.r.k0(this, getString(R.string.lbl_copy_to_clipboard_msg));
}

private void Q(String str, final ImageView imageView) {
    try {
        ...
    }
}

```

I) Shared library binary analysis:

We get the libraries that are used in the application with all related information:

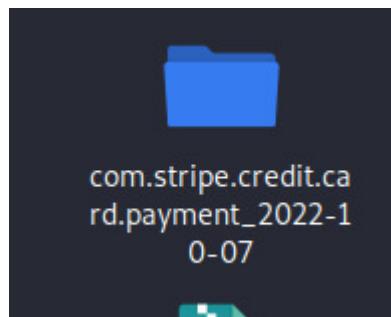
NO ↴	SHARED OBJECT	NX	STACK CANARY	RPATH	RUNPATH	FORTIFY	SYMBOLS STRIPPED ↴
1	lib/x86_64/libcardioRecognizer_tegra2.so	True Info	True Info	None Info	None Info	False Warning	True Info Symbols are stripped.
2	lib/x86_64/libcardioRecognizer.so	True Info	True Info	None Info	None Info	False Warning	True Info Symbols are stripped.
3	lib/x86_64/libcardioDecider.so	True Info	True Info	None Info	None Info	False Warning	True Info Symbols are

We use the apktool in order to decompress the .apk file and get its components:

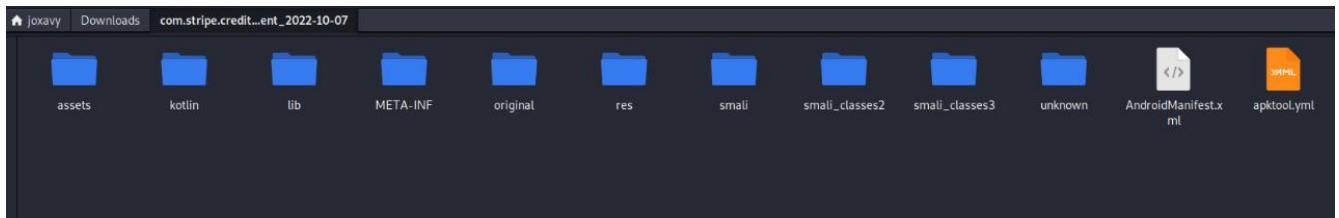
```
—(joxavy㉿kali)-[~/Downloads]
└─$ apktool d com.stripe.credit.card.payment_2022-10-07.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1-dirty on com.stripe.credit.card.payment_2022-10-07.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /home/joxavy/.local/share/apktool/framework/1.apk
I: Regular manifest package ...
I: Decoding file-resources ...
I: Decoding values */* XMLs ...
I: Baksmaling classes.dex ...
I: Baksmaling classes2.dex ...
I: Baksmaling classes3.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...
I: Copying META-INF/services directory

—(joxavy㉿kali)-[~/Downloads]
└─$
```

A folder holding the same name as the apk file, gets created:

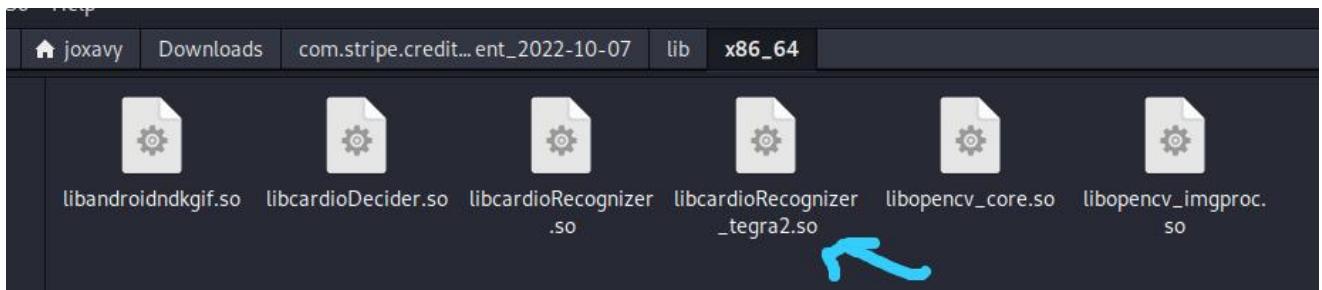


We check the components of this folder, we see:



After entering the lib folder, we find different folders that are issued for distinct architectures like x86, Arm,etc.

The file we are interested in is located inside the x86 architecture's folder:

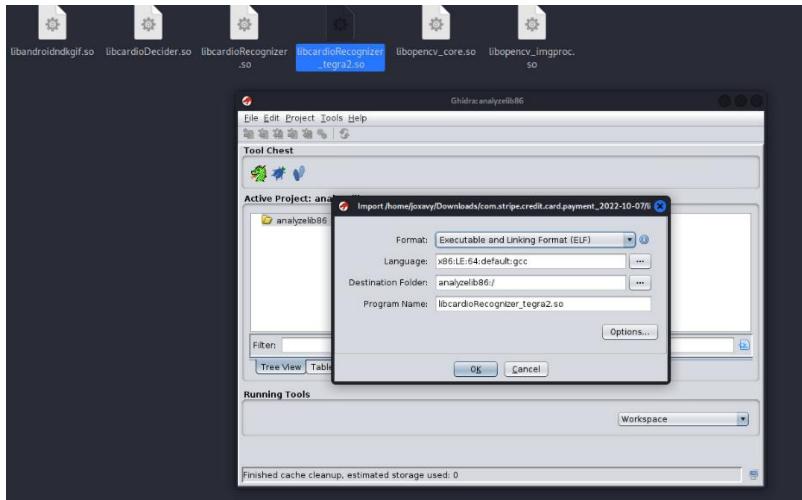


That is the first file that appeared in analysis table:

NO ↑	SHARED OBJECT	NX	STACK CANARY	RPATH	RUNPATH	FORTIFY	SYMBOLS STRIPPED
1	lib/x86_64 /libcardioRecognizer_tegra2.so	True <small>info</small> The shared object has NX bit set. This marks a memory page non-executable making attacker injected shellcode non-executable.	True <small>info</small> This shared object has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.	None <small>info</small> The shared object does not have run-time search path or RPATH set.	None <small>info</small> The shared object does not have RUNPATH set.	False <small>warning</small> The shared object does not have any fortified functions. Fortified functions provide buffer overflow checks against glibc's common insecure functions like strcpy, gets etc. Use the compiler option -D_FORTIFY_SOURCE=2 to fortify functions.	True <small>info</small> Symbols are stripped.

We are going to check it for fortified functions as the warning states it doesn't have any.

First step we will take, is that we will open the .so file with Ghidra:



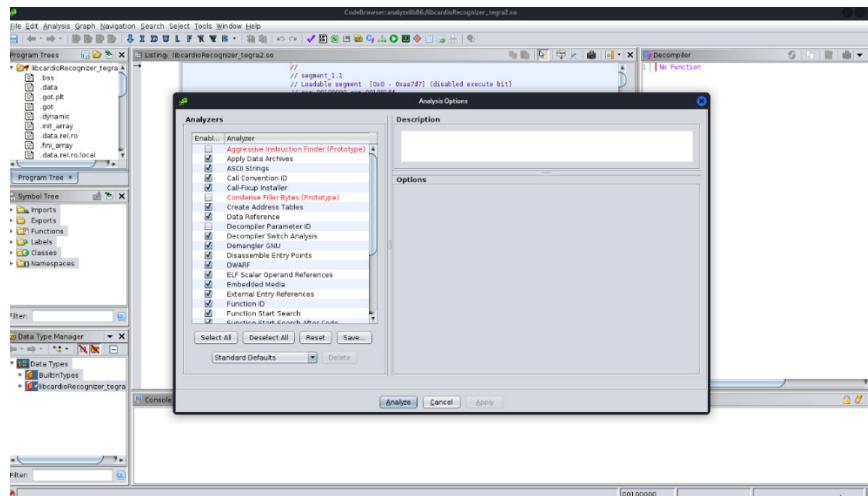
We get general information about the file:

Import Results Summary

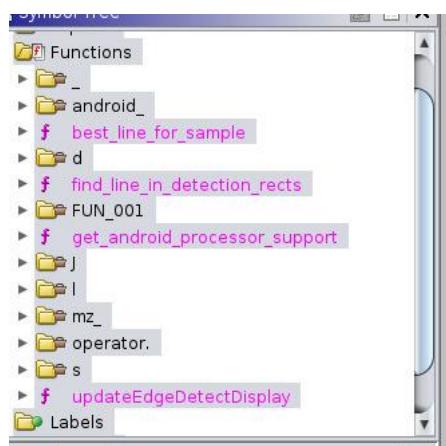
Project File Name: libcardioRecognizer_tegra2.so
Last Modified: Tue Dec 13 10:47:18 EST 2022
Readonly: false
Program Name: libcardioRecognizer_tegra2.so
Language ID: x86:LE:64:default (2.13)
Compiler ID: gcc
Processor: x86
Endian: Little
Address Size: 64
Minimum Address: 00100000
Maximum Address: _elfSectionHeaders::000006ff
of Bytes: 984928
of Memory Blocks: 30
of Instructions: 854
of Defined Data: 1410
of Functions: 489
of Symbols: 577
of Data Types: 45
of Date Type Categories: 2
Created With Ghidra Version: 10.1.4
Date Created: Tue Dec 13 10:47:15 EST 2022
ELF File Type: shared object
ELF Original Image Base: 0x0
ELF Prelinked: false
ELF Required Library [0]: libopencv_imgproc.so
ELF Required Library [1]: libopencv_core.so
ELF Required Library [2]: liblog.so
ELF Required Library [3]: libstdc++.so
ELF Required Library [4]: libgraphics.so
ELF Required Library [5]: libdl.so
ELF Required Library [6]: libc.so
ELF Required Library [7]: libm.so
ELF Required Library [8]: libstdc++.so
Executable Format: Executable and Linking Format (ELF)
Executable Location: /home/joxavy/Downloads/com.stripe.credit.card.payment_2022-10-07/lib/x86_64/
Executable MD5: 1ad6cb2f58135b055a2fd6d61e8435f1
Executable SHA256: a288e5351fe1ef1be4fd1d8ddcbe397352b1b39c5d3c5c3b76abde24c4119d
FSL: file:///home/joxavy/Downloads/com.stripe.credit.card.payment_2022-10-07/lib/
Relocatable: true

Additional Information

Analyze the file:



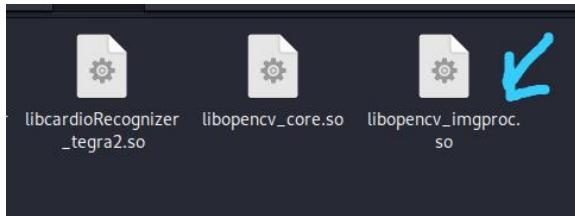
After the analysis is done, we check the functions' part:



As we can see we can't find the fortified functions.

I will then proceed to check the second file of the table that is "libopencv_imgproc.so" in order to detect if it contains stack canary values.

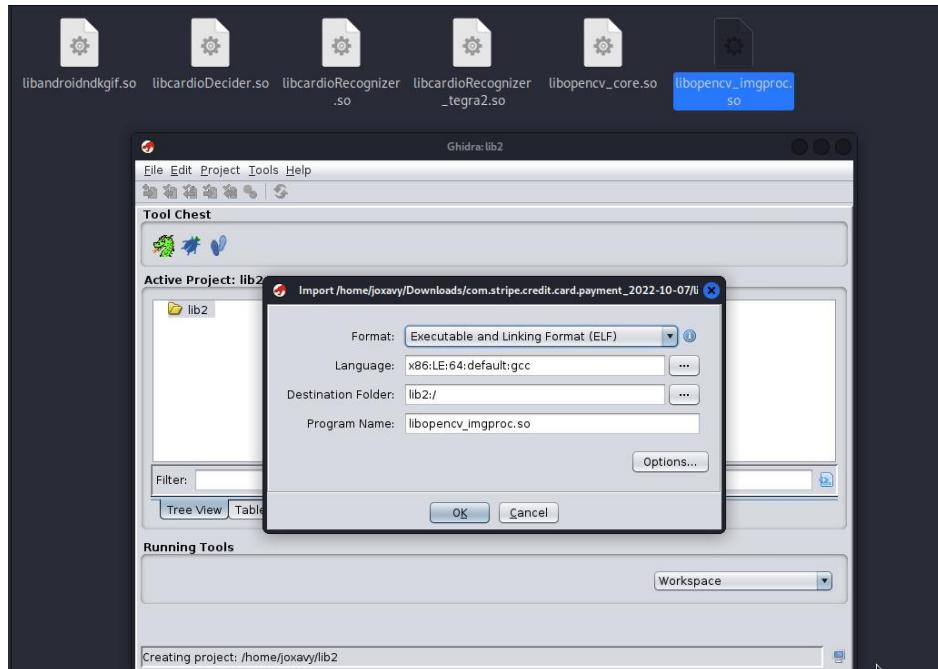
What are stack canaries? They are a secret value placed on the stack which changes every time the program is started. Prior to a function return, the stack canary is checked and if it appears to be modified, the program exits immediately.



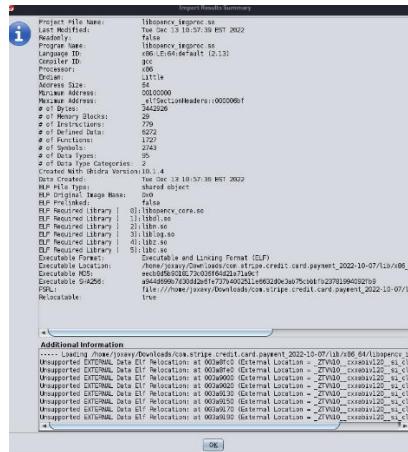
As stated in the warning

		True <small>Info</small>	False <small>High</small>	None <small>Info</small>	None <small>Info</small>	False <small>Warning</small>	
5	lib/x86_64/libopencv_imgproc.so	The shared object has NX bit set. This marks a memory page non-executable making attacker injected shellcode non-executable.	This shared object does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries.	The shared object does not have run-time search path or RPATH set.	The shared object does not have RUNPATH set.	The shared object does not have any fortified functions. Fortified functions provides buffer overflow checks against glibc's commons insecure functions like strcpy, gets etc. Use the compiler option -D_FORTIFY_SOURCE=2 to fortify functions.	True <small>Info</small> Symbols are stripped.

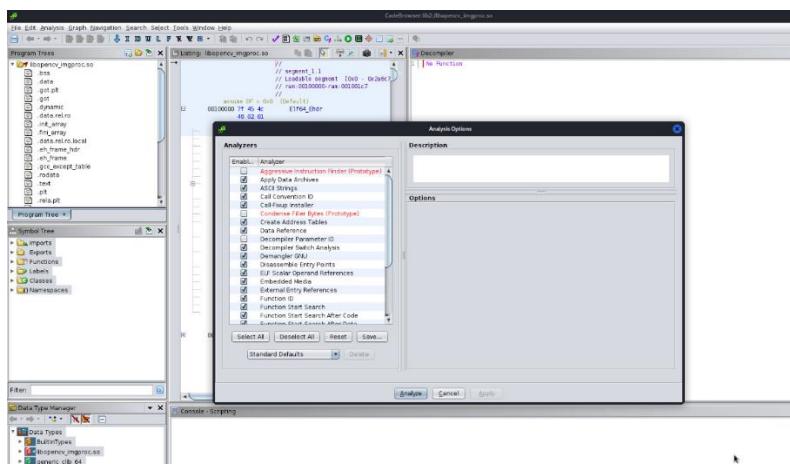
We open it with Ghidra:



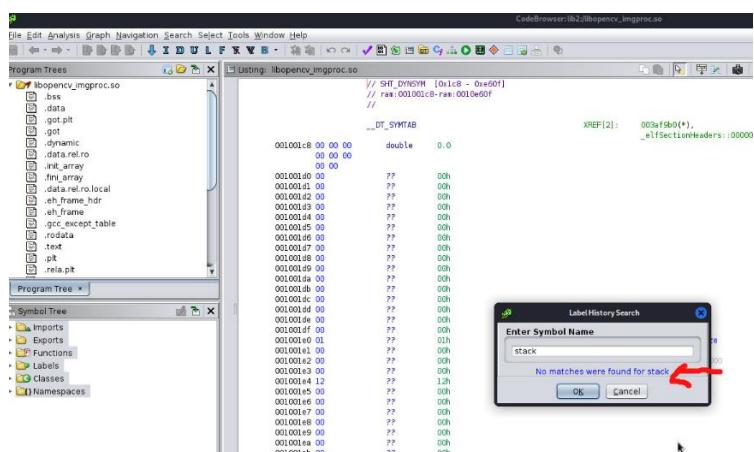
The general information about this library is shown as follows:



We analyze it:



After searching for it, we see that there is no match.



I will then compare it with another library to see the difference, here is the result after analyzing "libcardReconginzer_tegra2.so" library:

Address	Action	Label	User	Modification
001f1061	Add	__stack_chk_fail	joxavy	Dec 13, 20...
001f1061	Add	__stack_chk_fail	joxavy	Dec 13, 20...
001f1061	Rename	__stack_chk_fa...	joxavy	Dec 13, 20...
001f1061	Remove	__stack_chk_fa...	joxavy	Dec 13, 20...
EXTER...	Add	__stack_chk_fail	joxavy	Dec 13, 20...

As I search for the stack canaries, I get the output above.

m) Niap analysis:

NIAP ANALYSIS v1.3				
NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION
11	FCS_CKM.1.1(1)	Selection-Based Security Functional Requirements	Cryptographic Asymmetric Key Generation	The application generate asymmetric cryptographic keys not in accordance with FCS_CKM.1.1(1) using key generation algorithm RSA schemes and cryptographic key sizes of 1024-bit or lower.
12	FCS_CKM.1.1(3),FCS_CKM.1.3(2)	Selection-Based Security Functional Requirements	Password Conditioning	A password/passphrase shall perform [Password-based Key Derivation Functions] in accordance with a specified cryptographic algorithm..
13	FCS_COP.1.1(1)	Selection-Based Security Functional Requirements	Cryptographic Operation - Encryption/Decryption	The application perform encryption/decryption in accordance with a specified cryptographic algorithm AES-CBC (as defined in NIST SP 800-38A) mode or AES-GCM (as defined in NIST SP 800-38D) and cryptographic key sizes 256-bit/128-bit.
14	FCS_COP.1.1(2)	Selection-Based Security Functional Requirements	Cryptographic Operation - Hashing	The application perform cryptographic hashing services not in accordance with FCS_COP.1.1(2) and uses the cryptographic algorithm RC2/RC4/MD4/MD5.

What are the NIAP "Requirements for Vetting Mobile Applications"?

NIAP is a collaboration between the National Institute for Standards and Technology (NIST) and the National Security Agency (NSA).

NIAP oversees the Common Criteria evaluation and certification guidelines used to evaluate whether commercial IT products meet security standards for government deployments.

The NIAP mobile app vetting requirements grew out of the Protection Profile for Application Software. The profile explains the security standards against which an application will be assessed before it's approved for deployment within a government environment.

Why is there a warning?

To understand it, first, we need to explain what FCS_CKM1.1 is.

FCS_CKM1.1, Cryptographic key generation, is a component that requires the cryptographic key sizes and method used to generate cryptographic keys to be specified, this can be in accordance with an assigned standard. It should be used to specify the cryptographic key sizes and the method (e.g. algorithm) used to generate the cryptographic keys.

Only one instance of the component is needed for the same method and multiple key sizes. The key size could be common or different for the various entities, and could be either the input to or the output from the method.

The application uses an asymmetric key that is not in accordance with the FCS_CKM1.1 key generation method.

14	FCS_COP.1.1(2)	Selection-Based Security Functional Requirements	Cryptographic Operation - Hashing	The application perform cryptographic hashing services not in accordance with FCS_COP.1.1(2) and uses the cryptographic algorithm RC2/RC4/MD4/MD5.
----	----------------	--	-----------------------------------	--

As detected in the problem, the application also uses a cryptographic hashing service that isn't in accordance with the FCS_COP.1.1 and instead uses the algorithms RC2/RC4/MD4/MD5.

Here is a list of the algorithms that are actually supposed to be used with this operation:

FCS_COP.1(2) Cryptographic Operation - Hashing (Refined)

FCS_COP.1.1(2)

The OS shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [selection]:

- SHA-1,
- SHA-256,
- SHA-384,
- SHA-512]

and message digest sizes [selection]:

- 160 bits,
- 256 bits,
- 384 bits,
- 512 bits]

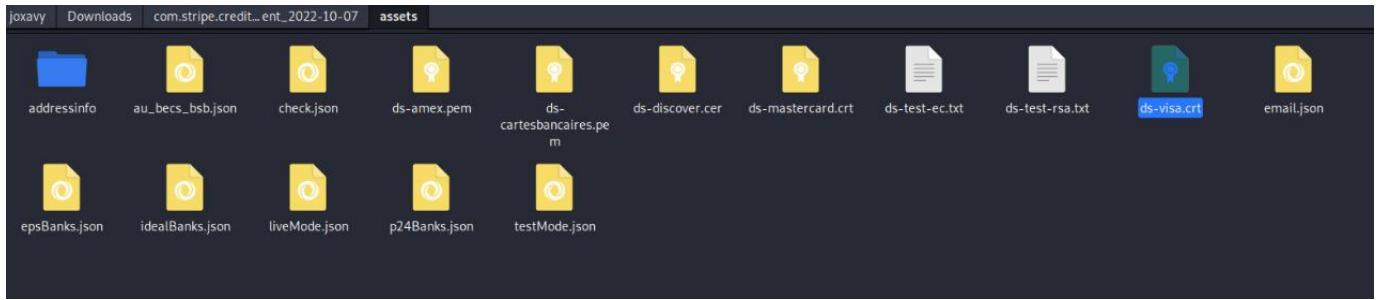
n) File analysis:

FILE ANALYSIS		Se:
NO	ISSUE	FILES
1	Certificate/Key files hardcoded inside the app.	META-INF/services/io.grpc.ManagedChannelProvider META-INF/services/java.security.Provider assets/ds-amex.pem assets/ds-cartesbancaires.pem assets/ds-discover.cer assets/ds-mastercard.crt assets/ds-visa.crt

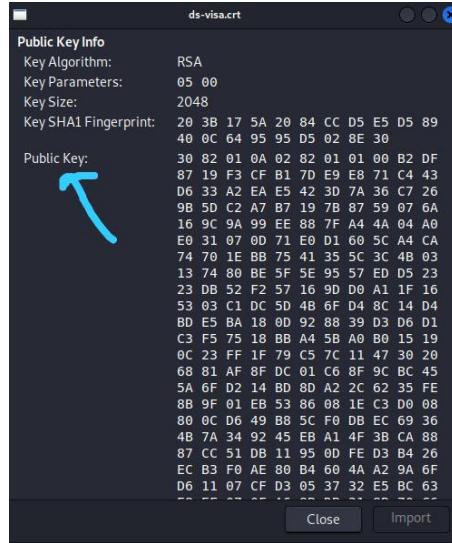
Showing 1 to 1 of 1 entries

It displays the files where the certificate/key files are hardcoded.

We will check one of these files by accessing their location on the decompressed apk folder:

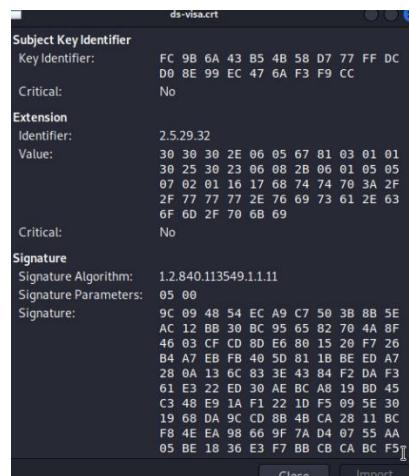


Such as the ds-visa.crt file:

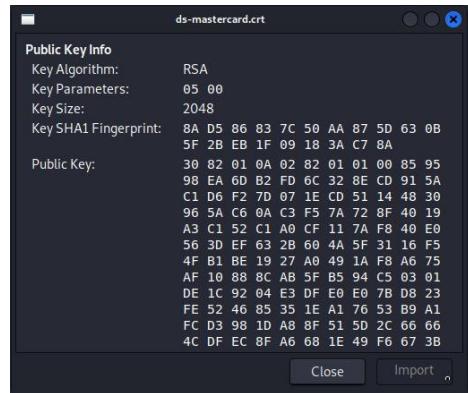


After clicking it, we see the public key used to encrypt this file as shown above.

There is also information about the signature:



We can also check the other ds-mastercard.crt file:



The public key for it is also shown, even the SHA1 fingerprint.

o) Server locations:

The tool also provides the server locations which are connected to the app.



p) Firebase database:

FIREBASE DATABASE

FIREBASE URL	DETAILS
https://easypay-for-stripe.firebaseio.com	Info App talks to a Firebase database.

Showing 1 to 1 of 1 entries

q) Trackers:

TRACKERS

TRACKER NAME	CATEGORIES	URL
Google CrashLytics	Crash reporting	https://reports.exodus-privacy.eu.org/trackers/27
Google Firebase Analytics	Analytics	https://reports.exodus-privacy.eu.org/trackers/49

Showing 1 to 2 of 2 entries

As we can see here, it shows which trackers are tracking users' information.

r) Activities:

We can see all the different activities that are used in the app:

ACTIVITIES

```
com.easypay.stripe.activity.ChangePassword  
com.easypay.stripe.activity.ForgotPassword  
com.easypay.stripe.activity.LoginActivity  
com.easypay.stripe.activity.SignUpActivity  
com.easypay.stripe.activity.RecurringPaymentCreateActivity  
com.easypay.stripe.activity.RecurringChargeDetailFromCustomer  
com.easypay.stripe.activity.RecurringPaymentDetailActivity  
com.easypay.stripe.activity.WebViewLoadActivity  
com.easypay.stripe.activity.QrCodeGenerateActivity  
com.easypay.stripe.activity.PaymentLinkSendActivity  
com.easypay.stripe.activity.PaymentLinkActivity  
com.easypay.stripe.activity.PaidPdfLoadActivity  
com.easypay.stripe.activity.CustomerListActivity  
com.easypay.stripe.activity.EditInvoiceItemActivity  
com.easypay.stripe.activity.NewInvoiceActivity  
com.easypay.stripe.activity.InvoiceItemListActivity  
com.easypay.stripe.activity.AddInvoiceItemActivity  
com.easypay.stripe.activity.EditDiscountActivity  
com.easypay.stripe.activity.EditTaxActivity  
com.easypay.stripe.activity.AddDiscountActivity  
com.easypay.stripe.activity.DiscountListActivity  
com.easypay.stripe.activity.AddTaxActivity  
com.easypay.stripe.activity.TaxRateListActivity  
com.easypay.stripe.activity.InvoiceSendActivity  
com.easypay.stripe.activity.InvoiceViewActivity  
com.easypay.stripe.activity.SplashActivity  
com.easypay.stripe.activity.MainActivity  
com.easypay.stripe.setup.SelectLanguageActivity  
com.easypay.stripe.activity.WebConnectActivity
```

s) Services:

We can also see the services used:

SERVICES

```
com.easypay.stripe.notification.MyFirebaseMessagingService  
com.easypay.stripe.services.CallingApiService  
com.easypay.stripe.services.AccountSync ApiService  
com.google.firebaseio.components.ComponentDiscoveryService  
com.google.android.gms.cast.framework.media.MediaNotificationService  
com.google.android.gms.cast.framework.ReconnectionService  
com.google.firebase.messaging.FirebaseMessagingService  
com.google.android.gms.auth.api.signin.RevocationBoundService  
com.google.android.gms.measurement.AppMeasurementService  
com.google.android.gms.measurement.AppMeasurementJobService  
androidx.work.impl.background.systemalarm.SystemAlarmService  
androidx.work.impl.background.systemjob.SystemJobService  
androidx.work.impl.foreground.SystemForegroundService  
com.google.android.datatransport.runtime.backends.TransportBackendDiscovery  
com.google.android.datatransport.runtime.scheduling.jobscheduling.JobInfoSchedulerService  
androidx.room.MultiInstanceInvalidationService
```

t) The receivers and the providers:

RECEIVERS

```
com.google.android.gms.cast.framework.media.MediaIntentReceiver  
com.google.firebaseio.id.FirebaseInstanceIdReceiver  
com.google.android.gms.measurement.AppMeasurementReceiver  
androidx.work.impl.utils.ForeStopRunnable$BroadcastReceiver  
androidx.work.impl.background.systemalarm.ConstraintProxy$BatteryChargingProxy  
androidx.work.impl.background.systemalarm.ConstraintProxy$BatteryNotLowProxy  
androidx.work.impl.background.systemalarm.ConstraintProxy$StorageNotLowProxy  
androidx.work.impl.background.systemalarm.ConstraintProxy$NetworkStateProxy  
androidx.work.impl.background.systemalarm.RescheduleReceiver  
androidx.work.impl.background.systemalarm.ConstraintProxy$UpdateReceiver  
androidx.work.impl.diagnostics.DiagnosticsReceiver  
com.google.android.datatransport.runtime.scheduling.jobscheduling.AlarmManagerSchedulerBroadcastReceiver  
androidx.profileinstaller.ProfileInstallReceiver
```

PROVIDERS

```
androidx.core.content.FileProvider  
com.google.firebaseio.perf.provider.FirebasePerfProvider  
com.google.firebaseio.provider.FirebaseInitProvider  
androidx.work.impl.WorkManagerInitializer  
androidx.startup.InitializationProvider
```

u) The libraries and files:

```


    LIBRARIES
    org.apache.http.legacy

    FILES
    AndroidManifest.xml
    DebugProbesKt.bin
    META-INF/3ds2dk_release.kotlin_module
    META-INF/SOSegmentSDK_release.kotlin_module
    META-INF/activity-compose_release.kotlin_module
    META-INF/activity-ktx_release.kotlin_module
    META-INF/androidx.activity_activity-compose.version
    META-INF/androidx.activity_activity-ktx.version
    META-INF/androidx.activity_activity_version
    META-INF/androidx.annotation_annotation-experimental.version
    META-INF/androidx.appcompat_appcompat_resources.version
    META-INF/androidx.appcompat_appcompat.version
    META-INF/androidx.core_core-runtime.version
    META-INF/androidx.asyncLayoutInflater_asyncLayoutInflater.version
    META-INF/androidx.autofill_autofill.version
    META-INF/androidx.biometric_biometric.version
    META-INF/androidx.browser_browser.version
    META-INF/androidx.cardview_cardview.version
    META-INF/androidx.compose.animation_animation-core.version
    META-INF/androidx.compose.animation_animation-version
    META-INF/androidx.compose.foundation_foundation-layout.version


```

IV. Dynamic analysis:

1. Dynamic analysis with MobSF:

We need to select the dynamic analyzer:

APP SCORES

FacilePay
Security Score: 48/100
Trackers Detected: 2/421

FILE INFORMATION

File Name: com.stripe.credit.card.payment_2022-10-07.apk
File Size: 32.89MB
MD5: Geaa1256a97972a541f42ece2b727c51d
SHA1: 54ec3a59e2cd546f042bc8d1ced47beaa124291
SHA256: f7d57fde88ad404a1480d21c519c37d2c218e41f255066cf1293e7f3b6c4c6da

APP INFORMATION

App Name: FacilePay
Package Name: com.stripe.credit.card.payment
Main Activity: com.easypay.stripe.activity.SplashActivity
Target SDK: 30 Min SDK: 21 Max SDK: 108
Android Version Range: 7.7.1G Android Version Count: 108

PLAYSTORE INFORMATION

Rating: 4.5384617 (100,000+)
Developer: Vbridge Technologies Inc.
Category: Finance
Play Store URL: com.stripe.credit.card.payment
Developer Address: 2 County Court Blvd., Suite 400, Office Number: 429 Brampton, L6W 3W8, Ontario, Canada
Developer Website: https://www.facilepay.ca/
Developer Email: support@facilepay.ca
Release Date: Feb 16, 2018 Privacy Policy: Privacy link
Description: FacilePay - Payment for Stripe makes it easy for tradespeople and other small business owners to collect credit/debit card payment right on the spot with their smartphone!

And start analyzing the app by clicking on the start analysis button.

MobSF Dynamic Analyzer Supports

- Genymotion Android VM version 4.1 - 10.0 (x86, upto API 29)
- Android Emulator AVD (non production) version 5.0 - 9.0 (arm, arm64, x86, and x86_64 upto API 28)

Recommended Android version is 7.0 +
Detected Android Version: 10.0, SDK: 29
Frida will be used for Instrumentation.

MobSF Android Runtime

Android instance: 127.0.1:5555

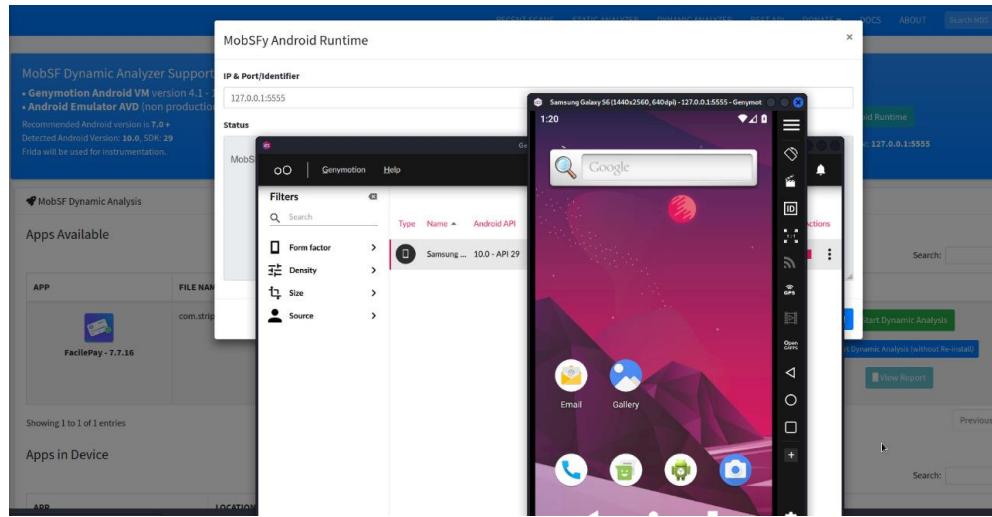
MobSF Dynamic Analysis

APP	FILE NAME	PACKAGE	ACTION
FacilePay - 7.7.1G	com.stripe.credit.card.payment_2022-10-07.apk	com.stripe.credit.card.payment	<input type="button" value="Start Dynamic Analysis"/> <input type="button" value="Start Dynamic Analysis (without Frida)"/> <input type="button" value="View Report"/>

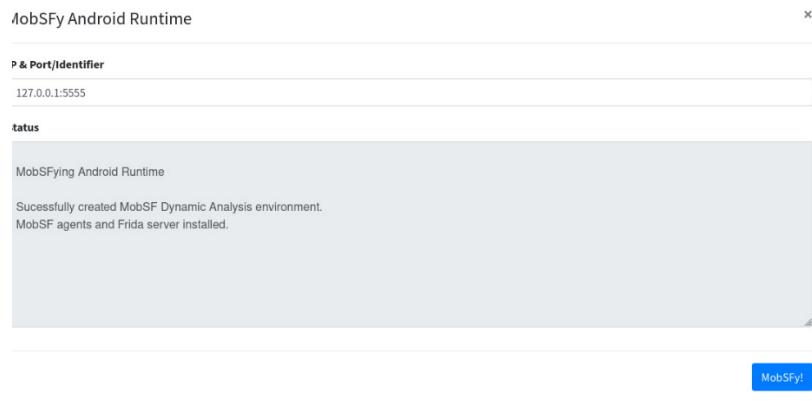
Showing 1 to 1 of 1 entries

Which requires Genymotion to be already running and connected, after installing it, we launch the emulator and connect it to the dynamic analyzer of MOBSF.

It automatically detects the emulator and connects to it via the Ip address 127.0.0.1 and the port 5555

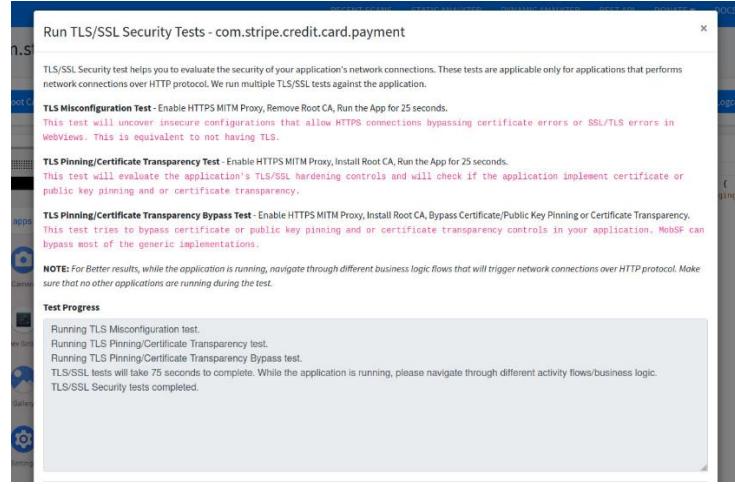


As seen this result:



The environment has been successfully created.

After connecting it, we can start by running TLS/SSL security tests and see if the tool can bypass them:



After running them, we get the following result:

TESTS	RESULT
TLS Misconfiguration Test	✓
TLS Pinning/Certificate Transparency Test	✗
TLS Pinning/Certificate Transparency Bypass Test	✗
Cleartext Traffic Test	✓

Which means that the tool was able to bypass both TLS misconfiguration test and cleartext traffic test.

2. Dynamic analysis with burp suite:

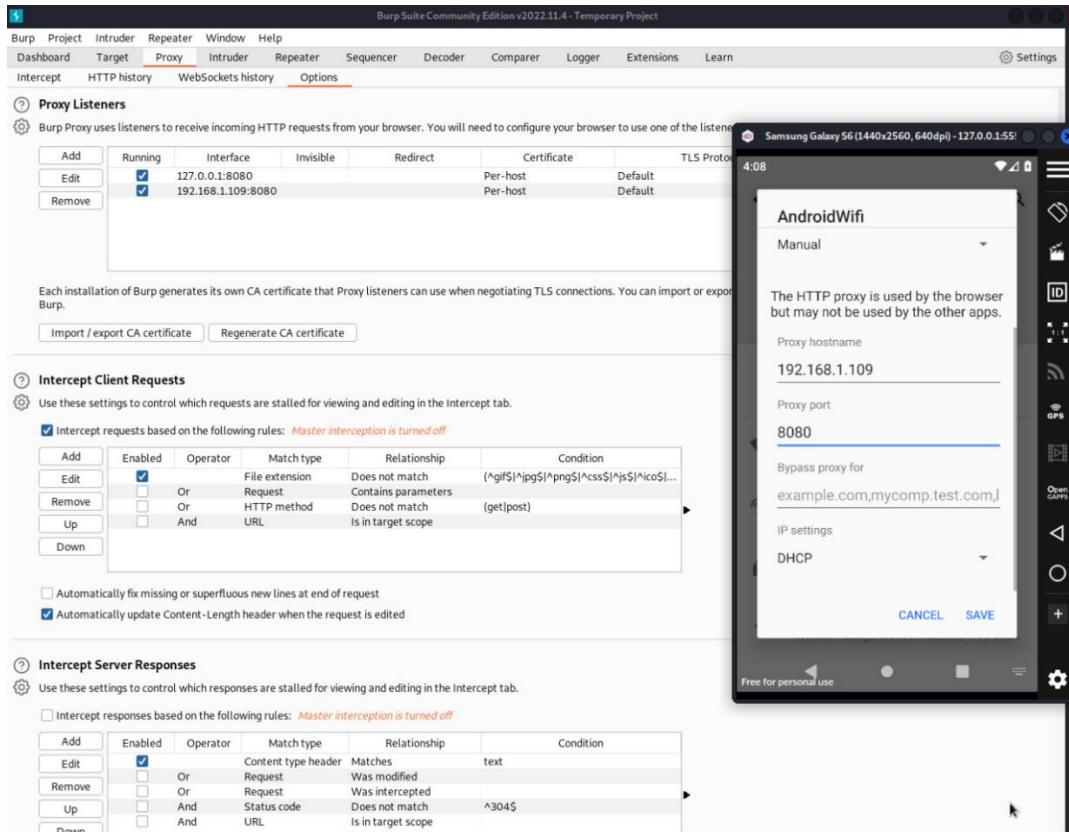
After running the genymotion emulator, we will download the apk file into it.

And install burp suite tool:

```
(joxavy㉿kali)-[~]
└─$ sudo apt install burpsuite
[sudo] password for joxavy:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  android-libborongssl faraday-client libarmadillo10 libcharls2 libgdal29 libgeos3.10.1 libicu71:i386
  libigdmm11 libodbc1 libodbcrc2 libpython3.9-dev libghull8.0 libtbb2 libvpx6 odbcinst odbcinstdebian2
  python-mpltoolkits.basemap-data python3-deprecation python3-lvmlite python3-pyproj python3-pyshp python3.9
  python3.9-dev python3.9-minimal
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  burpsuite
1 upgraded, 0 newly installed, 0 to remove and 1420 not upgraded.
Need to get 230 MB of archives.
After this operation, 42.3 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 burpsuite amd64 2022.11.4-0kali1 [230 MB]
Ign:1 http://kali.download/kali kali-rolling/main amd64 burpsuite amd64 2022.11.4-0kali1
Get:1 http://kali.download/kali kali-rolling/main amd64 burpsuite amd64 2022.11.4-0kali1 [230 MB]
Fetched 39.0 MB in 7min 31s (86.3 kB/s)
(Reading database ... 307006 files and directories currently installed.)
Preparing to unpack .../burpsuite_2022.11.4-0kali1_amd64.deb ...
Unpacking burpsuite (2022.11.4-0kali1) over (2021.10.2-0kali3) ...
Setting up burpsuite (2022.11.4-0kali1) ...
Processing triggers for kali-menu (2021.4.2) ...
Scanning processes ...
Scanning processor microcode ...
Scanning linux images ...

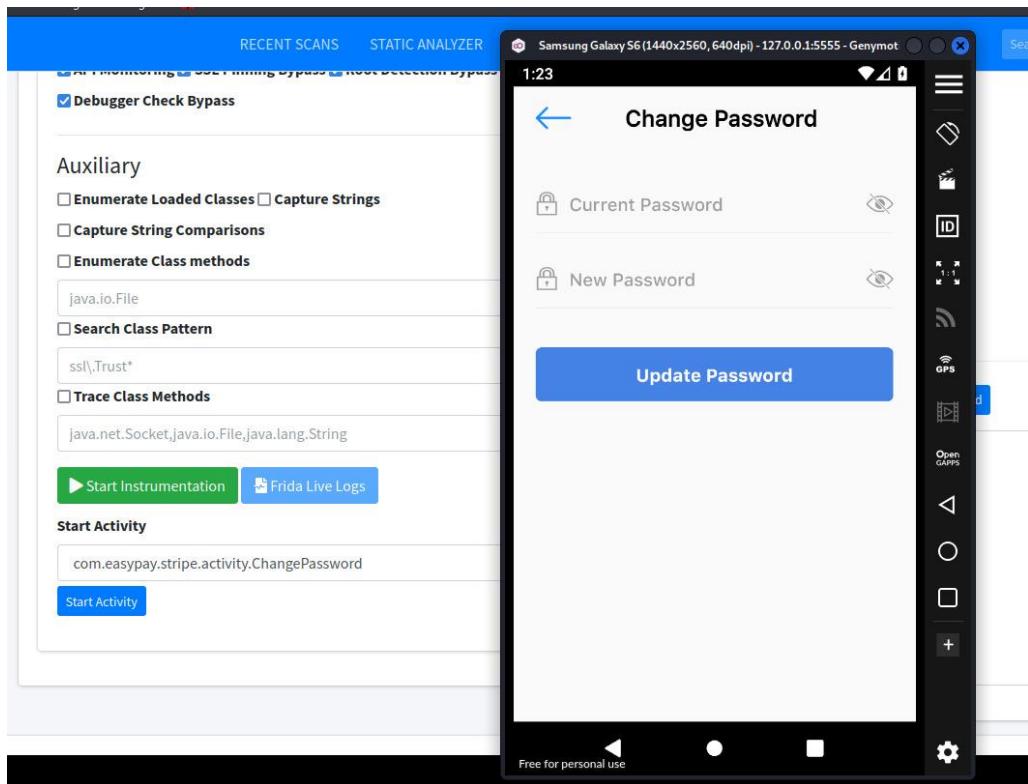

```

And then, set up a proxy inside burpsuite, by going to proxy, options and add, give it the following ip address: 192.168.1.109 with the port number:8080



Do the same thing inside the emulator, in order to connect it to the burpsuite proxy.

After that, we will launch an activity:

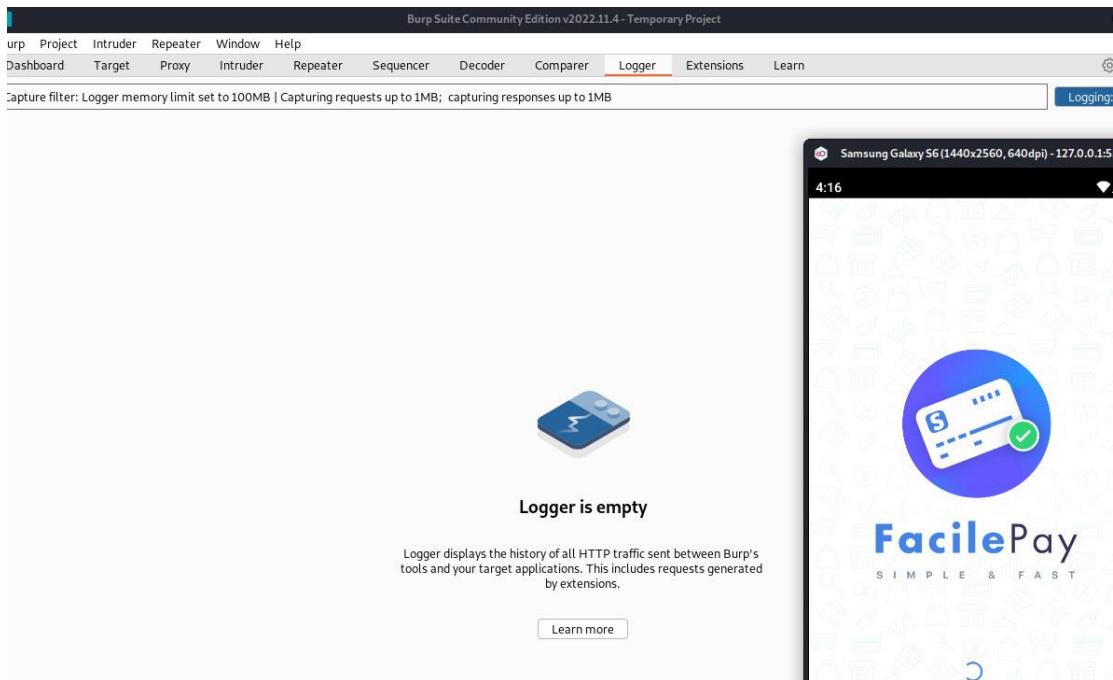


Now each issue inside that activity will be shown on burpsuite:

Issue type	Host
! Suspicious input transformation (reflected)	http://insecure-bank.com /url-shorten
! SMTP header injection	http://insecure-website.... /contact-us
! Serialized object in HTTP message	http://insecure-bank.com /blog
! Cross-site scripting (DOM-based)	https://insecure-bank.com /
! XML external entity injection	https://vulnerable-websi... /product/stock
! External service interaction (HTTP)	https://insecure-website... /product
! Web cache poisoning	http://insecure-bank.com /contact-us
! Server-side template injection	http://insecure-bank.com /user-homepage
! SQL injection	https://vulnerable-websi... /
! OS command injection	https://insecure-website... /feedback/submit

it detects all of the issues.

And it also saves the cache data of the different sites used by the activity inside the cache memory:



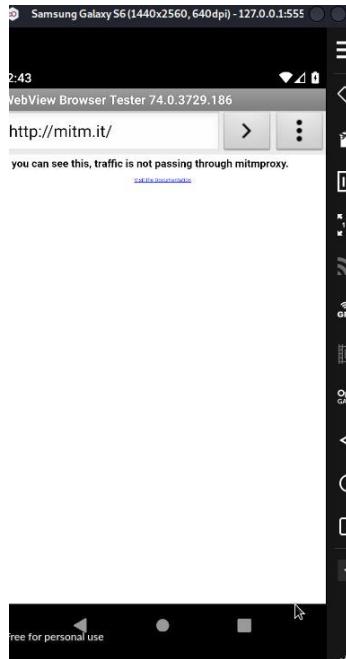
It shows inside its event log, information regarding the proxy and the device it is running on.

Event log			
Filter	Critical	Error	Info
Time	Type	Source	Message
16:00:07 13 Dec 2022	Info	Proxy	Proxy service started on 192.168.1.109:8080
15:54:41 13 Dec 2022	Info	Proxy	Proxy service started on 127.0.0.1:8080

3. Dynamic analysis with MITMproxy:

Mitmproxy starts as a [regular HTTP proxy](#) by default and listens on <http://localhost:8080>.

First things first, we will launch the mobile emulator without connecting it to the proxy and check the mitm.it site to see if the device is connected or not.

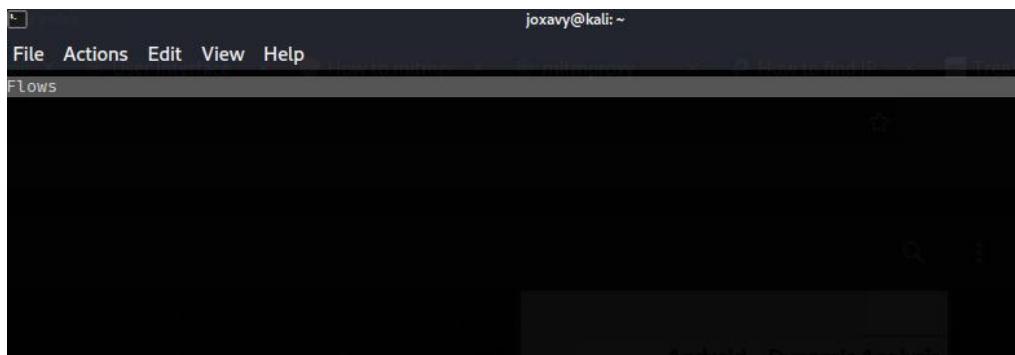


The result shows that the device's traffic is not passing through the proxy.

I will now check the command line interface of the mitmproxy by typing the following command:

```
File Actions Edit View
(joxavy㉿kali)-[~]
$ mitmproxy -p 8881
```

The result:



As you can see, no traffic is being registered.

Same thing appears on the GUI interface of the proxy:

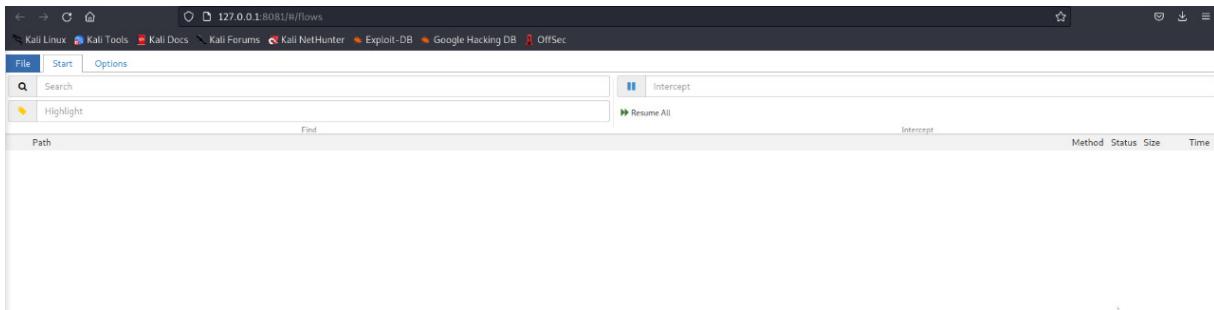
Running it:

```

File Actions Edit View Help
(joxavy㉿kali)-[~]
$ mitmproxy -p 8881
(joxavy㉿kali)-[~]
$ mitmproxy -p 8881
(joxavy㉿kali)-[~]
$ mitmweb
Web server listening at http://127.0.0.1:8081/
Proxy server listening at *:8080

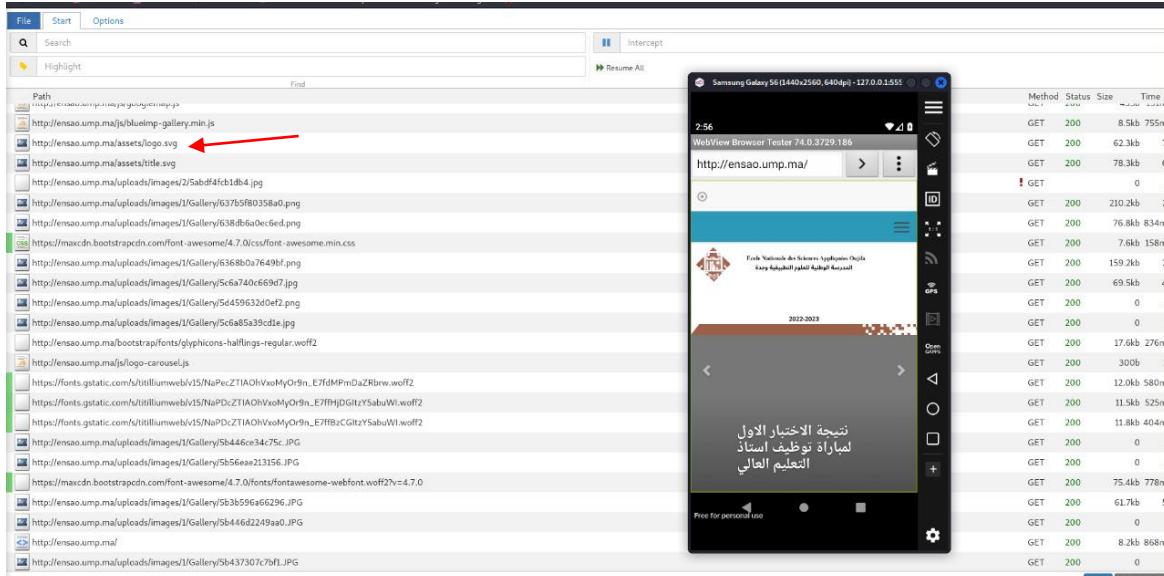
```

Checking the interface on Browser:



I will then configure the proxy on the mobile emulator by using my device's IP address as server name and the port number 8080 that the proxy server is listening to as a port.

Then I will navigate through the phone to different websites as seen below:



As you can see above, the traffic is being caught on the mitmproxy's interface.

They're also being caught under the command:

```
(joxavy㉿kali)-[~]
$ mitmweb
Web server listening at http://127.0.0.1:8081/
Proxy server listening at *:8080
192.168.1.109:44722: client connect
192.168.1.109:44722: client disconnect
192.168.1.109:44724: client connect
192.168.1.109:44724: client disconnect
192.168.1.109:44726: client connect
192.168.1.109:44726: client disconnect
192.168.1.109:44728: client connect
192.168.1.109:44728: server connect www.google.com:443 (142.250.200.196:443)
192.168.1.109:44730: client connect
192.168.1.109:44730: server connect fonts.gstatic.com:443 (142.250.201.3:443)
192.168.1.109:44732: client connect
192.168.1.109:44734: client connect
192.168.1.109:44736: client connect
192.168.1.109:44738: client connect
192.168.1.109:44740: client connect
192.168.1.109:44742: client connect
192.168.1.109:44744: client connect
192.168.1.109:44746: client connect
192.168.1.109:44748: client connect
192.168.1.109:44752: server connect www.gstatic.com:443 (172.217.19.150:443)
192.168.1.109:44752: server connect play.google.com:443 (142.250.200.206:443)
192.168.1.109:44758: client connect
192.168.1.109:44758: error establishing server connection: [Errno -5] No address associated with hostname
192.168.1.109:44756: client disconnect
192.168.1.109:44756: client connect
192.168.1.109:44758: client connect
192.168.1.109:44758: error establishing server connection: [Errno -2] Name or service not known
192.168.1.109:44758: client disconnect
192.168.1.109:44756: client disconnect
192.168.1.109:44756: error establishing server connection: client disconnected
192.168.1.109:44760: client connect
192.168.1.109:44760: client disconnect
```

Checking the same website again that is mitm.it to see if the device is connected:



Information is also displayed in the command line interface of mitmproxy:

```

joxavy@kali:~$ ./NetworkMiner -r /root/Desktop/NetworkMiner.pcap
[...]
Flows
>>15:10:13 HTTPS GET www.google.com /m?hl=en&source=android-launcher-widge... 200 text/html 112k 1.36s
15:10:14 HTTPS POST www.google.com /gen_204?s=web&t=aft&atyp=csi&ei=JX-bY... 204 [no content] 215ms
15:10:14 HTTPS GET www.google.com /xjs/_/js/k=xjs.qs.en_GB.GBhQRIdTxEA.O... 200 ...ntent missing]
15:10:15 HTTPS GET www.google.com /xjs/_/js/k=xjs.qs.en_GB.GBhQRIdTxEA.O... 200 ...ntent missing]
15:10:19 HTTP GET ensao / err ..vice not known
15:10:19 HTTP GET ensao /favicon.ico err ..vice not known

```

By clicking on any link, we get diverse information:

Request info:

```

File Actions View Help
File Details
202-12-15 15:14:49 GET http://ensao.ump.ma/
    < 200 OK text/html 8.1k 610ms
        Request Response Detail
Host: ensao.ump.ma
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 ((Linux; Android 10; Galaxy S6 Build/QQID.200105.002; rv: ) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/74.0.3729.186 Mobile Safari/537.36
Accept: image/webp,image/apng,image/*,*;q=0.8
Referer: http://ensao.ump.ma/
Content-Encoding: deflate
Content-Language: en-US,ar;q=0.9
Cookie: _ga=GA1.2.1187313986.1671135230; _gat=1; XSRF-TOKEN=eyJzdjI6InhhdZyWhcU0lKtzlYHMAuUFaxREE9PSisInZhblvIi5oiZpLdf43vnFZS0Fjd2lVZG5Rjz5dnROMpPVDJVVNrb3dInFlnc3JpUmh1RkdKVHtcTQ2c5dK0JFWlhza1dQmHDZC2ewhlmhMhRelizVhSm3c9PSisIm1hYy161j4c42jfhNG0zZm1M2Y1Zm2Q4Zm12MwMfdkz3m1MzRmTm5z10M3QwOvhZAJN1g4MwZjNmWVt1fQAJ0K3D; ensao_session=eyJpd1bIKk2ZG1QltMVGSEcWlhQ3BkUXgrUEH40RMm01EW02KRNKyy1i7Q3DmSD
Requested-With: org.chromium.webview_shell
Request content

```

Response info:

```

Flow Details
>>15:14:49 GET http://ensao.ump.ma/
    < 200 OK text/html 8.1k 610ms
        Request Response Detail
Date: Thu, 15 Dec 2022 20:14:49 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Frame-Protector: 11 mode-block
Content-Type: text/html; charset=UTF-8
Content-Security-Policy: default-src 'self'; img-src 'self' 'unsafe-inline'; style-src 'self'; script-src 'self' 'unsafe-inline'; connect-src 'self';
Content-Origin: http://ensao.ump.ma
Content-Transfer-Encoding: binary
Content-Length: 8160
Content-Encoding: gzip
Content-Security-Policy: max-age=31536000; includeSubDomains; preload
Content-SecurePolicy: default-src 'self'; img-src 'self'; style-src 'self'; font-src 'self'; script-src 'self'; 'unsafe-inline'; connect-src 'self';
Content-Type: text/html; charset=UTF-8
[decoded gzip] HTML
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>École Nationale des Sciences Appliquées d'Oran</title>
        <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,200,100,1001,400,4001,6001,700,7001,900&subset=latin-ext" rel="stylesheet">
        <link href="https://fonts.googleapis.com/css?family=Changa+One:subset=arabic" rel="stylesheet">
        <link href="https://fonts.googleapis.com/css?family=Changa+One:subset=arabic" rel="stylesheet">
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.7.0/css/font-awesome.min.css">
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.7.0/css/bootstrap.min.css">
        <link rel="stylesheet" href="https://ensao.ump.ma/bootstrap/css/bootstrap.min.css">
        <link custom="true" for="menu" type="text/css" href="http://ensao.ump.ma/plugins/mega-menu/css/smartermenus.css">
        <link rel="stylesheet" type="text/css" href="http://ensao.ump.ma/plugins/mega-menu/css/mega-menu.css">
        <link href="https://ensao.ump.ma/css/carsus.css" rel="stylesheet">
        <link href="https://ensao.ump.ma/css/lightbox-gallery.min.css" rel="stylesheet">
        <link rel="stylesheet" type="text/css" href="http://ensao.ump.ma/css/main.css">
    </head>
    <body>
        <header>
            <nav class="navbar navbar-default stroke">
                <div class="container">
                    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                        <span class="fa fa-dot-circle-o" style="font-size: 18px;"></span>
                    </button>
                </div>
                <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                    <ul class="nav navbar-nav">
                        <li class="nav-item">
                            <a href="https://messagerie.ump.ma" target="_blank">Messagerie (AMP)</a>
                        </li>
                        <li class="nav-item" target="_blank">

```

Detail info:

```

Flow Details
2022-12-15 15:18:18 POST https://www.instagram.com/logging/falco HTTP/2.0
  < 200 application/json 150 1.26s
Request
Server Connection:
  Address www.instagram.com:443
  Resolved Address 157.246.212.174:443
  HTTP Version HTTP/2.0
  ALPN h2
Server Certificate:
  Type RSA, 2048 bits
  SHA256 digest f2 f2 86 43 14 31 77 fd d3 49 9b 20 4e ce ba f5 90 5b 2e 3f a1 fe 27 55 c9 61 ce 4c 00 08 eb 97
  Valid from 2022-12-24 00:00:00+00:00
  Valid to 2023-12-23 23:59:59+00:00
  Serial 1322646411486139018647641782605203814
  Subject C US
    ST California
    L Menlo Park
    O Facebook, Inc.
    CN *.www.instagram.com
  Issuer C US
    O DigiCert Inc
    OU www.digicert.com
    CN DigiCert SHA2 High Assurance Server CA
    *.www.instagram.com, www.instagram.com
Alt names
  Client Connection:
    Address 192.168.1.109:44964
    HTTP Version HTTP/2.0
    TLS Version TLSv1.3
    Server Name Indication www.instagram.com
    Cipher Name TLS_AES_256_GCM_SHA384
    ALPN h2
Timing:
  Client conn. established 2022-12-15 15:18:00.157
  Server conn. initiated 2022-12-15 15:18:00.165
  Server conn. TCP handshake 2022-12-15 15:18:05.315
  Server conn. TLS handshake 2022-12-15 15:18:05.420
  Client conn. TLS handshake 2022-12-15 15:18:05.434
  First request byte 2022-12-15 15:18:18.407
  Request complete 2022-12-15 15:18:18.411
  First response byte 2022-12-15 15:18:19.686
  Response complete 2022-12-15 15:18:19.690
  Client conn. closed -
  Server conn. closed -

```

After connecting my mobile device (not the emulator) to the mitmproxy, and trying to navigate through websites, I get the following result on the proxy command line interface:

```
[ 50/441]
Warn: 192.168.1.103:48398: Client TLS handshake failed. The client does not trust the proxy's certificate for update.googleapis.com (OpenSSL Error([["SSL routines", "", "sslv3 alert certificate unknown"]]))
```

After launching the app on the emulator, the information also gets displayed in the proxy:

Path	Method	Status	Size	Time	Request	Response	Error	Connection	Timing
https://cdn.apimonk.com/logos/com.mobilwik_new_80x80.png	GET	200	3.44B	228ms					
https://cdn.apimonk.com/logos/com.paypal.android.p2pmobile_80x80.png	GET	200	2.2B	243ms					
https://cdn.apimonk.com/logo/com.mercadopago.wallet_80x80.png	GET	200	5.0B	320ms					
https://cdn.apimonk.com/logo/com.squareup.cash_80x80.png	GET	200	3.7B	312ms					
https://apk.apimonk.com/apkD7-7com.stripe.credit.card.payment_2022-10-07.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=IYVHACU06eQ56m9L5ZcF2e222121%2fus-east-1%2f33%2faws4-request&X-Amz-Date=20221215T204295Z&X-Amz-Expires=2408x-Amz-SignedHeaders=host&X-Amz-Signature=99185e937974cd277fcf459f4ac93r5ed5d5707e1f6f1690d8662e354d2440 HTTP/2.0	GET	200	0	-					
upgrade-insecure-requests: 1									
user-agent: Mozilla/5.0 (Linux; Android 10; Galaxy S6 Build/QQID.209105.002; rv: 60.0) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/74.0.3729.108 Mobile Safari/537.36									
v=83									
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3									
referer: https://www.apimonk.com/download/app/com.stripe.credit.card.payment_7com.stripe.credit.card.payment_2022-10-07.apk/									
/7 com.stripe.credit.card.payment_2022-10-07.apk									
accept-encoding: gzip, deflate									
accept-language: en-US,en;q=0.9									
cookie:_ga=GA1.2.1221871520.1670964295									
cookie:_gads=ID:0b1e2b9ce22af-22299656dc0706b0:RT:1670964300:S=ALNI_MQ0B3zaBU19reuhHydICuUnmekuyCw									
cookie:_gid=GAI.2.1493947311.1671136188									
cookie:_gpl1=U1D:e00000bab77fa0f:T=1670964300:RT=1671136192:S=ALNI_M2z0ZcxzvTEEy0Ba9B1172_Qr1aIQ									
cookie:FCNEC-X3B85BX22AKs501-40Xb2G7yPRAK1x4o/xz1Q0Q4z12EwUojoPnR7jWkJ-1gGLURhnx7Y7YNNzwyvpQ289B_Z0ggperryDwVcPa4ujb0d0RxB906IBWsgY92pR4-v0j-Tty-VtwJygbJez47hfFdrwn_-uB10Q5Zw%3D%3D									
N225SDN2nu112C058p05NSD									
x-requested-with: org.chromium.webview_shell									
Query									
X-Amz-Algorithm: AWS4-HMAC-SHA256									
X-Amz-Credential: IYVHACU06eQ56m9L5ZcF2e222121%2fus-east-1%2f33%2faws4-request									
X-Amz-Date: 20221215T204295Z									
X-Amz-Expires: 2408									
X-Amz-SignedHeaders: host									
X-Amz-Signature: 99185e937974cd277fcf459f4ac93r5ed5d5707e1f6f1690d8662e354d2440									

It includes different information about it including the algorithm used, the certificate date, the expiry date, etc.

4. Dynamic analysis with Frida:

Frida is a free and open-source dynamic code instrumentation toolkit that lets you execute snippets of JavaScript into native apps on Android and iOS (as well as on other platforms).

First, let's check which emulators are connected:

```
(joxavy㉿kali)-[~]
$ adb devices
List of devices attached
127.0.0.1:5555    device
```

Installing Frida and Frida tools:

```
(joxavy㉿kali)-[~]
$ pip install frida
Defaulting to user installation because normal site-packages is not writeable
Collecting frida
  Downloading frida-16.0.8-cp37-abi3-manylinux_2_5_x86_64.manylinux1_x86_64.whl (19.2 MB)
    19.2/19.2 MB 588.1 kB/s eta 0:00:00
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from frida) (65.5.0)
Installing collected packages: frida
Successfully installed frida-16.0.8

(joxavy㉿kali)-[~]
$ pip install frida-tools
Defaulting to user installation because normal site-packages is not writeable
Collecting frida-tools
  Downloading frida-tools-12.0.4.tar.gz (177 kB)
    177.6/177.6 kB 466.0 kB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: frida<17.0.0,>=16.0.0 in ./local/lib/python3.10/site-packages (from frida-tools) (16.0.8)
Requirement already satisfied: pygments<3.0.0,>=2.0.2 in /usr/lib/python3/dist-packages (from frida-tools) (2.7.1)
Requirement already satisfied: prompt-toolkit<4.0.0,>=2.0.0 in ./local/lib/python3.10/site-packages (from frida-tools) (3.0.19)
Requirement already satisfied: colorama<1.0.0,>=0.2.7 in /usr/lib/python3/dist-packages (from frida-tools) (0.4.4)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from frida<17.0.0,>=16.0.0→frida-tools) (65.5.0)
Requirement already satisfied: wcwidth in /usr/lib/python3/dist-packages (from prompt-toolkit<4.0.0,>=2.0.0→frida-tools) (0.1.9)
Building wheels for collected packages: frida-tools
  Building wheel for frida-tools (pyproject.toml) ... done
  Created wheel for frida-tools: filename=frida_tools-12.0.4-py3-none-any.whl size=187228 sha256=2f6516192ab59042d ab471b04947c2a0a70d3de69e24a944e7562f3010dc2b1a
  Stored in directory: /home/joxavy/.cache/pip/wheels/1b/5f/95/ff066ec78a422d8577494f72e0e5b0b06f7d442896b7bcfaf4
Successfully built frida-tools
Installing collected packages: frida-tools
WARNING: The scripts frida, frida-apk, frida-compile, frida-create, frida-discover, frida-join, frida-kill, frida-ls, frida-ls-devices, frida-ps, frida-pull, frida-push, frida-rm and frida-trace are installed in '/home/joxavy/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed frida-tools-12.0.4

(joxavy㉿kali)-[~]
```

Entering the adb shell:

```
(joxavy㉿kali)-[~]
$ adb shell
vbox86p:/ # cd /data
vbox86p:/data # ls
adb      app-private  drm       misc_ce   property          system_ce  vendor_ce
anr      app-staging   gsi       misc_de   resource-cache   system_de  vendor_de
apex     backup        local     nfc      rollback         tombstones
app      bootchart    lost+found ota      rollback-observer unencrypted
app-asec  cache        media    ota_package server_configurable_flags user
app-ephemeral dalvik-cache mediadm  per_boot ss           user_de
app-lib   data         misc    preload   system          vendor
```

Placing ourselves inside the sdcard/downloads in order to retrieve the compressed file for the Frida server and copying it into the new path “data/local/tmp”:

```

2|vbox86p:/ # cd sdcard
vbox86p:/sdcard # ls
Alarms Android DCIM Download Movies Music Notifications Pictures Podcasts Ringtones
vbox86p:/sdcard # cd Download
vbox86p:/sdcard/Download # ls
frida-server-16.0.8-android-x86.xz mitmproxy-9.0.1-linux.tar.gz
vbox86p:/sdcard/Download # tar -xf frida-server-16.0.8-android-x86.xz
tar: invalid tar format
1|vbox86p:/sdcard/Download # unxz frida-server-16.0.8-android-x86.xz
vbox86p:/sdcard/Download # ls
frida-server-16.0.8-android-x86 mitmproxy-9.0.1-linux.tar.gz
vbox86p:/sdcard/Download # mv frida-server-16.0.8-android-x86 /data/local/tmp
vbox86p:/sdcard/Download # cd /data/local/tmp
vbox86p:/data/local/tmp # ls

```

Renaming it and giving it execution permissions, then running it:

```

12|vbox86p:/data/local/tmp # rm frida-server
vbox86p:/data/local/tmp # mv frida-server-16.0.8-android-x86 frida-server
vbox86p:/data/local/tmp # ./frida-server
/system/bin/sh: ./frida-server: can't execute: Permission denied
126|vbox86p:/data/local/tmp # chmod 755 frida-server
vbox86p:/data/local/tmp # ./frida-server
■

```

With Frida-server running, you should now be able to get a list of running processes with the following command :

PID	Name	Identifier
4		
3769	Amaze	com.amaze.filemanager
2619	FacilePay	com.stripe.credit.card.payment
3635	Files	com.android.documentsui
2459	Gallery	com.android.gallery3d
2195	Phone	com.android.dialer
3103	Search	com.android.quicksearchbox
2569	Settings	com.android.settings
3407	WebView Shell	org.chromium.webview_shell
-	Calendar	com.android.calendar
-	Camera	com.android.camera2
-	Clock	com.android.deskclock
-	Contacts	com.android.contacts
-	Custom Locale	com.android.customlocale2
-	Development Settings	com.android.development_settings
-	Email	com.android.email
-	Messaging	com.android.messaging
-	Superuser	com.genymotion.superuser

This will show the names and identifiers of all apps, if they are currently running it will also show their PIDs. As we can see amaze has a PID while Clock doesn't.

The PID of my application (FacilePay)in this list is 2619(it can change)

Running the following command with the PID of the app:

```

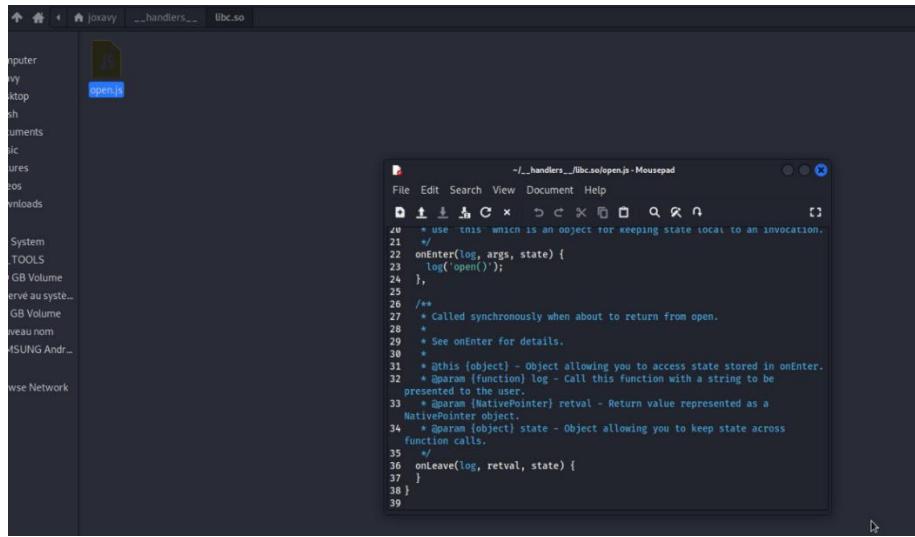
└─(joxavy㉿kali)-[~]
$ frida-trace -U 2619 -i "open"
Instrumenting ...
open: Auto-generated handler at "/home/joxavy/_handlers__/libc.so/open.js"
Started tracing 1 function. Press Ctrl+C to stop.
■

```

This generates a little JavaScript in _handlers__/libc.so/open.js, which Frida injects into the process.

The script traces all calls to the open function in libc.so.

Here is the script that has been generated:



The tracing result:

```
(joxavy㉿kali)-[~]
└─$ frida-trace -O 2619 -i "open"
Instrumenting ...
open: Auto-generated handler at /home/joxavy/_handlers__/_libc.so/open.js"
Started tracing 1 function. Press Ctrl+C to stop.

281602 ms open()
281605 ms open()
    /* TID 0xa50 */
281612 ms open()
    /* TID 0x1116 */
281618 ms open()
    /* TID 0xa50 */
281619 ms open()
    /* TID 0x1116 */
281703 ms open()
281833 ms open()
281833 ms open()
281837 ms open()
    /* TID 0xa3b */
281917 ms open()
    /* TID 0x1116 */
281925 ms open()
    /* TID 0xa3b */
281943 ms open()
281987 ms open()
282019 ms open()
    /* TID 0x1116 */
282026 ms open()
    /* TID 0xa3b */
282037 ms open()
    /* TID 0x1116 */
282126 ms open()
282228 ms open()
282232 ms open()
    /* TID 0xa3b */
282233 ms open()
    /* TID 0x1116 */
282239 ms open()
    /* TID 0xa3b */
282250 ms open()
    /* TID 0x1116 */
282444 ms open()
283198 ms open()
283201 ms open()
283216 ms open()
283468 ms open()
283609 ms open()
    /* TID 0xa61 */
283741 ms open()
    /* TID 0x1116 */
285938 ms open()
285934 ms open()
285936 ms open()


No Customers
Add and Save Customers to sale items
and future records

```

Use the Frida CLI tool (`frida`) to work with Frida interactively. It hooks into a process and gives you a command line interface to Frida's API.

```

└─(root㉿kali)-[~/home/joxavy]
# frida -U 2619

      /_|
| ( )| Frida 16.0.8 - 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 A50 (id=127.0.0.1:5555)

[A50::PID::2619 ]→ █

```

The JADX decompiler can generate Frida snippets through its graphical code browser.

To use this feature, open the APK or DEX with jadx-gui, browse to the target method, right click the method name, and select "Copy as frida snippet (f)".

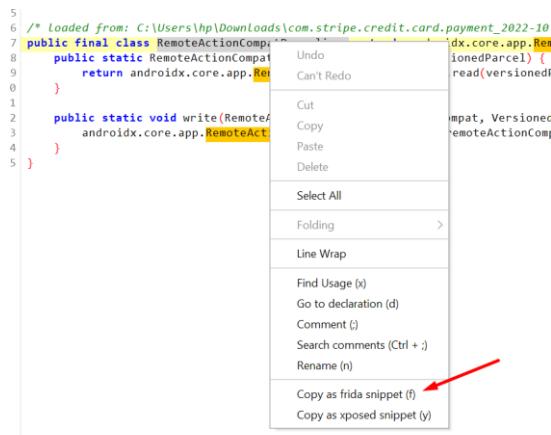


```

1 package android.support.v4.app;
2
3 import androidx.core.app.RemoteActionCompat;
4 import androidx.versionedparcelable.VersionedParcel;
5
6 /* renamed from: android.support.v4.app.RemoteActionCompatParcelizer */
7 /* loaded from: classes.dex */
8 public final class RemoteActionCompatParcelizer extends androidx.core.app.RemoteActionCompatParcelizer {
9     public static RemoteActionCompat read(VersionedParcel versionedParcel) {
10         return androidx.core.app.RemoteActionCompatParcelizer.read(versionedParcel);
11     }
12
13     public static void write(RemoteActionCompat remoteActionCompat, VersionedParcel versionedParcel) {
14         androidx.core.app.RemoteActionCompatParcelizer.write(remoteActionCompat, versionedParcel);
15     }
16 }

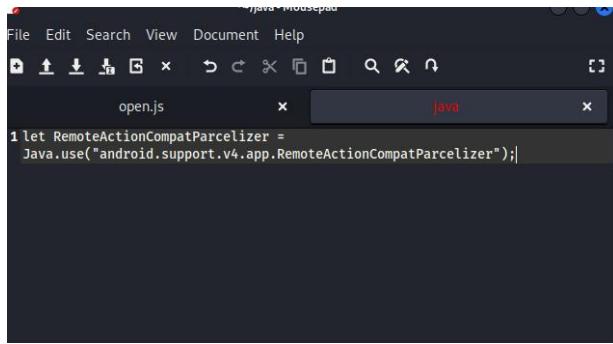
```

Copying the snippet:



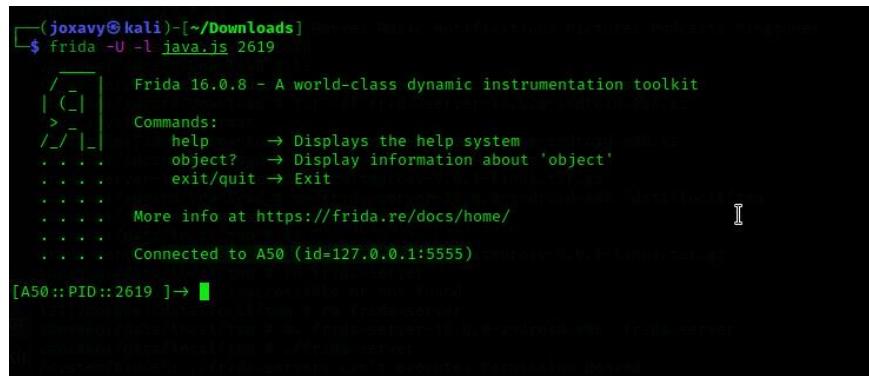
The above steps place the following output in the pasteboard, which you can then paste in a JavaScript file and feed into frida -U -l.

Pasting the result into a JS file:



```
File Edit Search View Document Help
open.js      x      java      x
let RemoteActionCompatParcelizer =
Java.use("android.support.v4.app.RemoteActionCompatParcelizer");
```

Feeding it into Frida:



```
(joxavy㉿kali)-[~/Downloads]
$ frida -U -l java.js 2619

    / \   |  Frida 16.0.8 - 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 A50 (id=127.0.0.1:5555)

[A50 :: PID::2619 ] →
```

Frida also lets you search for and work with instantiated objects that are on the heap. The following script searches for instances of android.view.View objects and calls their `toString` method. The result is printed to the console:

```
setImmediate(function() {
  console.log("[*] Starting script");

  Java.perform(function () {
    Java.choose("android.view.View", {
      "onMatch":function(instance){
        console.log("[*] Instance found: " + instance.toString());
      };
      "onComplete":function() {
        console.log("[*] Finished heap search");
      };
    });
  });
});
```

Feeding it into frida:

```
[joxavy@kali:~] -[~]
$ frida -U -l /home/joxavy/Downloads/myscript.js 2225
[✓] Frida 16.0.6 - 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 A80 (id:127.0.0.1:5555)
[Attaching...]
[*] Starting script
[AS0*:PID:2225] → Process crashed: Trace/BPT trap

[*]
[*] x86
[*] Build Fingerprint: "google/vbox80xp/vbox80xp:10/QQ10.200105.002/475:userdebug/test-keys"
[*] Revision: '0'
[*] RAM: 8GB
[*] Timestamp: 2022-12-26 08:27:30-0800
[*] pid: 2225, tid: 2679, name: Thread-10 >>> com.stripe.credit.card.payment <<<
[*] uid: 10101
[*] signum: 10 (SIGABRT)
[*] code: -1 (SI_QUEUE), fault addr _____
[*] User input: "M1 DETECTED ERROR IN APPLICATION: use of invalid jobkey @+ee2e18"
[*] eip: 00000000 ebx: 00000001 ecx: 00000007 edx: 00000006
[*] edi: 718456 esi: b0d4bf0f
[*] ebp: 00000000 esp: 00000000
[*] flags: [KERNAL]

[*] Backtrace:
#00 pc 000000b9 [vdso] (_kernel_vsyscall+)
#01 pc 00002328 [/apex/com.android.runtime/lib/bionic/libc.so (syscall+40)] (BuildId: 790c696781f8a5032d633
77a900000000)
#02 pc 00000d5f [/apex/com.android.runtime/lib/bionic/libc.so (offset 1+ad000) (abort+193)] (BuildId: 790c696781f8a5032d633
77a900000000)
#03 pc 00000040 <anonymous:demod7000>
[*]
[*] AS0*:PID:2225 ->
```

5. Dynamic analysis with radare2:

radare2 (r2) is a popular open-source reverse engineering framework for disassembling, debugging, patching and analyzing binaries that is scriptable and supports many architectures and file formats including Android and iOS apps.

For Android, Dalvik DEX (odex, multidex), ELF (executables, .so, ART) and Java (JNI and Java classes) are supported.

It also contains several useful scripts that can help you during mobile application analysis as it offers low level disassembling and safe static analysis that comes in handy when traditional tools fail.

installing radare2:

```
[joxavy@kali:~] -> $ git clone https://github.com/radarorg/radar2
Cloning into 'radar2'...
remote: Enumerating objects: 206822, done.
remote: Counting objects: 100K (6/2/2), done.
remote: Compressing objects: 100K (4/7/2), done.
remote: Writing objects: 100K (2/2/2), done.
Resolving deltas: 100% (206822/206822), 160.24 MiB | 3.05 MiB/s, done.
Resolving deltas: 100% (206891/206891), done.

[joxavy@kali:~] -> $ cd radar2
[joxavy@kali:~/radar2] -> $ ./radar2
WARNING: Updating from remote repository
From https://github.com/radarorg/radar2
          master      →  FETCH_HEAD

Already up to date.
Warning: Your system-wide capstone is too old for me
[*] Finding gnmake is a tracked alias for /usr/bin/gmake OK
[*] Checking out capstone...  OK
[*] Checking out vector35-arm64...  OK
[*] Checking out vector35-armv7...  OK
[*] Running configure...  OK
[*] Running make...  OK
[*] Running make install...  OK
/usr/bin/zsh ~/sys/build.sh
/home/joxavy/radar2

Building on Linux : computing number of allowed parallel jobs.
Number of parallel jobs per job is 158000 KiB.
Number of CPUs is 4 and Current Free RAM allows us to run 40 jobs in parallel.
So, the build will run on 40 jobs.

/home/joxavy/radar2
configure-plugins: Copying dist/plugins-def.cfg/plugins.def.cfg
configure-plugins: Generating libm/config.h
configure-plugins: Generating libm/asm/di/config.inc
configure-plugins: Generating libc/config.mk

STATIC: anal.lso02 anal.8005 anal.alpha anal.armc anal.arm.cs anal.arm_gnu anal.svr anal.anl.anl_bp
        anal.hpc.cs anal.chip8 anal.cri6 anal.cris anal.dvlk anal.libc anal.eval_cv anal_gb anal.h300 anal.hppa_gnu
        anal.i1800 anal.java anal.kvz anal.lanal_gnu anal.180001 anal.3002 anal.loongarch_gnu anal.m6809_cv anal.m6809_cs
        anal.m6809_gnu anal.msp432 anal.nios2 anal.riscv anal.s390 anal.s390_gnu anal.sparc_cv anal.sparc_gnu
        anal.sparc_i386_gnu anal.tms320cv32 anal.tricore_cv anal.v850 anal.vax anal.wave anal.x86_cv anal.xcore_cv anal
        xtensa_cv anal.z80_cv arch.ad59k arch.and92k arch.any_v3m arch.arch_arcs arch.arch_i486
        arch.i386_riscv arch.i386_riscv arch.m32r arch.m68k arch.x86_64 null_arch bin_arch bin_cv
        bin_i386 bin_i386_cv bin_i386_riscv bin_i386_riscv_cv bin_i386_riscv_i386 bin_i386_riscv_i386_cv
        bin_i386_riscv_i386_cv_cv bin_i386_riscv_i386_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
        bin_i386_riscv_i386_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv_cv
```

The radare2 framework comprises a set of small utilities that can be used from the r2 shell or independently as CLI tools. These utilities include rabin2, rasm2, rahash2, radiff2, rafind2, ragg2, rarun2, rax2, and of course r2, which is the main one.

For example, you can use rafind2 to read strings directly from an encoded Android Manifest (AndroidManifest.xml):

For permission:

```
[joxavy@kali:~/Downloads]$ cd com.stripe.credit.card.payment_2022-10-07
[joxavy@kali:~/Downloads/com.stripe.credit.card.payment_2022-10-07]$ ls
AndroidManifest.xml assets lib  original small  small_classes3
apktool.yml kotlin META-INF res  small_classes2  unknown
[joxavy@kali:~/Downloads/com.stripe.credit.card.payment_2022-10-07]$ rafind2 -zs permission AndroidManifest.xml
0x131
0x13d
0x13f
0x142
0x1cf
0x1f0
0x1f2
0x233
0x29c
0x29d
0x2a0
0x2a5
0x2a1
0x2a1
0x222
0x343
0x364
0x371
0x3b2
0x3d1
0x3f2
0x406
0x42f
0x44f
0x470
0x49f
0x4c0
0x4e0
0x50f
0x535
0x556
0x575
0x59d
0x5c8
0x7b9
```

For activities:

```
[joxavy@kali:~/Downloads/com.stripe.credit.card.payment_2022-10-07]$ rafind2 -zs Activity AndroidManifest.xml
0xdb5
0xe40
0x939
0x1103
0x117b
0x11f6
0x1272
0x1338
0x1362
0x13db
0x147f
0x1516
0x1556
0x16fe
0x179a
0x183a
0x1800
0x196
0x1a16
0x1ab6
0x1b56
0x1c10
0x1f33
0x2187
0x2257
0x22f6
0x2350
0x243d
0x24db
0x2580
0x2618
0x272
0x2827
0x28d0
0x2976
0x2a35
0x2a87
0x2b68
0x2c0b
0x2c0b
0x2d50
0x2d5b
0x2e99
```

Using rabin2 to get information about a binary file for classes.dex :

For classes2.dex:

```
[joxavy㉿kali] -[~/Downloads/com.stripe.credit.card.payment_2022-10-07.apk_FILES]
$ rabin2 -i classes2_dex
[Imports]
nth vaddr      bind type lib name

0    0x000d0a74 NONE FUNC      Landroid/animation/Animator.method.addListener(Landroid/animation/Animator$Animator
Listener;)V
1    0x000d0a7c NONE FUNC      Landroid/animation/Animator.method.cancel()V
2    0x000d0a84 NONE FUNC      Landroid/animation/Animator.method.end()V
3    0x000d0a8c NONE FUNC      Landroid/animation/Animator.method.setDuration(J)Landroid/animation/Animator;
4    0x000d0a94 NONE FUNC      Landroid/animation/Animator.method.setStartDelay(J)V
5    0x000d0a9c NONE FUNC      Landroid/animation/Animator.method.start()V
6    0x000d0aa4 NONE FUNC      Landroid/animation/AnimatorListenerAdapter.method.<init>()V
7    0x000d0aae NONE FUNC      Landroid/animation/AnimatorListenerAdapter.method.onAnimationCancel(Landroid/animat
ion/Animator;)V
8    0x000d0ab4 NONE FUNC      Landroid/animation/AnimatorListenerAdapter.method.onAnimationEnd(Landroid/animat
ion/Animator;)V
9    0x000d0abc NONE FUNC      Landroid/animation/AnimatorListenerAdapter.method.onAnimationRepeat(Landroid/animat
ion/Animator;)V
10   0x000d0ac4 NONE FUNC      Landroid/animation/AnimatorListenerAdapter.method.onAnimationStart(Landroid/animati
on/Animator;)V
11   0x000d0acc NONE FUNC      Landroid/animation/AnimatorSet$Builder.method.with(Landroid/animation/Animator;)Lan
droid/animation/AnimatorSet$Builder;
12   0x000d0ad4 NONE FUNC      Landroid/animation/AnimatorSet.method.<init>()V
13   0x000d0adc NONE FUNC      Landroid/animation/AnimatorSet.method.addListener(Landroid/animation/Animator$Anima
torListener;)V
```

For libopencv library:

```
[+] $ ./libandroid.so libcardioRecognizer.so libopenvc_core.so libcardioCalculator.so libcardioRecognizer_tetra2_imgorpc.so

[+] $ ./libopenvc_core.so
[Cannot determine entrypoint, using <0x002c200>.

[Imports]
    nth vaddr bind type lib name
1 0x0002c1e0 GLOBAL FUNC dl_iterate_phdr
2 0x0002b040 GLOBAL FUNC __cxa_finalize
3 0x0002b050 GLOBAL FUNC __cxa_savexit
4 0x00000000 GLOBAL FUNC __stack_chk_fail
5 0x0002b060 GLOBAL FUNC __stack_chk_guard
6 0x0002b070 GLOBAL FUNC strlch
60 0x0002b060 GLOBAL FUNC strcmp
62 0x0002b090 GLOBAL FUNC lrint
121 0x0002b0f0 GLOBAL FUNC fprintf
125 0x0002b0f0 GLOBAL FUNC fflush
126 0x0002b0f0 GLOBAL FUNC fputc
137 0x0002b0f0 GLOBAL FUNC pthread_init
140 0x00000000 WEAK FUNC pthread_create
143 0x0002b0f0 GLOBAL FUNC pthread_mutex_destroy
145 0x0002b0e0 GLOBAL FUNC free
150 0x0002b0e0 GLOBAL FUNC malloc
162 0x0002b0e0 GLOBAL FUNC clock_gettime
166 0x0002b0e0 GLOBAL FUNC vsprintf
173 0x0002b0e0 GLOBAL FUNC getenv
177 0x0002b0e0 GLOBAL FUNC remove
180 0x0002b0e0 GLOBAL FUNC munmap
182 0x0002b0e0 GLOBAL FUNC close
224 0x0002b1f0 GLOBAL FUNC pthread_mutex_unlock
```

Use the main r2 utility to access the r2 shell. You can load DEX binaries just like any other binary:

```
[joxavy@kali:~/Downloads/com.stripe.credit.card.payment_2022-10-07.apk$ FILES]
$ r2 classes.dex
WARNING: No calling convention defined for this file, analysis may be inaccurate.
0x000cdca0> ls
play-services-measurement-impl.properties    firebase-abt.properties   firebase-crashlytics.properties   stamp-cert-
stamp-certs
classes2.dex   firebase-iid-interop.properties   firebase-encoders.properties   protolite-well-known-types.pro-
properties
play-services-measurement-sdk.properties   firebase-analytics.properties   AndroidManifest.xml   play-services-meas-
urement-base.properties
firebase-firebase.properties   kotlin/   res/   firebase-encoders-JSON.properties
firebase-core.properties   play-services-identity.properties   firebase-installations.properties   transport-api.pro-
perties
play-services-measurement.properties   firebase-database-collection.properties   play-services-wallet.properties
play-services-cloud-messaging.properties   firebase-messaging.properties   play-services-ads-identifier.properties   play-services-meas-
urement-api.properties
classes.dex.cache/   firebase-messaging.properties   play-services-ads-identifier.properties   play-services-meas-
urement-api.properties
Firebase-auth.properties   .. / firebase-auth-interop.properties   play-services-maps.properties
play-services-measurement-sdk-api.properties   play-services-tasks.properties   firebase-encoders-proto.properties
okhttp3/
firebase-installations-interop.properties   transport-backend-cct.properties   play-services-auth-base.properties
lib/
play-services-stats.properties   google/   firebase-perf.properties   firebase-common.properties
billing.properties   firebase-measurement-connector.properties   classes3.dex   DebugProbeskt.bin
META-INF/   play-services-auth-api-phone.properties   org/   play-services-basement.property
com/   play-services-auth.properties   firebase-components.properties   classes.dex
Firebase-datatransport.properties   ./ play-services-cast.properties   resources.arsc
play-services-cast-framework.properties   androidsupportmultidexversion.txt   firebase-appcheck-interop.properties
assets/
transport-runtime.properties   smartcard_list.txt   firebase-annotations.properties   play-services-base.properties
play-services-flags.properties
```

Once in the r2 shell, you can also access functions offered by the other radare2 utilities.

To print all the strings, use rabin2 -Z or the command iz (or the less verbose izq) from the r2 shell.

```
Usage: head / [file]
[0x003cd430]> izq
Do you want to print 30723 lines? (y/N) y
0x607634 12 12 0000000000000000
0x607646 13 13 0000000000000000
0x60765c 13 13 0000000000000000
0x607672 16 16 0000000000000000
0x60768d 16 16 0000000000000000
0x6076a8 16 16 0000000000000000
0x6076c3 16 16 0000000000000000
0x6076de 16 16 0000000000000000
0x6076f9 16 16 0000000000000000
0x607714 15 15 0000000000000000
0x60772b 15 15 0000000000000000
0x607742 15 15 0000000000000000
0x607759 15 15 0000000000000000
0x607770 19 19 0000000000000000
0x607790 19 19 0000000000000000
0x6077b0 19 19 0000000000000000
0x6077d0 19 19 0000000000000000
0x6077f0 19 19 0000000000000000
0x607810 19 19 0000000000000000
0x607830 19 19 0000000000000000
0x607850 19 19 0000000000000000
0x60786f 18 18 0000000000000000 shell
0x60788b 18 18 0000000000000000
0x6078a7 17 17 0000000000000000
0x608344 9 9 Found:
0x60834f 48 48
Id Class Name %s State Unique Name Tags
0x608381 59 59
call GlideException#logRootCauses(String) for more detail
0x6083be 24 24
%s %s %s %s %s %s
0x6083d8 10 10
Strategy=
0x6083e4 19 19
There was 1 cause:
0x6083f9 12 12
There were
0x608407 9 9
batch {

0x608412 16 16
event_filter {

0x608424 19 19
property_filter {

0x608443 17 17
View Holder 2:
0x60846a 11 11 values: Ability to switch file (z shell). You can load DEX binaries via
0x608477 11 11 view =
0x608489 21 21 Created new loader
0x6084a0 14 14 Destroying:
0x6084b0 24 24 Filter did not match: Access functions offered by the other m
0x6084ca 27 27 Filter matched! match=0x
0x6084e7 31 31 Filter's target already added
0x608508 6 6 Op #
0x608510 27 27 Re-using existing loader
0x60852d 13 13 Resetting:
0x60853c 12 12 Starting:
0x60854a 12 12 Stopping:
0x608558 11 11 UNKNOWN
0x608565 12 12 filters {
0x608573 8 8 layout
```

Most of the time you can append special options to your commands such as q to make the command less verbose or j to give the output in JSON format (use ~{} to prettify the JSON string):

```
(joxavy㉿kali)-[~/Downloads/com.stripe.credit.card.payment_2022-10-07.apk_FILES]
$ r2 classes2.dex
WARNING: No calling convention defined for this file, analysis may be inaccurate.
[0x0028c5d4]> izj{-}
Do you want to print 339562 lines? (y/N)
[0x0028c5d4]> izj{-}
Do you want to print 339562 lines? (y/N) y
[
  {
    "vaddr": 4990526,
    "paddr": 4990526,
    "ordinal": 2,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990546,
    "paddr": 4990546,
    "ordinal": 3,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990546,
    "paddr": 4990546,
    "ordinal": 4,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990565,
    "paddr": 4990565,
    "ordinal": 5,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990565,
    "paddr": 4990565,
    "ordinal": 6,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990585,
    "paddr": 4990585,
    "ordinal": 7,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  },
  {
    "vaddr": 4990604,
    "paddr": 4990604,
    "ordinal": 8,
    "size": 12,
    "length": 12,
    "section": "data",
    "type": "ascii",
    "string": "\u0000\u0001\u0000\u0000\u0001\u0001\u0001\u0000"
  }
]
```

You can print the class names and their methods with the r2 command ic (information classes):

```
[0x0028c5d4]> ic
Do you want to print 58050 lines? (y/N) y
0x0013e13c [0x001759a4 - 0x00175a2e] 138 class @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C :: Landroid/animation/AnimatorListenerAdapter;
0x001759a4 method 0 C @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C;.method.
<init>(@Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton;Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C;)V
0x001759c4 method 1 p @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C;.method.
onAnimationCancel(Landroid/animation/Animator;)V
0x001759ed method 2 p @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C;.method.
onAnimationEnd(Landroid/animation/Animator;)V
0x00175a1c method 3 p @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$C;.method.
onAnimationStart(Landroid/animation/Animator;)V
0x0013e15c [0x00175a40 - 0x00175adc] 156 class 1 Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D :: Landroid/util/Property;
0x00175a88 method 0 C @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D;.method.
<init>(@java/lang/Class;@java/lang/String;)V
0x00175a40 method 1 p @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D;.method.
a(Landroid/view/View;)Ljava/lang/Float;
0x00175aa0 method 2 p @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D;.method.
b(Landroid/view/View;Ljava/lang/Float;)V
0x00175a68 method 3 ph @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D;.method.
get(@Ljava/lang/Object;)Ljava/lang/Object;
0x00175acc method 4 ph @ Lcom/google/android/material/floatingactionbutton/ExtendedFloatingActionButton$D;.method.
```

You can print the imported methods with the r2 command ii (information imports):

```
[0x0028c5d4]> ii
Do you want to print 4963 lines? (y/N) y
[Imports]
 nth vaddr bind type lib name
 0 0x00000074 NONE FUNC  Landroid/animation/Animator.method.addListener(Landroid/animation/Animator$AnimatorListener;)V
 1 0x0000007c NONE FUNC  Landroid/animation/Animator.method.cancel()V
 2 0x00000084 NONE FUNC  Landroid/animation/Animator.method.end()V
 3 0x0000008c NONE FUNC  Landroid/animation/Animator.method.setDuration(J)Landroid/animation/Animator;
 4 0x00000090 NONE FUNC  Landroid/animation/Animator.method.setStartDelay(J)V
 5 0x0000009c NONE FUNC  Landroid/animation/Animator.method.start()V
 6 0x000000a4 NONE FUNC  Landroid/animation/AnimatorListenerAdapter.method.<init>()V
 7 0x000000ac NONE FUNC  Landroid/animation/AnimatorListenerAdapter.method.onAnimationCancel(Landroid/animation/Animator;)V
 8 0x000000b4 NONE FUNC  Landroid/animation/AnimatorListenerAdapter.method.onAnimationEnd(Landroid/animation/Animator;)V
 9 0x000000bc NONE FUNC  Landroid/animation/AnimatorListenerAdapter.method.onAnimationRepeat(Landroid/animation/Animator;)V
10 0x000000c4 NONE FUNC  Landroid/animation/AnimatorListenerAdapter.method.onAnimationStart(Landroid/animation/Animator;)V
11 0x000000dc NONE FUNC  Landroid/animation/AnimatorSet$Builder.method.with(Landroid/animation/Animator;)Lan
droid/animation/AnimatorSet$Builder;
12 0x000000e4 NONE FUNC  Landroid/animation/AnimatorSet$Builder.method.setDuration(J)V
13 0x000000fc NONE FUNC  Landroid/animation/AnimatorSet$Builder.method.addListener(Landroid/animation/Animator$Anim
atorListener;)V
14 0x000000a4 NONE FUNC  Landroid/animation/AnimatorSet.method.cancel()V
15 0x000000b4 NONE FUNC  Landroid/animation/AnimatorSet.method.end()V
16 0x000000f4 NONE FUNC  Landroid/animation/AnimatorSet.method.isPlaying()Z
17 0x000000fc NONE FUNC  Landroid/animation/AnimatorSet.method.play(Landroid/animation/Animator;)Landroid/anim
ation/AnimatorSet$Builder;
18 0x000000b4 NONE FUNC  Landroid/animation/AnimatorSet.method.playSequentially([Landroid/animation/Animator;
```

- A common approach when inspecting a binary is to search for something, navigate to it and visualize it in order to interpret the code.
 - One of the ways to find something using radare2 is by filtering the output of specific commands, i.e. to grep them using ~ plus a keyword (~+ for case-insensitive).
 - For example, we might know that the app is verifying something, we can inspect all radare2 flags and see where we find something related to "verify".
 - When loading a file, radare2 tags everything it's able to find. These tagged names or references are called flags. You can access them via the command f.
 - In this case we will grep the flags using the keyword "verify":

```
[0x0028c5d4]> f+~verify
[0x0028c5d4]> f+~verify
0x0008269c 0 sym.Lcom_stripe_R_string.sfield_msg_verify_purchase:I
0x00082bb4 0 sym.Lcom_stripe_R_string.sfield_stripe_verify_your_payment:I
0x0008f6bc 0 sym.Lcom_stripe_android_R_string.sfield_stripe_verify_your_payment:I
0x000a1b5c 0 sym.Lcom_stripe_android_payments_R_String.sfield_stripe_verify_your_payment:I
0x000af03c 0 sym.Lcom_stripe_android_paymentsheet_R_string.sfield_stripe_verify_your_payment:I
0x000ca68c 0 sym.Lcom_stripe_model_PaymentIntent_NextAction_VerifyWithMicrodeposits.ifield_arrivalDate:Ljava_lang
_lang
0x000ca694 0 sym.Lcom_stripe_model_PaymentIntent_NextAction_VerifyWithMicrodeposits.ifield_hostedVerificationUrl:
Ljava_lang_String
0x000ca71c 0 sym.Lcom_stripe_model_PaymentIntent_NextAction.ifield_verifyWithMicrodeposits:Lcom_stripe_model_Paym
entIntent_NextAction_VerifyW
0x000cb94c 0 sym.Lcom_stripe_model_SetupIntent_NextAction_VerifyWithMicrodeposits.ifield_arrivalDate:Ljava_lang_L
ong
0x000cb954 0 sym.Lcom_stripe_model_SetupIntent_NextAction_VerifyWithMicrodeposits.ifield_hostedVerificationUrl:Lj
ava_lang_String
0x000cb974 0 sym.Lcom_stripe_model_SetupIntent_NextAction.ifield_verifyWithMicrodeposits:Lcom_stripe_model_SetupI
ntent_NextAction_VerifyWithM
0x000dd058c 0 sym.Lj_b_e_b_t.sfield_VERIFY_FIELD_NUMBER:I
0x000dc27c 0 sym.imp.Landroid_view_ViewGroup.method.verifyDrawable_Landroid_graphics_drawable_Drawable_Z
```

Let's navigate (seek) to the method "0x0008269c" by using its flag and print the disassembly with the command pd:

```
[0x00000000]: + Lcom/stripe/IString_field;msg_verify_purchase@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_cancelling_invoice@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_cancelling_payment_request@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_create_charge@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_customer@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_discount@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_invoice@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_invoice_item@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_product_and_plan_for_invoice_item@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_creating_tax@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_email@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_phone@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_shipping_address@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_source@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_subscription@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_usage_plan@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_usage_plan_item@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_usage_plan_item_plan@I  
[0x00000000]: + Lcom/stripe/IString_field;msg_wait_change_usage_plan_item_product@I
```

Instead of just printing the disassembly to the console you may want to enter the so-called Visual Mode by typing V:

File	Actions	Edit	View	Help	Tab [1] [0x00175ac]
File	Settings	Edit	View	Tools	Emulate Debug Analyze Help
File	Classnames [129]				if [X] xit [Cache] Off
3e17e	- [0x00175a0]	Breakpoints			0 1 2 3 4 5 6 7 8
3e19c	- [0x00175b9]	Classes		
3e1bc	- [0x00175c4]	Comments			0c 01 11 01
3e1c4	- [0x00175d0]	Console			0
3e1c5	- [0x00175d1]	Debugger			0
3e1c6	- [0x0017738]	Decompiler			10
3e1c8	- [0x0017739]	Decompiler With Offsets			19
3e1c9	- [0x001746f]	Disassembly Summary			20
3e1ca	- [0x0017470]	Disassembly			21
3e1cb	- [0x0017523]	Entropy			22
3e1cd	- [0x0017529]	Entropy Fire			23
3e1cc	- [0x0017551]	File Hashes			24
3e1cf	- [0x0017751]	Function Calls			25
3e1d1	- [0x0017759]	Functions			26
3e1d2	- [0x0017760]	Groups			27
3e1d3	- [0x0017761]	Headers			28
3e1d4	- [0x001754b]	Hedump			29
3e1d6	- [0x0017788]	Imports			30
3e1d8	- [0x0017774]	Info			31
3e1d9	- [0x0017789]	Methods			32
3e1dc	- [0x0017400]	Relocs			33
3e1ec	- [0x001759a]	Sections			34
3e1e3	- [0x00174c3]	Segments			35
3e1e4	- [0x0017603]	Show All Decompiler Output	R		36
3e1e5	- [0x0017604]	Start Address			37
3e1e6	- [0x0017604]	Strings in data sections	ter		38
3e1e7	- [0x0017604]	Strings in the whole bin	ter		39
3e1dc	- [0x00176f5]	Summary			40
3e1fc	- [0x0017f62]	Symbols			41
3e1fd	- [0x0017f63]	Type Graph			42
3e1fe	- [0x0017f77]	Var READ Address			43
3e1ff	- [0x0018009]	Var WRITE address			44
3e200	- [0x0017001]	Xrefs Here			45
3e201	- [0x0017659]				46
3e202	- [0x001765a]				47
3e203	- [0x001765c]	- 0x0017d4aae [Lcom/google/android/mater			48
3e204	- [0x001765d]	- 0x0017d4bb0 [Lcom/google/android/mater			49
3e205	- [0x001765e]	- 0x0017d4c1 [Lcom/google/android/mater			50
3e206	- [0x001765f]	- 0x0017d4e1 [Lcom/google/android/mater			51
3e207	- [0x0017660]	- 0x0017f510 [Lcom/google/android/mater			52
3e208	- [0x0017661]	- 0x0017f512c [Lcom/google/android/mater			53
3e209	- [0x0017662]	- 0x0017f53dc [Lcom/google/android/mater			54
3e20a	- [0x0017663]	- 0x0017f560 [Lcom/google/android/mater			55
3e20b	- [0x0017664]	- 0x0017f560c [Lcom/google/android/mater			56
3e20c	- [0x0017665]	- 0x0017f560e [Lcom/google/android/mater			57
3e20d	- [0x0017666]	- 0x0017f560f [Lcom/google/android/mater			58
3e20e	- [0x0017667]	- 0x0017f560g [Lcom/google/android/mater			59
3e20f	- [0x0017668]	- 0x0017f560h [Lcom/google/android/mater			60
3e210	- [0x0017669]	- 0x0017f560i [Lcom/google/android/mater			61
3e211	- [0x001766a]	- 0x0017f560j [Lcom/google/android/mater			62
3e212	- [0x001766b]	- 0x0017f560k [Lcom/google/android/mater			63
3e213	- [0x001766c]	- 0x0017f560l [Lcom/google/android/mater			64
3e214	- [0x001766d]	- 0x0017f560m [Lcom/google/android/mater			65
3e215	- [0x001766e]	- 0x0017f560n [Lcom/google/android/mater			66
3e216	- [0x001766f]	- 0x0017f560o [Lcom/google/android/mater			67
3e217	- [0x0017670]	- 0x0017f560p [Lcom/google/android/mater			68
3e218	- [0x0017671]	- 0x0017f560q [Lcom/google/android/mater			69
3e219	- [0x0017672]	- 0x0017f560r [Lcom/google/android/mater			70
3e21a	- [0x0017673]	- 0x0017f560s [Lcom/google/android/mater			71
3e21b	- [0x0017674]	- 0x0017f560t [Lcom/google/android/mater			72
3e21c	- [0x0017675]	- 0x0017f560u [Lcom/google/android/mater			73
3e21d	- [0x0017676]	- 0x0017f560v [Lcom/google/android/mater			74
3e21e	- [0x0017677]	- 0x0017f560w [Lcom/google/android/mater			75
3e21f	- [0x0017678]	- 0x0017f560x [Lcom/google/android/mater			76
3e220	- [0x0017679]	- 0x0017f560y [Lcom/google/android/mater			77
3e221	- [0x001767a]	- 0x0017f560z [Lcom/google/android/mater			78
3e222	- [0x001767b]	- 0x0017f560{ [Lcom/google/android/mater			79
3e223	- [0x001767c]	- 0x0017f560 [Lcom/google/android/mater			80
3e224	- [0x001767d]	- 0x0017f560~ [Lcom/google/android/mater			81
3e225	- [0x001767e]	- 0x0017f560` [Lcom/google/android/mater			82
3e226	- [0x001767f]	- 0x0017f560^ [Lcom/google/android/mater			83
3e227	- [0x0017680]	- 0x0017f560< [Lcom/google/android/mater			84
3e228	- [0x0017681]	- 0x0017f560> [Lcom/google/android/mater			85
3e229	- [0x0017682]	- 0x0017f560# [Lcom/google/android/mater			86
3e22a	- [0x0017683]	- 0x0017f560@ [Lcom/google/android/mater			87
3e22b	- [0x0017684]	- 0x0017f560& [Lcom/google/android/mater			88
3e22c	- [0x0017685]	- 0x0017f560% [Lcom/google/android/mater			89
3e22d	- [0x0017686]	- 0x0017f560\$ [Lcom/google/android/mater			90
3e22e	- [0x0017687]	- 0x0017f560' [Lcom/google/android/mater			91
3e22f	- [0x0017688]	- 0x0017f560'` [Lcom/google/android/mater			92
3e22g	- [0x0017689]	- 0x0017f560'^ [Lcom/google/android/mater			93
3e22h	- [0x001768a]	- 0x0017f560'< [Lcom/google/android/mater			94
3e22i	- [0x001768b]	- 0x0017f560'> [Lcom/google/android/mater			95
3e22j	- [0x001768c]	- 0x0017f560'% [Lcom/google/android/mater			96
3e22k	- [0x001768d]	- 0x0017f560'\$ [Lcom/google/android/mater			97
3e22l	- [0x001768e]	- 0x0017f560'' [Lcom/google/android/mater			98
3e22m	- [0x001768f]	- 0x0017f560''` [Lcom/google/android/mater			99
3e22n	- [0x0017690]	- 0x0017f560''^ [Lcom/google/android/mater			100
3e22o	- [0x0017691]	- 0x0017f560''< [Lcom/google/android/mater			101
3e22p	- [0x0017692]	- 0x0017f560''> [Lcom/google/android/mater			102
3e22q	- [0x0017693]	- 0x0017f560''% [Lcom/google/android/mater			103
3e22r	- [0x0017694]	- 0x0017f560''\$ [Lcom/google/android/mater			104
3e22s	- [0x0017695]	- 0x0017f560''' [Lcom/google/android/mater			105
3e22t	- [0x0017696]	- 0x0017f560'''` [Lcom/google/android/mater			106
3e22u	- [0x0017697]	- 0x0017f560'''^ [Lcom/google/android/mater			107
3e22v	- [0x0017698]	- 0x0017f560'''< [Lcom/google/android/mater			108
3e22w	- [0x0017699]	- 0x0017f560'''> [Lcom/google/android/mater			109
3e22x	- [0x001769a]	- 0x0017f560'''% [Lcom/google/android/mater			110
3e22y	- [0x001769b]	- 0x0017f560'''\$ [Lcom/google/android/mater			111
3e22z	- [0x001769c]	- 0x0017f560'''' [Lcom/google/android/mater			112
3e22{	- [0x001769d]	- 0x0017f560''''` [Lcom/google/android/mater			113
3e22^	- [0x001769e]	- 0x0017f560''''' [Lcom/google/android/mater			114
3e22~	- [0x001769f]	- 0x0017f560'''''` [Lcom/google/android/mater			115
3e22`	- [0x00176a0]	- 0x0017f560'''''' [Lcom/google/android/mater			116
3e22^	- [0x00176a1]	- 0x0017f560''''''` [Lcom/google/android/mater			117
3e22~	- [0x00176a2]	- 0x0017f560''''''' [Lcom/google/android/mater			118
3e22`	- [0x00176a3]	- 0x0017f560'''''''` [Lcom/google/android/mater			119
3e22^	- [0x00176a4]	- 0x0017f560'''''''' [Lcom/google/android/mater			120
3e22~	- [0x00176a5]	- 0x0017f560''''''''` [Lcom/google/android/mater			121
3e22`	- [0x00176a6]	- 0x0017f560''''''''' [Lcom/google/android/mater			122
3e22^	- [0x00176a7]	- 0x0017f560'''''''''` [Lcom/google/android/mater			123
3e22~	- [0x00176a8]	- 0x0017f560'''''''''` [Lcom/google/android/mater			124
3e22`	- [0x00176a9]	- 0x0017f560'''''''''` [Lcom/google/android/mater			125
3e22^	- [0x00176a0]	- 0x0017f560'''''''''` [Lcom/google/android/mater			126
3e22~	- [0x00176a1]	- 0x0017f560'''''''''` [Lcom/google/android/mater			127
3e22`	- [0x00176a2]	- 0x0017f560'''''''''` [Lcom/google/android/mater			128
3e22^	- [0x00176a3]	- 0x0017f560'''''''''` [Lcom/google/android/mater			129
3e22~	- [0x00176a4]	- 0x0017f560'''''''''` [Lcom/google/android/mater			130
3e22`	- [0x00176a5]	- 0x0017f560'''''''''` [Lcom/google/android/mater			131
3e22^	- [0x00176a6]	- 0x0017f560'''''''''` [Lcom/google/android/mater			132
3e22~	- [0x00176a7]	- 0x0017f560'''''''''` [Lcom/google/android/mater			133
3e22`	- [0x00176a8]	- 0x0017f560'''''''''` [Lcom/google/android/mater			134
3e22^	- [0x00176a9]	- 0x0017f560'''''''''` [Lcom/google/android/mater			135
3e22~	- [0x00176a0]	- 0x0017f560'''''''''` [Lcom/google/android/mater			136
3e22`	- [0x00176a1]	- 0x0017f560'''''''''` [Lcom/google/android/mater			137
3e22^	- [0x00176a2]	- 0x0017f560'''''''''` [Lcom/google/android/mater			138
3e22~	- [0x00176a3]	- 0x0017f560'''''''''` [Lcom/google/android/mater			139
3e22`	- [0x00176a4]	- 0x0017f560'''''''''` [Lcom/google/android/mater			140
3e22^	- [0x00176a5]	- 0x0017f560'''''''''` [Lcom/google/android/mater			141
3e22~	- [0x00176a6]	- 0x0017f560'''''''''` [Lcom/google/android/mater			142
3e22`	- [0x00176a7]	- 0x0017f560'''''''''` [Lcom/google/android/mater			143
3e22^	- [0x00176a8]	- 0x0017f560'''''''''` [Lcom/google/android/mater			144
3e22~	- [0x00176a9]	- 0x0017f560'''''''''` [Lcom/google/android/mater			145
3e22`	- [0x00176a0]	- 0x0017f560'''''''''` [Lcom/google/android/mater			146
3e22^	- [0x00176a1]	- 0x0017f560'''''''''` [Lcom/google/android/mater			147
3e22~	- [0x00176a2]	- 0x0017f560'''''''''` [Lcom/google/android/mater			148
3e22`	- [0x00176a3]	- 0x0017f560'''''''''` [Lcom/google/android/mater			149 . . .

print 32 byte hex dump current block and filter flags with ~grep (same as |grep):

```
[0x000826a0]> px 32
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF Raw Bytes
0x000826a0 6698 0000 5e0d 0400 6798 0000 5e0d 0400 f...^...g...^...
0x000826b0 6898 0000 5e0d 0400 6998 0000 5e0d 0400 h...^...i...^...
[0x000826a0]> f~foo
0x0007dc1c 0 sym.Lcom_stripe_R_id.sfield_footer_container:I
0x0007f04c 0 sym.Lcom_stripe_R_id.sfield_stripe_add_payment_method_footer:I
0x0007f07c 0 sym.Lcom_stripe_R_id.sfield_stripe_payment_methods_footer:I
0x0008df24 0 sym.Lcom_stripe_android_R_id.sfield_footer_container:I
0x0008e5ec 0 sym.Lcom_stripe_android_R_id.sfield_stripe_add_payment_method_footer:I
0x0008e61c 0 sym.Lcom_stripe_android_R_id.sfield_stripe_payment_methods_footer:I
0x0008e574c 0 sym.Lcom_stripe_android_databinding_PaymentMethodsActivityBinding.isfiled_footerContainer:Landroid_widget_FrameLayout
0x000a006c 0 sym.Lcom_stripe_android_payments_R_id.sfield_footer_container:I
0x000a07bc 0 sym.Lcom_stripe_android_payments_R_id.sfield_stripe_add_payment_method_footer:I
0x000a07ec 0 sym.Lcom_stripe_android_payments_R_id.sfield_stripe_payment_methods_footer:I
0x000ad54c 0 sym.Lcom_stripe_android_paymentsheet_R_id.sfield_footer_container:I
0x000adc9c 0 sym.Lcom_stripe_android_paymentsheet_R_id.sfield_stripe_add_payment_method_footer:I
0x000adccc 0 sym.Lcom_stripe_android_paymentsheet_R_id.sfield_stripe_payment_methods_footer:I
0x000c9894 0 sym.Lcom_stripe_model_Customer_InvoiceSettings.isfiled_footer:Ljava_lang_String
0x000ca01c 0 sym.Lcom_stripe_model_Invoice.isfiled_footer:Ljava_lang_String
0x000cb524 0 sym.Lcom_stripe_model_Quote.isfiled_footer:Ljava_lang_String
0x005bc91a 6 str.footer
0x005bc922 15 str.footerContainer
0x005bc933 16 str.footer_container
0x005ff6fc 32 str.stripe_add_payment_method_footer
0x00600bc7 29 str.stripe_payment_methods_footer
0x006008a81 27 str.viewBinding.footerContainer
```

V. Android Code Quality and Build Settings:

Going to test some of the android code quality methods.

1. Making Sure That the App is Properly Signed (MSTG-CODE-1):

i. Static Analysis:

Make sure that the release build has been signed via both the v1 and v2 schemes for Android 7.0 (API level 24) and above and via all the three schemes for Android 9 (API level 28) and above, and that the code-signing certificate in the APK belongs to the developer.

We will verify this by using the following tool:

Downloading it:

```
(joxavy㉿kali)-[~/Downloads]
$ sudo apt-get -y install apksigner

[sudo] password for joxavy:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  faraday-client libarmadillo10 libcharls2 libgdal29 libgeos3.10.1 libicu71:i386 libigl
    python3-deprecation python3-llvmlite python3-pypyproj python3-pyshp python3.9 python3.
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libhansksig-java
```

Using it to check for signature:

```
(joxavy㉿kali)-[~/Downloads]
$ apksigner verify --verbose com.stripe.credit.card.payment_2022-10-07.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Verifies
Verified using v1 scheme (JAR signing): true
Verified using v2 scheme (APK Signature Scheme v2): true
Verified using v3 scheme (APK Signature Scheme v3): true
Verified using v4 scheme (APK Signature Scheme v4): false
Verified for SourceStamp: false
Number of signers: 1
WARNING: META-INF/3ds2sdk_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/SOSegmentSDK_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
```

As seen in the result, the apk has been signed with v1, v2 and v3. But not v4.

It also displays the files which are not protected by signature:

```
WARNING: META-INF/3ds2sdk_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/SOSegmentSDK_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/activity-compose_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/activity-ktx_release.kotlin_module not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/androidx.activity_activity-compose_version not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
WARNING: META-INF/androidx.activity_ktx_version not protected by signature. Unauthorized modifications to this JAR entry will not be detected. Delete or move the entry outside of META-INF/.
```

The contents of the signing certificate can be examined with jarsigner. Note that the Common Name (CN) attribute is set to "Android Debug" in the debug certificate.

The output for an APK signed with a debug certificate is shown below:

```
joxavy@kali:[~/Downloads]
$ jarsigner -verify -verbose com.stripe.credit.card.payment_2022-10-07.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
SM 44526 Thu Jan 01 01:01:02 EST 1981 AndroidManifest.xml
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 2714 Thu Jan 01 01:01:02 EST 1981 DebugProsekt.ktbin
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 26 Thu Jan 01 01:01:02 EST 1981 META-INF/3ds2sdk_release.kotlin_module
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 107 Thu Jan 01 01:01:02 EST 1981 META-INF/SOSegmentSDK_release.kotlin_module
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 139 Thu Jan 01 01:01:02 EST 1981 META-INF/activity-compose_release.kotlin_module
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 201 Thu Jan 01 01:01:02 EST 1981 META-INF/activity-ktx_release.kotlin_module
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
SM 6 Thu Jan 01 01:01:02 EST 1981 META-INF/androidx.activity-compose_version
<>>> Signer
X.509, CN=Android, O=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Signature algorithm: SHA256withRSA, 4096-bit key
[certificate is valid from 2/29/10, 4:41 AM to 2/16/40, 4:41 AM]
[invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
```

ii. Dynamic Analysis:

Static analysis should be used to verify the APK signature.

2. Testing Whether the App is Debuggable (MSTG-CODE-2):

The android: debuggable attribute in the Application element that is defined in the Android manifest determines whether the app can be debugged or not.

i. Static Analysis:

Check AndroidManifest.xml to determine whether the android: debuggable attribute has been set and to find the attribute's value.

We open the AndroidManifest.xml file and check the <application> part:

```
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-feature android:glEsVersion="0x00020000" android:required="true"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:configChanges="layoutDirection|locale" android:hardwareAccelerated="true" android:icon="@mipmap/launcher" android:label="@string/app_name" android:largeHeap="true" android:name="com.easypay.stripe.MyApplication" android:requestLegacyExternalStorage="true" android:roundIcon="@mipmap/launcher_round" android:supportsRtl="true" android:theme="@style/AppTheme.NoActionBar" android:usesCleartextTraffic="true">
    <activity android:exported="true" android:name="com.easypay.stripe.activity.ChangePassword" android:screenOrientation="portrait"/>
    <activity android:exported="true" android:name="com.easypay.stripe.activity.ForgotPassword" android:screenOrientation="portrait"/>
    <activity android:exported="true" android:name="com.easypay.stripe.activity.LoginActivity" android:screenOrientation="portrait"/>
```

We can see that it doesn't contain the android debuggable attribute.

You can use aapt tool from the Android SDK with the following command line to quickly check if the android: debuggable="true" directive is present, by using the following command:

```
(joxavy㉿kali)-[~/Downloads]
$ aapt d xmltree com.stripe.credit.card.payment_2022-10-07.apk AndroidManifest.xml | grep -Ec "android:debuggable\(\0x[0-9a-f]+\)\=\\"type\s0x[0-9a-f]+\0*fffffff"
```

If the command prints 1 then the directive is present, but in our case, it returns 0, as you can see 1 is not found. Therefore, the android debuggable directive is not present.

ii. Dynamic Analysis:

First, we need to determine the package name of the apk, by using this command:

```
(joxavy㉿kali)-[~/Downloads]
$ aapt dump badging com.stripe.credit.card.payment_2022-10-07.apk | grep package | awk '{print $2}' | sed s/name=//g | sed s/>\//g
com.stripe.credit.card.payment
(joxavy㉿kali)-[~/Downloads]
$ ./apkinfo.sh [path-to-apk-file.apk]
```

The output gives that: com.stripe.credit.card.payment

Is the name of package, so we will use it.

adb can be used to determine whether an application is debuggable.

Use the following command:

```
(joxavy㉿kali)-[~/Downloads]
$ adb shell dumpsys package com.stripe.credit.card.payment | grep -c "DEBUGGABLE"
0

(joxavy㉿kali)-[~/Downloads]
$
```

If the command prints a number superior to zero then the application have the debug flag, if not, it doesn't. in our case, the command prints 0, which means it indeed doesn't contain the debug flag.

3. Testing for Debugging Code and Verbose Error Logging (MSTG-CODE-4): Overview:

StrictMode is a developer tool for detecting violations, e.g., accidental disk or network access on the application's main thread. It can also be used to check for good coding practices, such as implementing performant code.

i. Static Analysis:

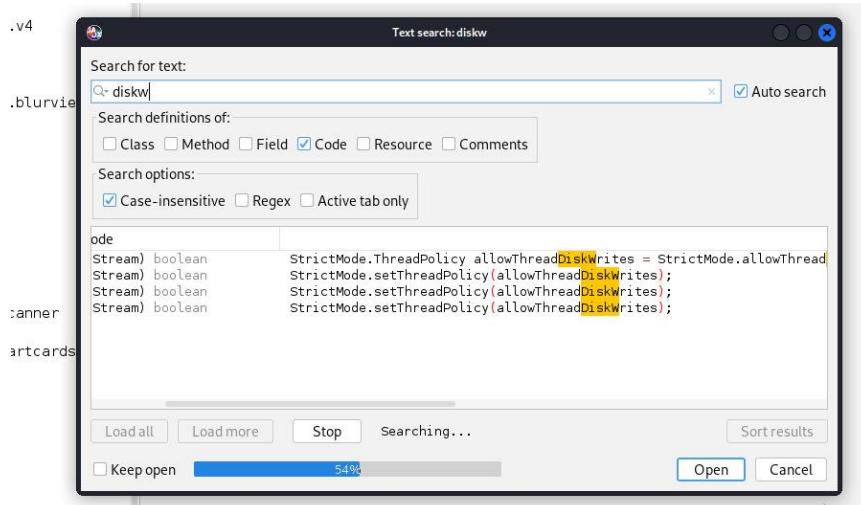
To determine whether StrictMode is enabled, you can look for the StrictMode.setThreadPolicy or StrictMode.setVmPolicy methods. Most likely, they will be in the onCreate method.

the detection methods for the thread policy are

- ◆ detectDiskWrites()
- ◆ detectDiskReads()
- ◆ detectNetwork()

we will check if these methods exist by opening the apk inside the Jadx tool and searching for them one by one:

search for detectDiskWrites:



As we can see, it exists in our apk.

Search for detectDiskReads:

Search for text: Auto search

Search definitions of: Class Method Field Code Resource Comments

Search options: Case-insensitive Regex Active tab only

```

ode
ad.engine.b0.b() int StrictMode.ThreadPolicy allowThreadDiskReads = StrictMode.allowThread
ad.engine.b0.b() int StrictMode.setThreadPolicy(allowThreadDiskReads);
ad.engine.b0.b() int StrictMode.setThreadPolicy(allowThreadDiskReads);
is.common.util.o.a() String StrictMode.ThreadPolicy allowThreadDiskReads;
is.common.util.o.a() String StrictMode.setThreadPolicy(allowThreadDiskReads);
is.common.x.a(String, t, k StrictMode.ThreadPolicy allowThreadDiskReads = StrictMode.allowThread
is.common.x.a(String, t, k StrictMode.setThreadPolicy(allowThreadDiskReads);
is.common.x.b(String, bool StrictMode.ThreadPolicy allowThreadDiskReads = StrictMode.allowThread

```

Load all Stop Searching... Sort results

Keep open

Search for DetectNetwrok:

Text search: detectnetwork

Search for text: Auto search

Search definitions of: Class Method Field Code Resource Comments

Search options: Case-insensitive Regex Active tab only

```

Code
setThreadPolicy(new StrictMode.ThreadPolicy.Builder().detectNetwork().penaltyDeath().build());

```

We conclude that the detection methods for the thread policies are indeed included in the app.

The penalties for thread policy violation are:

- ◆ `penaltyLog()` // Logs a message to LogCat
- ◆ `penaltyDeath()` // Crashes application, runs at the end of all enabled penalties
- ◆ `penaltyDialog()` // Shows a dialog

we will also check for the existence of the penalties with Jadx.

The screenshot shows a search interface with the following details:

- Text search:** penalty
- Search for text:** penalty
- Search definitions of:** Code (selected), Class, Method, Field, Resource, Comments
- Search options:** Case-insensitive (selected), Regex, Active tab only
- Results:** Code


```
icy(new StrictMode.ThreadPolicy.Builder().detectNetwork().penaltyDeath().build());
```
- Buttons at the bottom:** Load all, Load more, Stop, Found 1 (complete), Sort results

Only the penalty of death exists.

ii. Dynamic Analysis

There are several ways of detecting StrictMode; the best choice depends on how the policies' roles are implemented. They include

Logcat,

a warning dialog,

application crash.

VI. Graphing the apk dependencies:

Class dependency visualizer. Only apk file is needed.

Class coupling is one of the significant code metrics that shows how easy is to change, maintain and test the code. This tool helps to view whole picture of the project

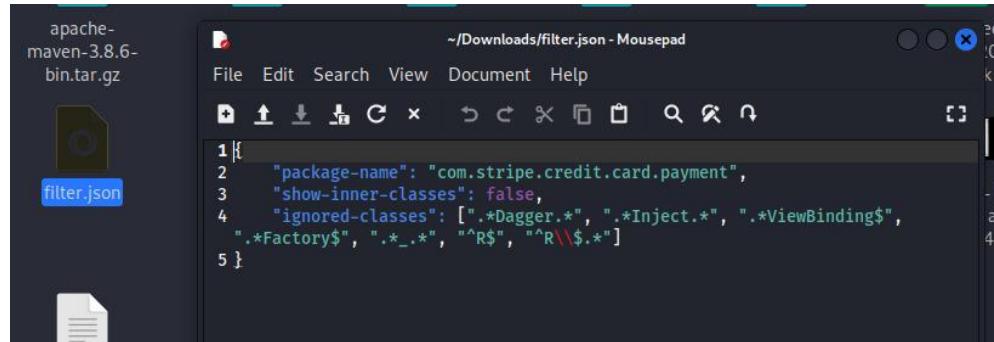
Unzipping the file:

```
(joxavy㉿kali)-[~/Downloads]
└─$ unzip apk-dependency-graph-scripts-0.3.1.zip
Archive: apk-dependency-graph-scripts-0.3.1.zip
  inflating: build/libs/apk-dependency-graph.jar
  inflating: gui/analyzed.js
  inflating: gui/index.html
  creating: gui/scripts/
  inflating: gui/scripts/graph.js
  inflating: gui/scripts/d3-setup-custom.js
  inflating: gui/scripts/graph-actions-select.js
  inflating: gui/scripts/parse.js
  inflating: gui/scripts/graph-actions-select-compiled.js
  inflating: gui/scripts/parse-compiled.js
  inflating: filters/default.json
  inflating: filters/instructions.txt
  inflating: run.bat
  inflating: run.sh
```

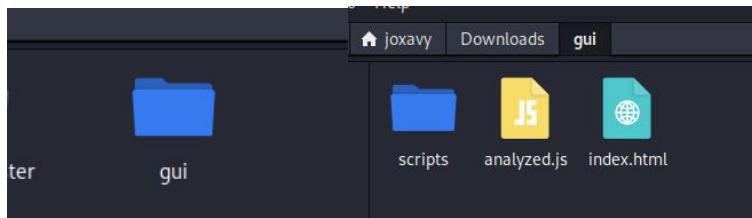
Running the tool:

```
(root㉿kali)-[~/home/joxavy/Downloads]
# ./run.sh com.stripe.credit.card.payment 2022-10-07.apk filter.json
Baksmaling classes.dex ...
Baksmaling classes2.dex ...
Baksmaling classes3.dex ...
Analyzing dependencies ...
Success! Now open gui/index.html in your browser.
```

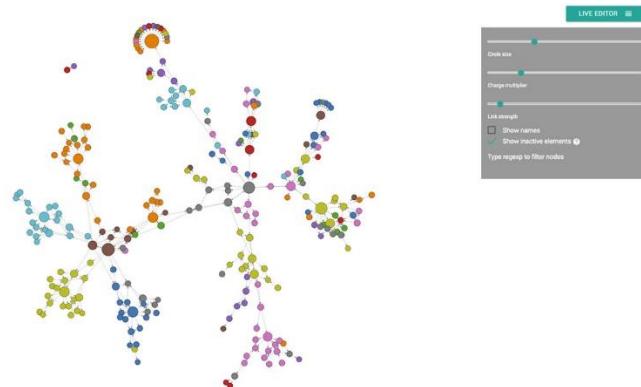
The filter.json file:



A file called index.html is created that contains the graph.



The output:



It's a good architecture with low class coupling.