Sri Lanka Institute of Information Technology

Static & Dynamic Analysis of an Android Application

and

Decentralized Application Concept

IE3112 - Mobile Security

Individual Assignment

Submitted by:

Student Registration Number	Student Name
IT19029146	Eranda H.P.D

26/06/2021
Date of submission

Table of Contents

Part A.....3

1. Introduction 3

2. Static Analysis.....4

3. Dynamic Analysis.....18

Part B.....40

1. Definition & Explanation of Decentralized Apps 40

2. Building a suitable Case Scenario.....42

Part A Introduction

In the modern world, there are millions of mobile applications which are available for both Android and iOS. When the Android Operating System is considered, most of the android applications have already become the victims of various cyber criminals. The main reason for this issue is that most of the android applications are not developed up to the necessary security standards. So in order to find security flaws and backdoors in mobile applications the tool called **"MobSF"** can be used.

What is MobSF ?

MobSF, also known as **"Mobile Security Framework",** is an open-source, all-in-one mobile application pen-testing tool developed by *Ajin Abraham.* This particular tool can be used for mobile penetration testing, malware analysis and security assessment of Android applications. MobSF is a framework written mostly in Python, with support for JavaScript, Smali code, and shell commands.

MobSF tool is capable of performing both static & dynamic penetration testing. This application supports both binaries ***(APK, IPA & APPX)*** and zipped source code. Performing those static & dynamic analysis tests, allows us to identify possible threats and vulnerabilities which may allow external party to access private information stored on the mobile device.

MobSF tool has a graphical user interface in the form of a web service. This particular web service consists with a dashboard which clearly represents the final results of the analysis. Additionally, mobSF tool has an integrated simulator and an API which allows users to trigger the analysis automatically. Usually, the analysis process automatically begins when an user sends the application towards this mobSF tool. Then this tool decompiles the application and analyze its decompiled source code. So, as the final outcome, this tool lists down all the types of analysis and vulnerabilities that it can find.

Static Analysis

In order to perform the static analysis, I'm going to use my Kali-Linux virtual machine as the host machine. However, there are several prerequisites that need to be fulfilled before beginning the static analysis procedure.

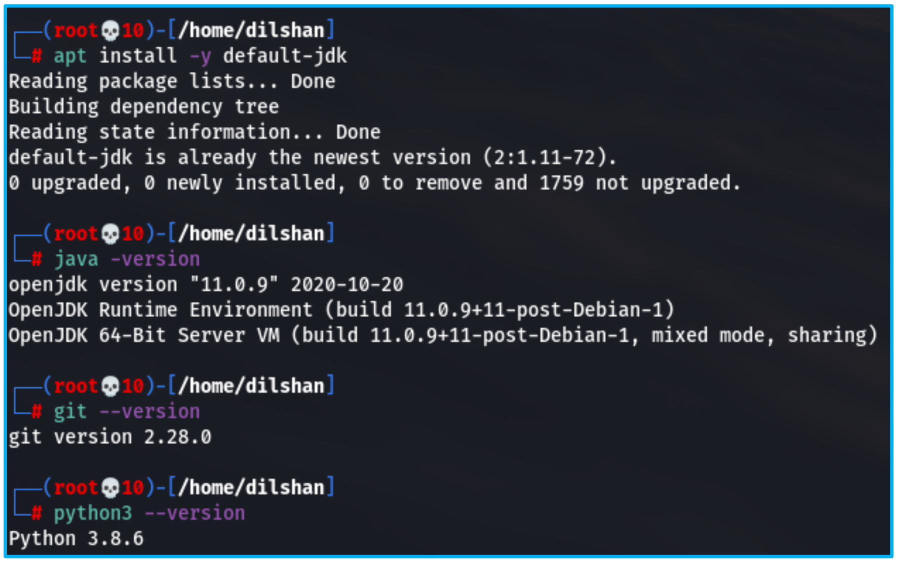
How to install mobSF ?

In order to install mobSF in our Kali-Linux virtual machine, the following requirements need to be fulfilled first.

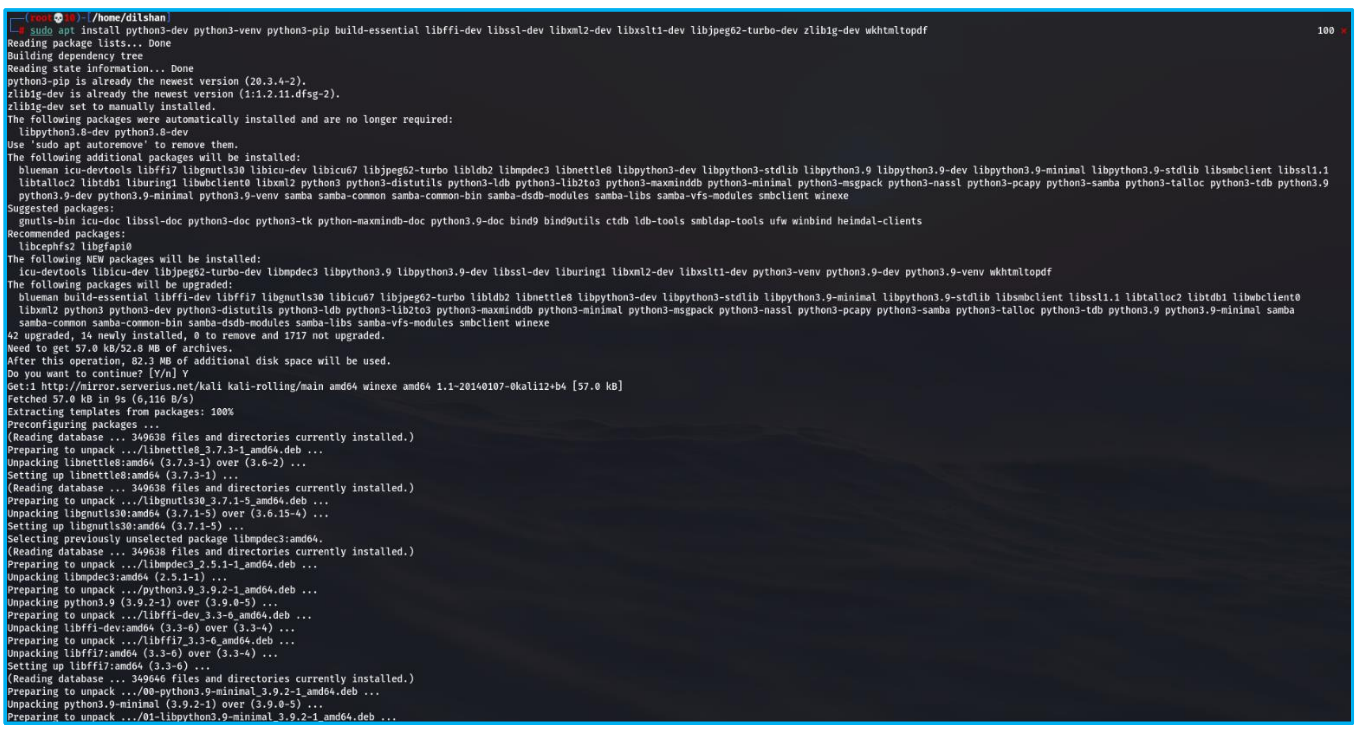
As the very first step you need to update all the packages & then install **"python3".**

```
┌──(root💀10)-[/home/dilshan]
└─# apt update
Get:1 http://mirror.serverius.net/kali kali-rolling InRelease [30.5 kB]
Get:2 http://mirror.serverius.net/kali kali-rolling/main amd64 Packages [17.7 MB]
Get:3 http://mirror.serverius.net/kali kali-rolling/contrib amd64 Packages [108 kB]
Get:4 http://mirror.serverius.net/kali kali-rolling/non-free amd64 Packages [199 kB]
Fetched 18.0 MB in 44s (414 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1759 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(root💀10)-[/home/dilshan]
└─# apt -y install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python-pip-whl python3-wheel
The following NEW packages will be installed:
  python-pip-whl python3-pip python3-wheel
0 upgraded, 3 newly installed, 0 to remove and 1759 not upgraded.
Need to get 2,284 kB/2,308 kB of archives.
After this operation, 3,669 kB of additional disk space will be used.
Get:1 http://mirror.serverius.net/kali kali-rolling/main amd64 python-pip-whl all 20.3.4-2 [1,947 kB]
Get:2 http://mirror.neostrada.nl/kali kali-rolling/main amd64 python3-pip all 20.3.4-2 [337 kB]
Fetched 2,284 kB in 21s (107 kB/s)
Selecting previously unselected package python-pip-whl.
(Reading database ... 349344 files and directories currently installed.)
Preparing to unpack .../python-pip-whl_20.3.4-2_all.deb ...
Unpacking python-pip-whl (20.3.4-2) ...
Selecting previously unselected package python3-wheel.
Preparing to unpack .../python3-wheel_0.34.2-1_all.deb ...
Unpacking python3-wheel (0.34.2-1) ...
Selecting previously unselected package python3-pip.
Preparing to unpack .../python3-pip_20.3.4-2_all.deb ...
Unpacking python3-pip (20.3.4-2) ...
Setting up python3-wheel (0.34.2-1) ...
Setting up python-pip-whl (20.3.4-2) ...
Setting up python3-pip (20.3.4-2) ...
Processing triggers for man-db (2.9.3-2) ...
Processing triggers for kali-menu (2020.4.0) ...
```

Then you are required to installed Java Development Kit (JDK8+). After successfully installing those packages, you can view the version numbers of them as following.



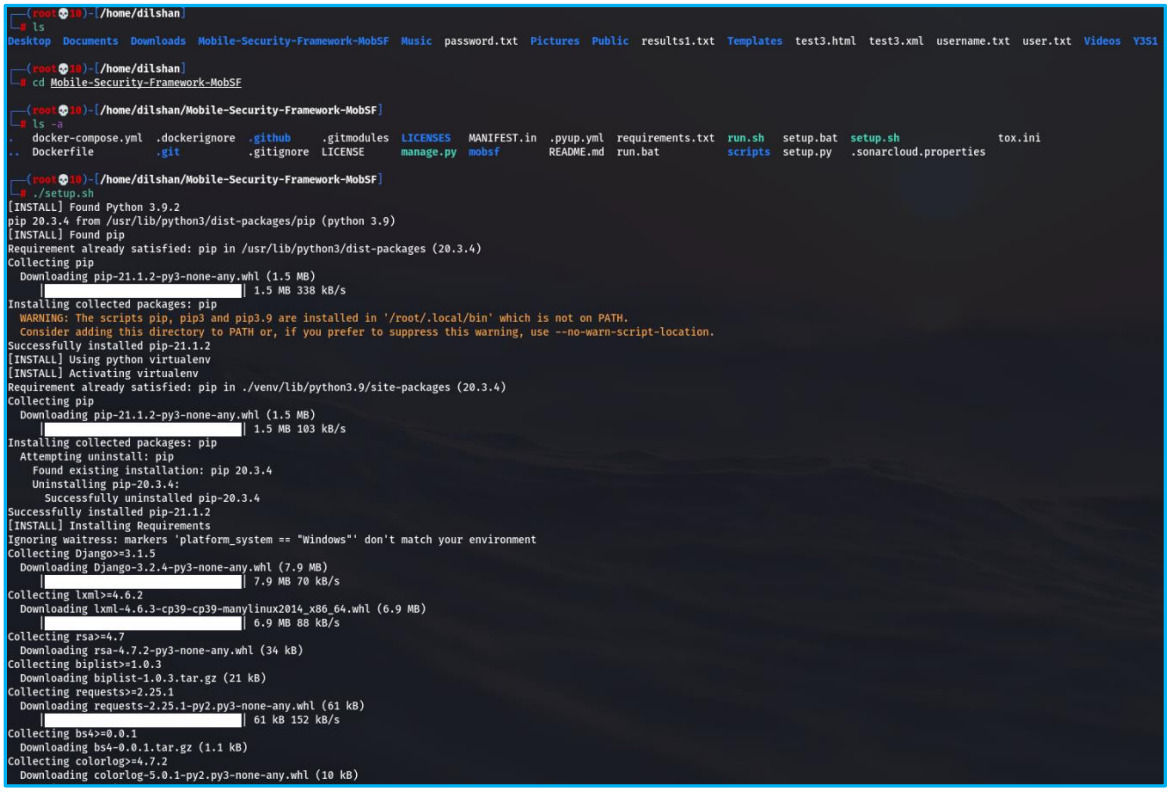
After that you need to install the following dependencies before running the MobSF setup files. The MobSF documentation has clearly mentioned the **libjpeg8-dev library** as a prerequisite.

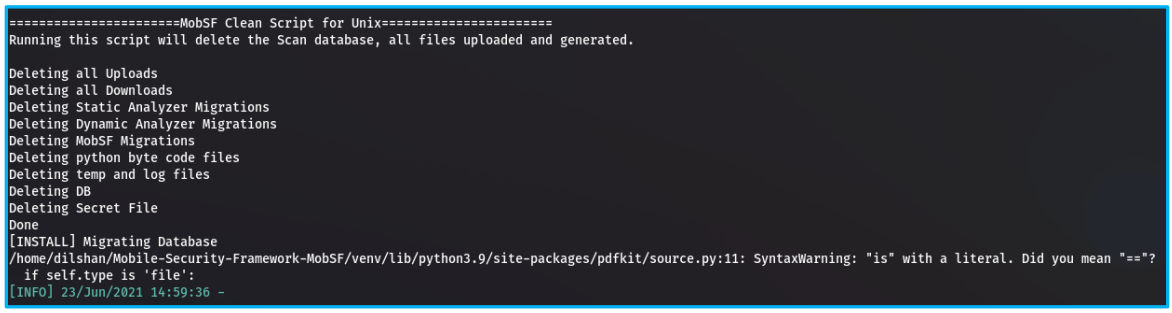


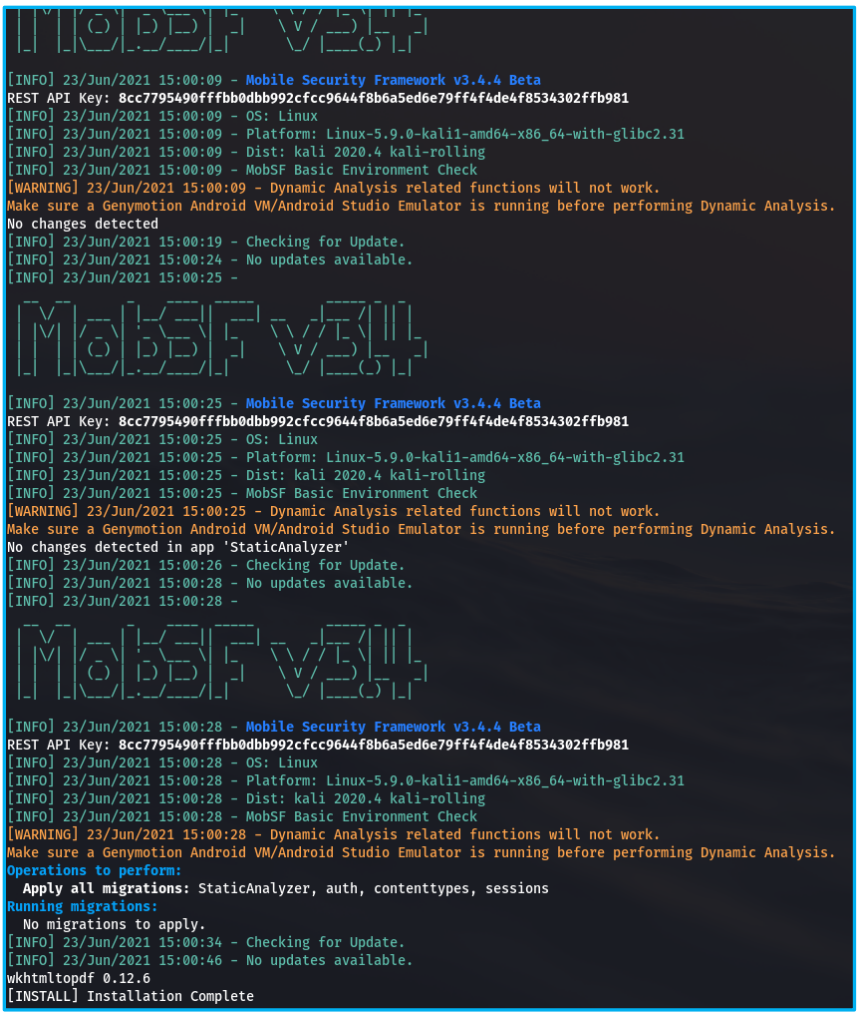
After the successful completion of all the above dependencies, you need to clone the MobSF source code.

```
┌──(root💀10)-[/home/dilshan]
└─# git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
Cloning into 'Mobile-Security-Framework-MobSF'...
remote: Enumerating objects: 17597, done.
remote: Counting objects: 100% (500/500), done.
remote: Compressing objects: 100% (288/288), done.
remote: Total 17597 (delta 281), reused 370 (delta 208), pack-reused 17097
Receiving objects: 100% (17597/17597), 1.11 GiB | 413.00 KiB/s, done.
Resolving deltas: 100% (8488/8488), done.
```

Then you can navigate to the tool's directory and then run the setup file in order to install the framework.





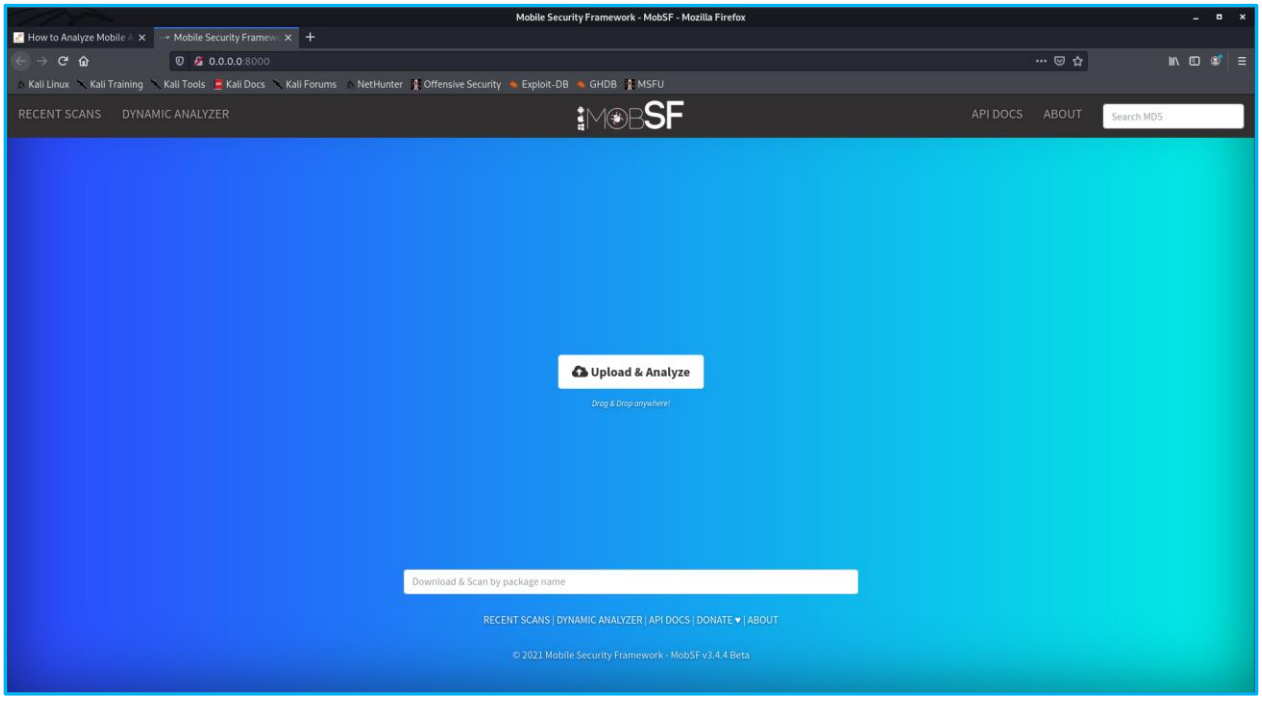

[INFO] 23/Jun/2021 15:00:09 - Mobile Security Framework v3.4.4 Beta
REST API Key: 8cc7795490fffbb0dbb992cfcc9644f8b6a5ed6e79ff4f4de4f8534302ffb981
[INFO] 23/Jun/2021 15:00:09 - OS: Linux
[INFO] 23/Jun/2021 15:00:09 - Platform: Linux-5.9.0-kali1-amd64-x86_64-with-glibc2.31
[INFO] 23/Jun/2021 15:00:09 - Dist: kali 2020.4 kali-rolling
[INFO] 23/Jun/2021 15:00:09 - MobSF Basic Environment Check
[WARNING] 23/Jun/2021 15:00:09 - 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
[INFO] 23/Jun/2021 15:00:19 - Checking for Update.
[INFO] 23/Jun/2021 15:00:24 - No updates available.
[INFO] 23/Jun/2021 15:00:25 -
[INFO] 23/Jun/2021 15:00:25 - Mobile Security Framework v3.4.4 Beta
REST API Key: 8cc7795490fffbb0dbb992cfcc9644f8b6a5ed6e79ff4f4de4f8534302ffb981
[INFO] 23/Jun/2021 15:00:25 - OS: Linux
[INFO] 23/Jun/2021 15:00:25 - Platform: Linux-5.9.0-kali1-amd64-x86_64-with-glibc2.31
[INFO] 23/Jun/2021 15:00:25 - Dist: kali 2020.4 kali-rolling
[INFO] 23/Jun/2021 15:00:25 - MobSF Basic Environment Check
[WARNING] 23/Jun/2021 15:00:25 - 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] 23/Jun/2021 15:00:26 - Checking for Update.
[INFO] 23/Jun/2021 15:00:28 - No updates available.
[INFO] 23/Jun/2021 15:00:28 -
[INFO] 23/Jun/2021 15:00:28 - Mobile Security Framework v3.4.4 Beta
REST API Key: 8cc7795490fffbb0dbb992cfcc9644f8b6a5ed6e79ff4f4de4f8534302ffb981
[INFO] 23/Jun/2021 15:00:28 - OS: Linux
[INFO] 23/Jun/2021 15:00:28 - Platform: Linux-5.9.0-kali1-amd64-x86_64-with-glibc2.31
[INFO] 23/Jun/2021 15:00:28 - Dist: kali 2020.4 kali-rolling
[INFO] 23/Jun/2021 15:00:28 - MobSF Basic Environment Check
[WARNING] 23/Jun/2021 15:00:28 - 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] 23/Jun/2021 15:00:34 - Checking for Update.
[INFO] 23/Jun/2021 15:00:46 - No updates available.
wkhtmltopdf 0.12.6
[INSTALL] Installation Complete

Performing Static Analysis

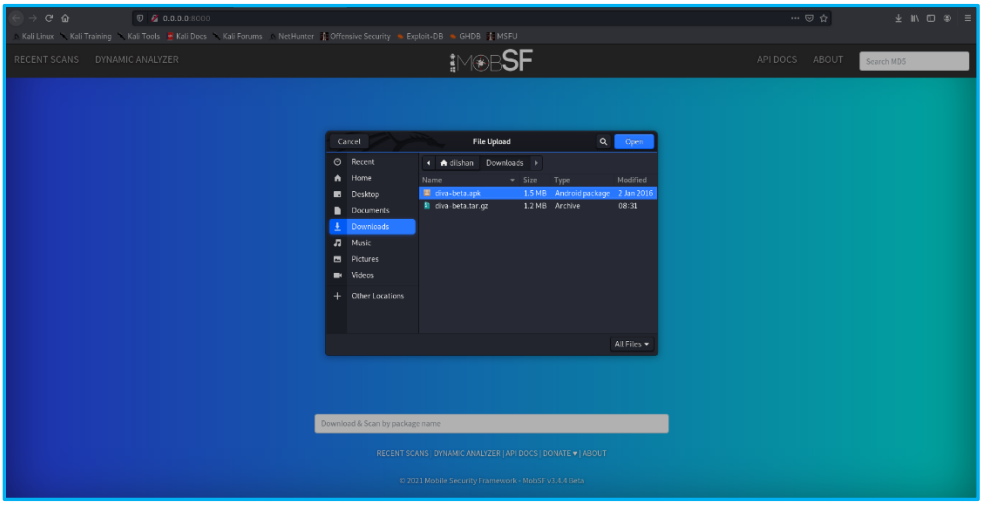
In order to run the mobSF tool, you need to give **"./run.sh"** command.

```
┌──(root💀10)-[/home/dilshan/Mobile-Security-Framework-MobSF]
└─# ./run.sh
[2021-06-23 20:40:20 +0530] [14071] [INFO] Starting gunicorn 20.1.0
[2021-06-23 20:40:20 +0530] [14071] [INFO] Listening at: http://0.0.0.0:8000 (14071)
[2021-06-23 20:40:20 +0530] [14071] [INFO] Using worker: gthread
[2021-06-23 20:40:20 +0530] [14072] [INFO] Booting worker with pid: 14072
[INFO] 23/Jun/2021 15:10:57 -

[INFO] 23/Jun/2021 15:10:57 - Mobile Security Framework v3.4.4 Beta
REST API Key: 8cc7795490fffbb0dbb992cfcc9644f8b6a5ed6e79ff4f4de4f8534302ffb981
[INFO] 23/Jun/2021 15:10:57 - OS: Linux
[INFO] 23/Jun/2021 15:10:57 - Platform: Linux-5.9.0-kali1-amd64-x86_64-with-glibc2.31
[INFO] 23/Jun/2021 15:10:57 - Dist: kali 2020.4 kali-rolling
[INFO] 23/Jun/2021 15:10:57 - MobSF Basic Environment Check
[WARNING] 23/Jun/2021 15:10:57 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
[INFO] 23/Jun/2021 15:11:02 - Checking for Update.
[INFO] 23/Jun/2021 15:11:13 - No updates available.
```

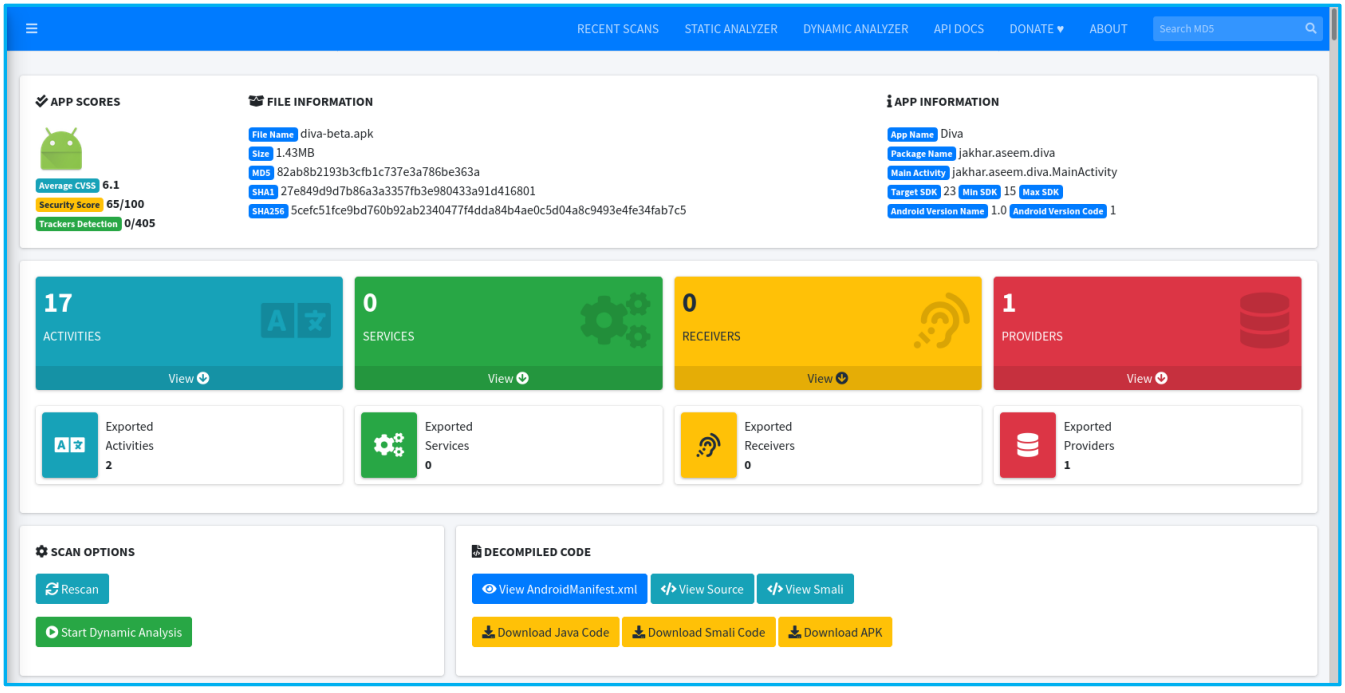
After executing the above command, it opens the MobSF dashboard on the localhost server running at the default port of **8000**.



Then you need to upload the **previously downloaded APK file** to the MobSF dashboard. In my case, I have downloaded the **“DIVA-beta”** android application in order to perform static analysis on it.



DIVA ***(Damn insecure and vulnerable App)*** is an app that is intentionally designed to be insecure. This particular app covers common vulnerabilities in Android apps ranging from insecure storage, input validation to access control issues. The following image provides a top-level overview of the target APK file showing the risk score and development information related with it. The higher this score the more secure app is.

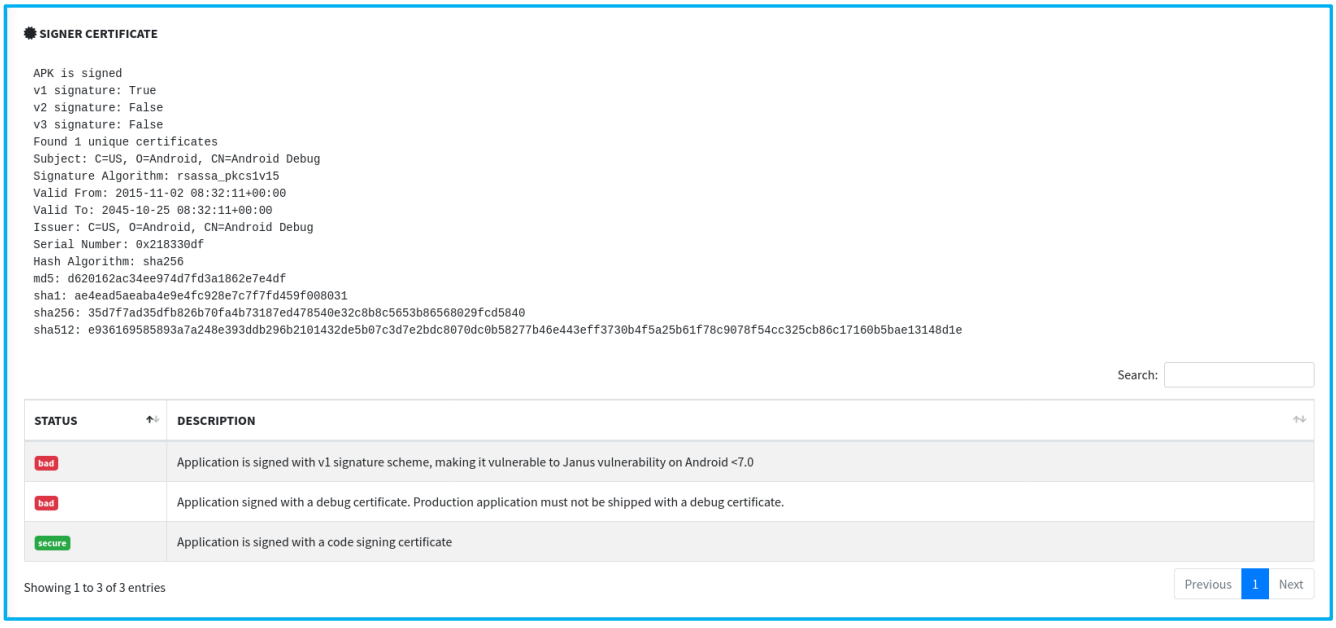


When we scroll down the results, different types of analytical information can be found. Some of them are as following.

1) Signer Certificate
2) Application Permissions
3) Android API
4) Browsable Activities
5) Network Security
6) Manifest Analysis
7) Code Analysis
8) NIAP Analysis
9) APK ID Analysis
10) Server Location
11) Domain Malware Check
12) Hardcoded Secrets

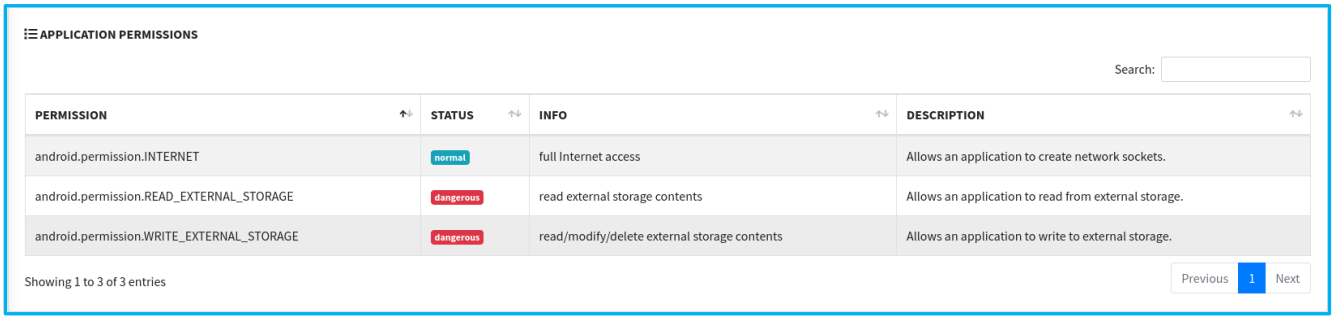
Signer Certificate

The application singing certificate gives detailed information about the supported hashes and encryption followed by the application.



Application Permissions

This shows the dangerous read/write permissions which are granted to the application. So, by looking at those permissions we can get a clear understanding on which permissions can lead to further damage.



APPLICATION PERMISSIONS

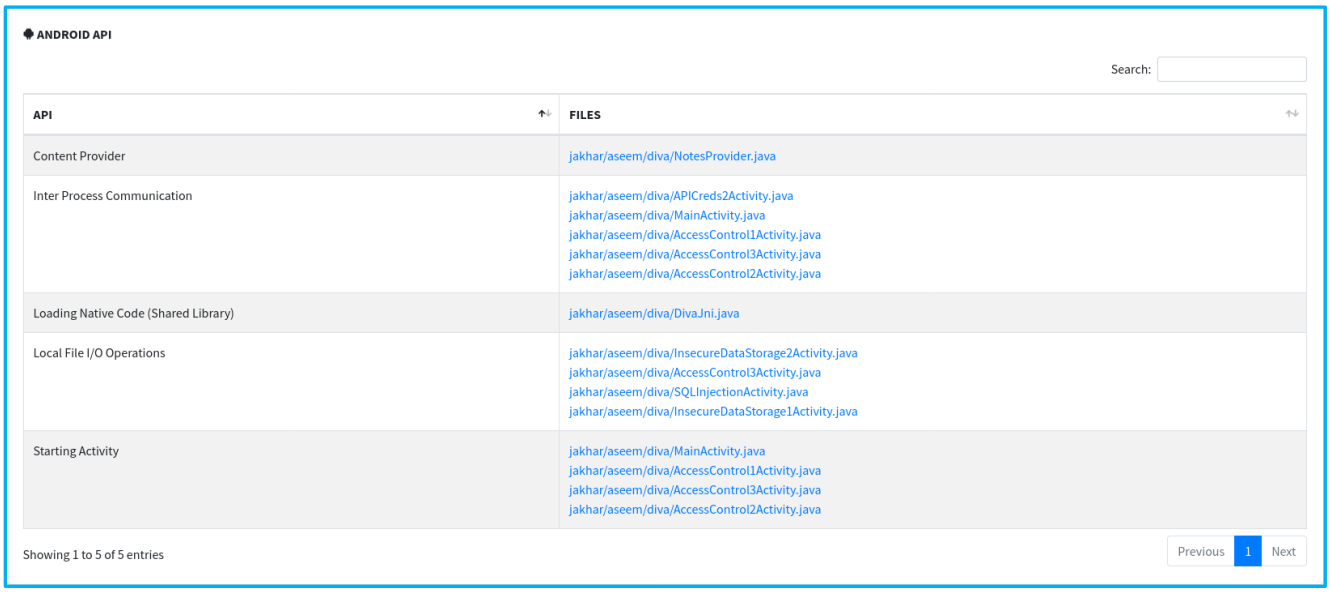
Search:

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows an application to read from external storage.
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.

Showing 1 to 3 of 3 entries

Previous 1 Next

Android API



ANDROID API

Search:

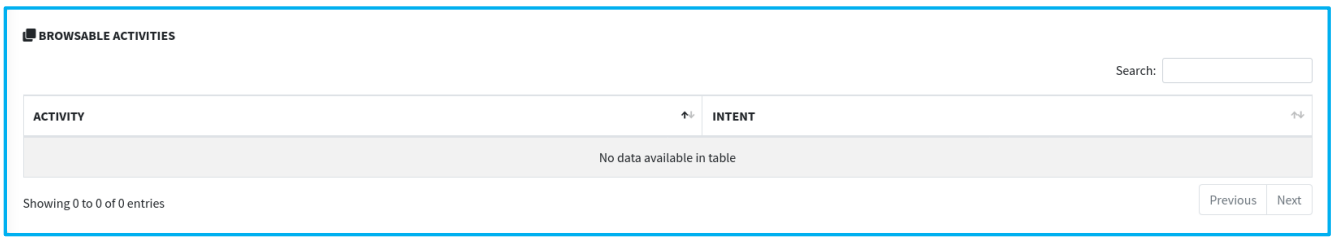
API	FILES
Content Provider	jakhar/aseem/diva/NotesProvider.java
Inter Process Communication	jakhar/aseem/diva/APICreds2Activity.java jakhar/aseem/diva/MainActivity.java jakhar/aseem/diva/AccessControl1Activity.java jakhar/aseem/diva/AccessControl3Activity.java jakhar/aseem/diva/AccessControl2Activity.java
Loading Native Code (Shared Library)	jakhar/aseem/diva/DivaJni.java
Local File I/O Operations	jakhar/aseem/diva/InsecureDataStorage2Activity.java jakhar/aseem/diva/AccessControl3Activity.java jakhar/aseem/diva/SQLInjectionActivity.java jakhar/aseem/diva/InsecureDataStorage1Activity.java
Starting Activity	jakhar/aseem/diva/MainActivity.java jakhar/aseem/diva/AccessControl1Activity.java jakhar/aseem/diva/AccessControl3Activity.java jakhar/aseem/diva/AccessControl2Activity.java

Showing 1 to 5 of 5 entries

Previous 1 Next

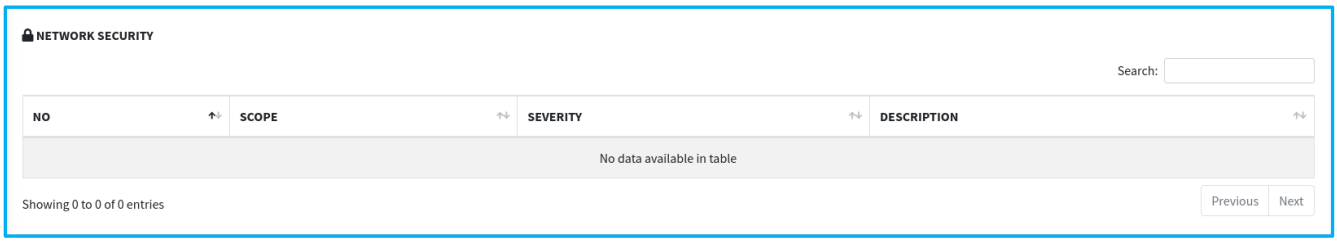
Browsable Activities

This shows all the activities that have implemented a deep link schema.



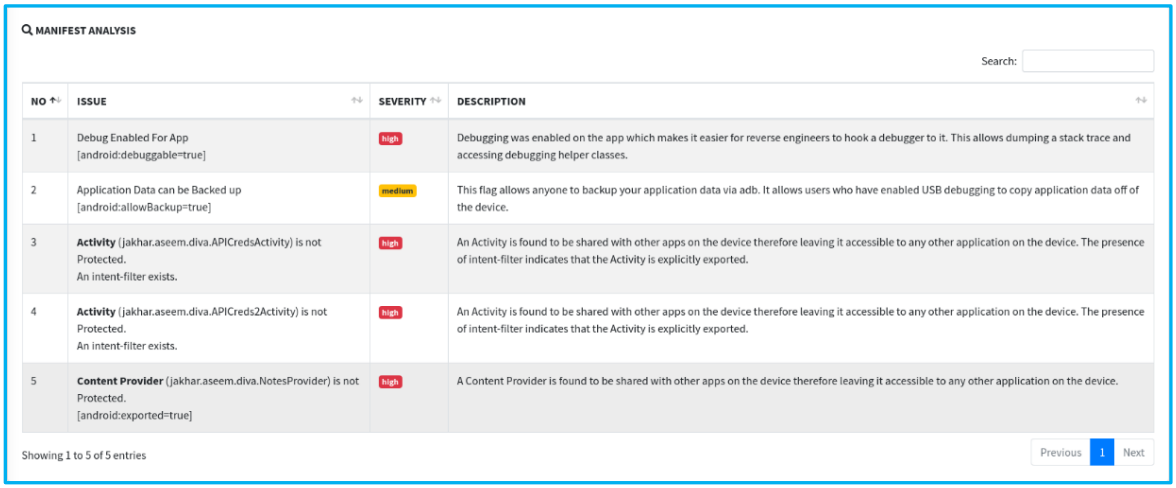
Network Security

This reveals some dangerous vulnerabilities, such as sending clear text traffic and trusting user certificates without verification. In here, we can find out information about network security issues related to the application.



Manifest Analysis

This reveals the security flaws found in the target application. Additionally, information from the android manifest file shows **which activities are exported, if the app debuggable or not, data schemas etc.**



MANIFEST ANALYSIS

Search:

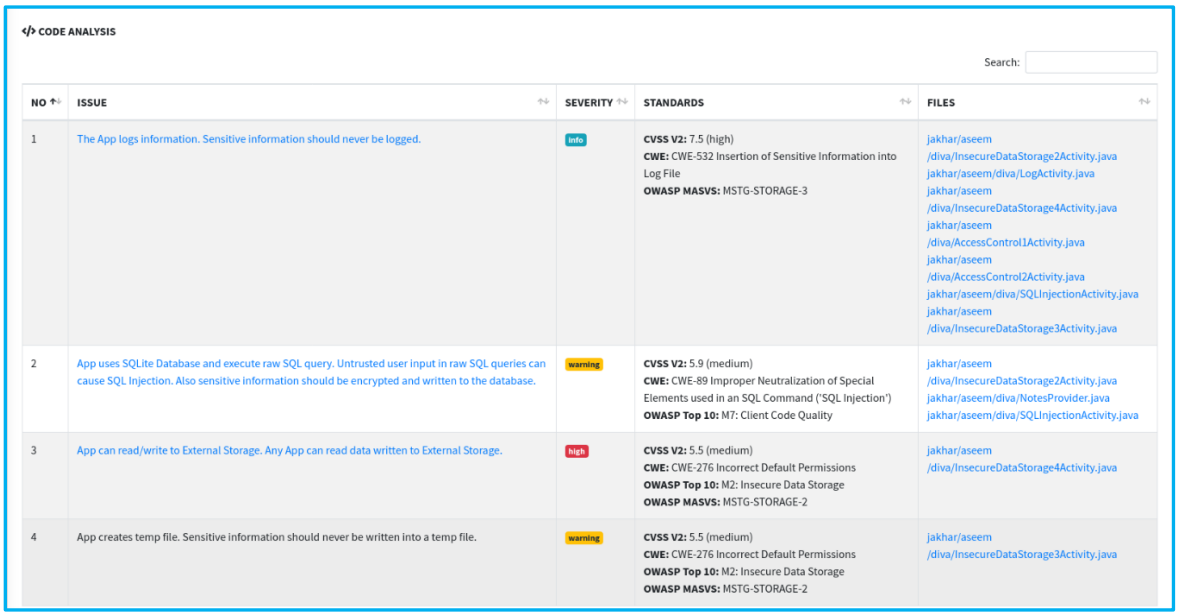
NO	ISSUE	SEVERITY	DESCRIPTION
1	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.
2	Application Data can be Backed up [android:allowBackup=true]	medium	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.
3	**Activity** (jakhar.aseem.diva.APICredsActivity) is not Protected. An intent-filter exists.	high	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	**Activity** (jakhar.aseem.diva.APICreds2Activity) is not Protected. An intent-filter exists.	high	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.
5	**Content Provider** (jakhar.aseem.diva.NotesProvider) is not Protected. [android:exported=true]	high	A Content Provider is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.

Showing 1 to 5 of 5 entries

Previous 1 Next

Code Analysis

In here the code has been analyzed and compared some behavior of the application based on industry security standard practices like OWASP MSTG and mapped the vulnerabilities with OWASP Top 10.



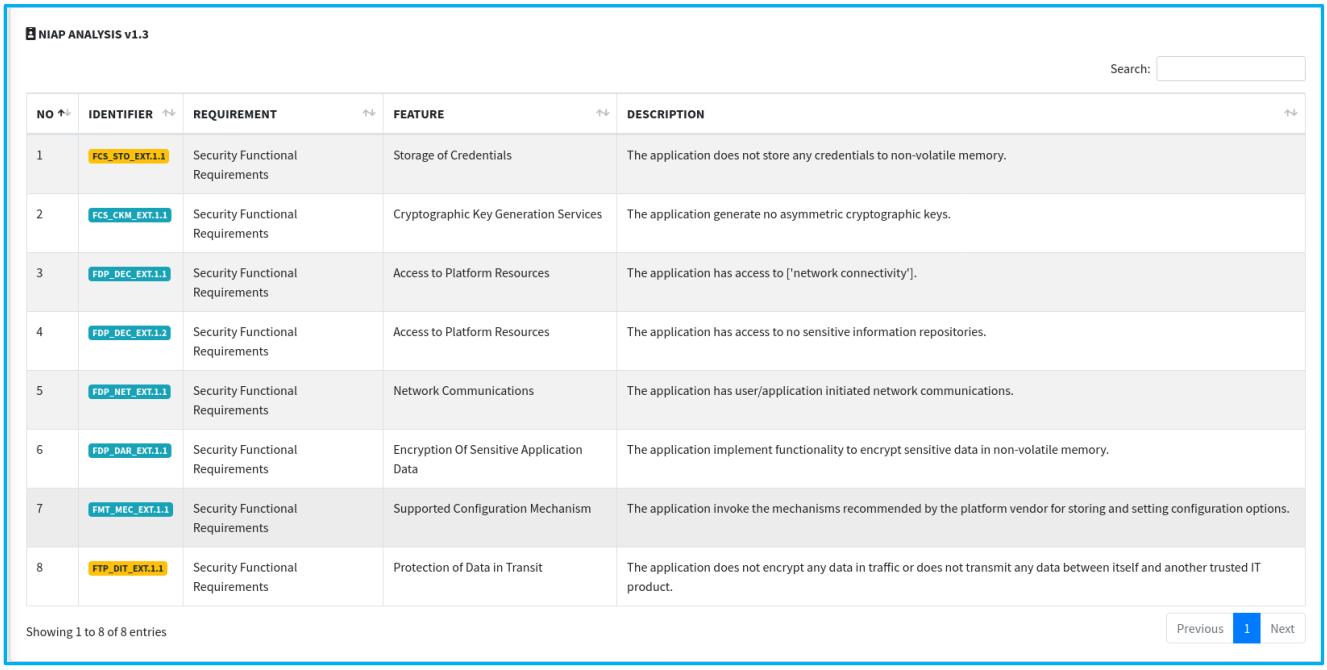
CODE ANALYSIS

Search:

NO	ISSUE	SEVERITY	STANDARDS	FILES
1	The App logs information. Sensitive information should never be logged.	info	**CVSS V2:** 7.5 (high) **CWE:** CWE-532 Insertion of Sensitive Information into Log File **OWASP MASVS:** MSTG-STORAGE-3	jakhar/aseem/diva/InsecureDataStorage2Activity.java jakhar/aseem/diva/LogActivity.java jakhar/aseem/diva/InsecureDataStorage4Activity.java jakhar/aseem/diva/AccessControl1Activity.java jakhar/aseem/diva/AccessControl2Activity.java jakhar/aseem/diva/SQLInjectionActivity.java jakhar/aseem/diva/InsecureDataStorage3Activity.java
2	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.	warning	**CVSS V2:** 5.9 (medium) **CWE:** CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') **OWASP Top 10:** M7: Client Code Quality	jakhar/aseem/diva/InsecureDataStorage2Activity.java jakhar/aseem/diva/NotesProvider.java jakhar/aseem/diva/SQLInjectionActivity.java
3	App can read/write to External Storage. Any App can read data written to External Storage.	high	**CVSS V2:** 5.5 (medium) **CWE:** CWE-276 Incorrect Default Permissions **OWASP Top 10:** M2: Insecure Data Storage **OWASP MASVS:** MSTG-STORAGE-2	jakhar/aseem/diva/InsecureDataStorage4Activity.java
4	App creates temp file. Sensitive information should never be written into a temp file.	warning	**CVSS V2:** 5.5 (medium) **CWE:** CWE-276 Incorrect Default Permissions **OWASP Top 10:** M2: Insecure Data Storage **OWASP MASVS:** MSTG-STORAGE-2	jakhar/aseem/diva/InsecureDataStorage3Activity.java

NIAP Analysis

This shows the app's adherence to National Information Assurance Partnership (NIAP) security policy.



NIAP ANALYSIS v1.3

Search:

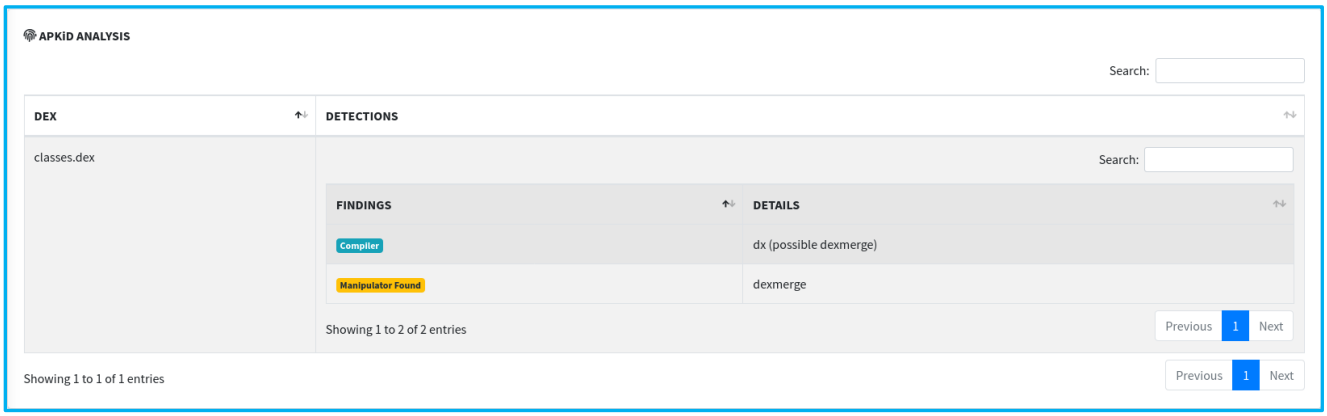
NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION
1	FCS_STO_EXT.1.1	Security Functional Requirements	Storage of Credentials	The application does not store any credentials to non-volatile memory.
2	FCS_CKM_EXT.1.1	Security Functional Requirements	Cryptographic Key Generation Services	The application generate no asymmetric cryptographic keys.
3	FDP_DEC_EXT.1.1	Security Functional Requirements	Access to Platform Resources	The application has access to ['network connectivity'].
4	FDP_DEC_EXT.1.2	Security Functional Requirements	Access to Platform Resources	The application has access to no sensitive information repositories.
5	FDP_NET_EXT.1.1	Security Functional Requirements	Network Communications	The application has user/application initiated network communications.
6	FDP_DAR_EXT.1.1	Security Functional Requirements	Encryption Of Sensitive Application Data	The application implement functionality to encrypt sensitive data in non-volatile memory.
7	FMT_MEC_EXT.1.1	Security Functional Requirements	Supported Configuration Mechanism	The application invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.
8	FTP_DIT_EXT.1.1	Security Functional Requirements	Protection of Data in Transit	The application does not encrypt any data in traffic or does not transmit any data between itself and another trusted IT product.

Showing 1 to 8 of 8 entries

Previous 1 Next

APK ID Analysis

APKiD is an open-source tool that is very helpful to identify various packers, compilers, obfuscators etc. in android files.



APKiD ANALYSIS

Search:

DEX	DETECTIONS
classes.dex	Search:

FINDINGS	DETAILS
Compiler	dx (possible dexmerge)
Manipulator Found	dexmerge

Showing 1 to 2 of 2 entries

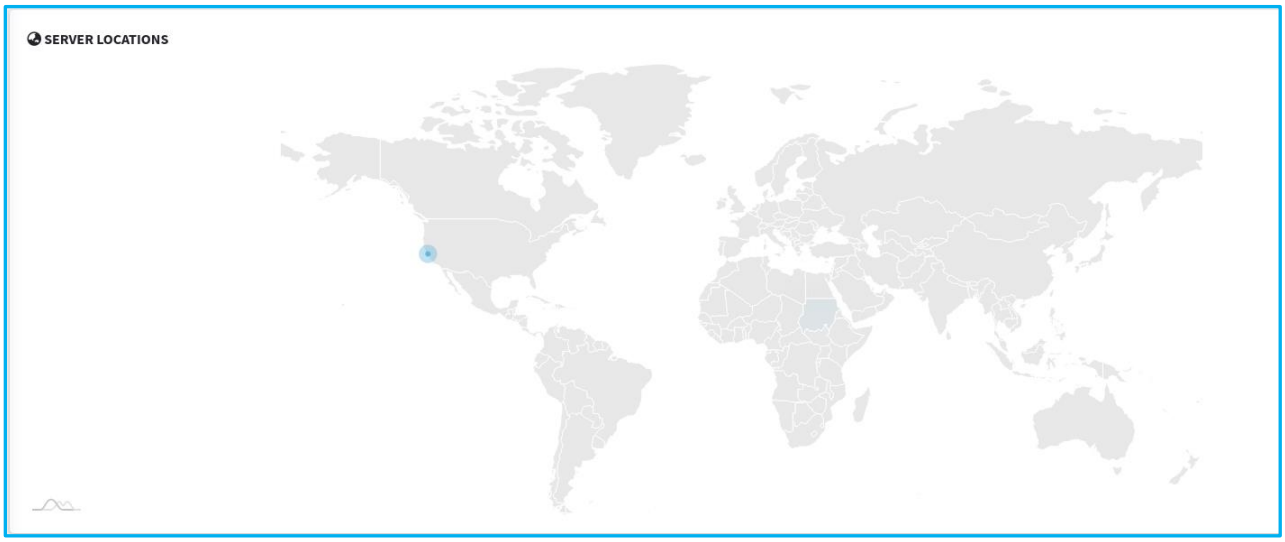
Previous 1 Next

Showing 1 to 1 of 1 entries

Previous 1 Next

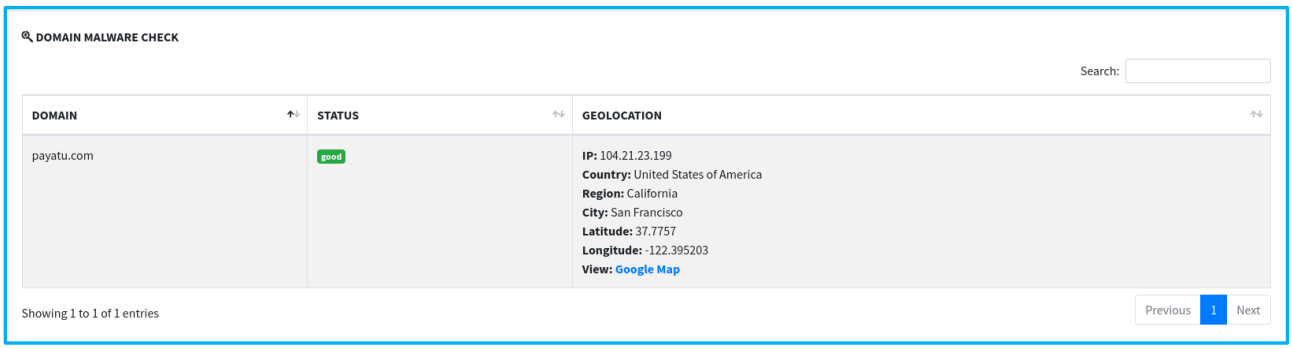
Server Location

It uses ip2location to give out its geolocation of the server.



Domain Malware Check

In here, it extracts all the URLs/IP addresses that are hard-coded or being used in the application and shows its malware status.



Hardcoded Secrets

In most of the times, developers have this habit of storing critical keys like AWS ID and credentials in strings.xml and use an object as a reference in java activity. However, as you can see, strings.xml can be decoded easily.

HARDCODED SECRETS

POSSIBLE SECRETS
"ids1_password" : "Enter 3rd party service password"
"ids1_user" : "Enter 3rd party service user name"
"pkey" : "notespin"

Risk Calculation

App Security Score Calculation

Every app is given an ideal score of 100 to begin with.
For every findings with severity high we reduce 15 from the score.
For every findings with severity warning we reduce 10 from the score.
For every findings with severity good we add 5 to the score.
If the calculated score is greater than 100, then the app security score is considered as 100.
And if the calculated score is less than 0, then the app security score is considered as 10.

Risk Calculation

APP SECURITY SCORE	RISK
0 - 15	CRITICAL
16 - 40	HIGH
41 - 70	MEDIUM
71 - 100	LOW

References

https://www.hackingloops.com/mobile-application-vulnerabilities-mobsf/

https://www.netguru.com/codestories/android-security-analysis-tools-part-four-mobsf

https://www.hackingarticles.in/android-pentest-automated-analysis-using-mobsf/

Dynamic Analysis

When it comes to the dynamic analysis, it performs the scan according to how user navigates into the Application. Additionally, it needs an Android VM to run its services.

In order to perform the dynamic analysis, the host machine ***(Where mobSF running)*** should be a **standalone machine**. It will not allow us to setup mobSF inside a Virtual machine. So due to that reason I am going to install mobSF in my Windows 10 host machine itself.

However, there are several requirements that need to be fulfilled before beginning the dynamic analysis procedure.

Requirements

1) Virtual Box
2) Genymotion

- Install Git
- Install Python 3.8-3.9
- Install JDK 8+
- Install Microsoft Visual C++ Build Tools
- Install OpenSSL (non-light)
- Download & Install wkhtmltopdf as per the wiki instructions
- Add the folder that contains `wkhtmltopdf` binary to environment variable PATH.

Installation of mobSF

As the very first step, you need to fulfill the above requirements.

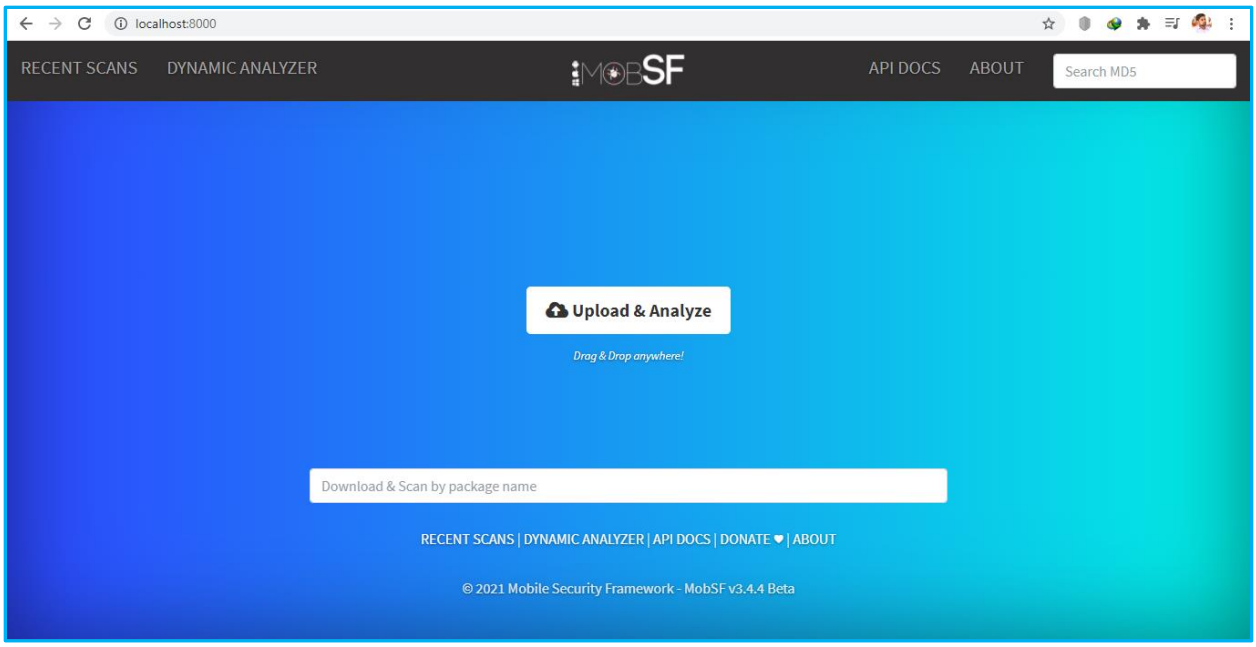
```
C:\Users\DILSHAN\Downloads\Programs>java -version
java version "16" 2021-03-16
Java(TM) SE Runtime Environment (build 16+36-2231)
Java HotSpot(TM) 64-Bit Server VM (build 16+36-2231, mixed mode, sharing)

C:\Users\DILSHAN\Downloads\Programs>git --version
git version 2.32.0.windows.1

C:\Users\DILSHAN\Downloads\Programs>python -V
Python 3.9.5

C:\Users\DILSHAN\Downloads\Programs>pip -V
pip 21.1.1 from c:\users\dilshan\appdata\local\programs\python\python39\lib\site-packages\pip (python 3.9)
```

```
C:\Users\DILSHAN\Downloads\Programs\Mobile-Security-Framework-MobSF>setup.bat
[INSTALL] Python is available
[INSTALL] Found Python 3.9.5
[INSTALL] Found pip
Requirement already satisfied: pip in c:\users\dilshan\appdata\local\programs\python\python39\lib\site-packages (21.1.1)
Collecting pip
  Downloading pip-21.1.2-py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 1.7 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.1.1
    Uninstalling pip-21.1.1:
      Successfully uninstalled pip-21.1.1
Successfully installed pip-21.1.2
[INSTALL] Found OpenSSL executable
[INSTALL] Found Visual Studio Build Tools
[INSTALL] Creating venv
Requirement already satisfied: pip in c:\users\dilshan\downloads\programs\mobile-security-framework-mobsf\venv\lib\site-packages (21.1.1)
Collecting pip
  Downloading pip-21.1.2-py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 1.7 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.1.1
    Uninstalling pip-21.1.1:
      Successfully uninstalled pip-21.1.1
Successfully installed pip-21.1.2
[INSTALL] Installing Requirements
Ignoring gunicorn: markers 'platform_system != "Windows"' don't match your environment
Collecting Django>=3.1.5
  Downloading Django-3.2.4-py3-none-any.whl (7.9 MB)
     |████████████████████████████████| 7.9 MB 1.7 MB/s
Collecting lxml>=4.6.2
  Downloading lxml-4.6.3-cp39-cp39-win_amd64.whl (3.5 MB)
     |████████████████████████████████| 3.5 MB 6.8 MB/s
Collecting rsa>=4.7
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Collecting biplist>=1.0.3
  Downloading biplist-1.0.3.tar.gz (21 kB)
Collecting requests>=2.25.1
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB ...
Collecting bs4>=0.0.1
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
Collecting colorlog>=4.7.2
  Downloading colorlog-5.0.1-py2.py3-none-any.whl (10 kB)
Collecting macholib>=1.14
  Downloading macholib-1.14-py2.py3-none-any.whl (37 kB)
Collecting whitenoise>=5.2.0
```

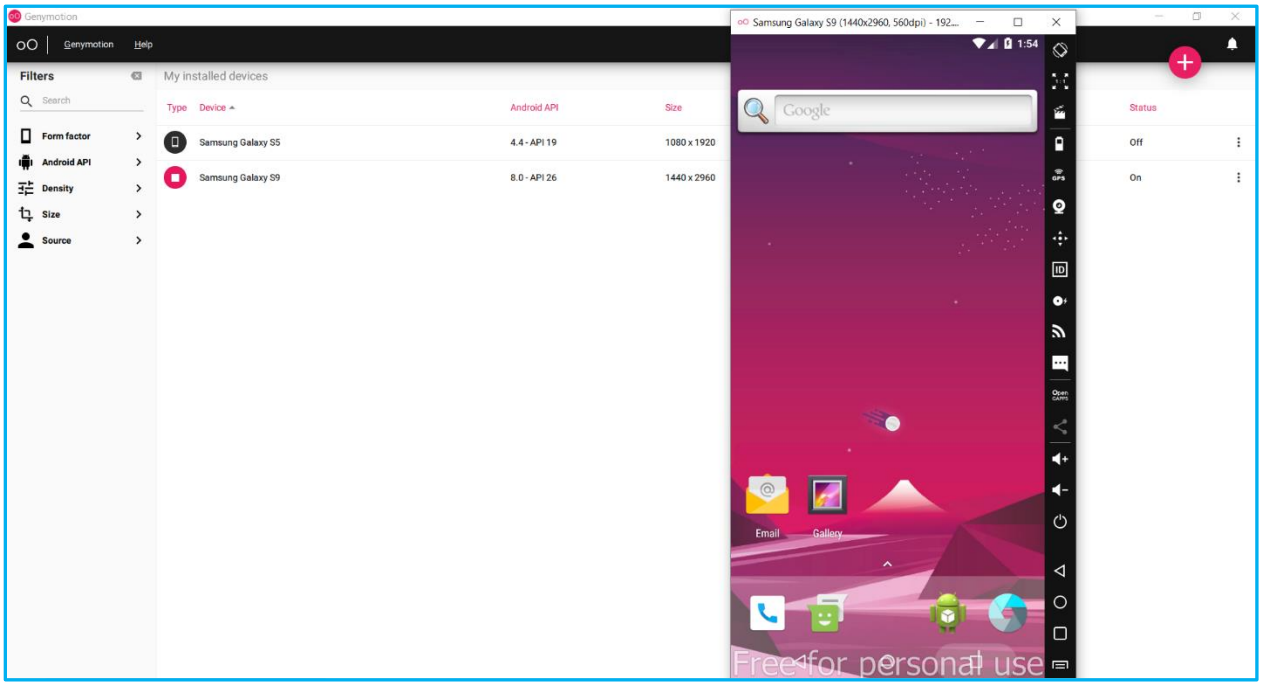
```
No changes detected
[INFO] 25/Jun/2021 06:33:01 - Checking for Update.
[INFO] 25/Jun/2021 06:33:01 - No updates available.
[INFO] 25/Jun/2021 06:33:02 -
 __  __       _     ____  _____              _____ _  _
 |  \/  | ___ | |__/ ___||  ___| __   _|__ /| || |
 | |\/| |/ _ \| '_ \___ \| |_    \ \ / / |_ \| || |_
 | |  | | (_) | |_) |__) |  _|    \ V / ___) |__   _|
 |_|  |_|\___/|_.__/____/|_|       \_/ |____(_) |_|

[INFO] 25/Jun/2021 06:33:02 - Mobile Security Framework v3.4.4 Beta
REST API Key: 16b25d1eeff934cb9c0bd97589c8dc5b710880ddd9041c45d537e008fc2ccca6
[INFO] 25/Jun/2021 06:33:02 - OS: Windows
[INFO] 25/Jun/2021 06:33:02 - Platform: Windows-10-10.0.19041-SP0
[INFO] 25/Jun/2021 06:33:02 - Dist:
[INFO] 25/Jun/2021 06:33:02 - MobSF Basic Environment Check
[WARNING] 25/Jun/2021 06:33:02 - 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] 25/Jun/2021 06:33:02 - Checking for Update.
[INFO] 25/Jun/2021 06:33:03 - No updates available.
[INFO] 25/Jun/2021 06:33:03 -
 __  __       _     ____  _____              _____ _  _
 |  \/  | ___ | |__/ ___||  ___| __   _|__ /| || |
 | |\/| |/ _ \| '_ \___ \| |_    \ \ / / |_ \| || |_
 | |  | | (_) | |_) |__) |  _|    \ V / ___) |__   _|
 |_|  |_|\___/|_.__/____/|_|       \_/ |____(_) |_|

[INFO] 25/Jun/2021 06:33:03 - Mobile Security Framework v3.4.4 Beta
REST API Key: 16b25d1eeff934cb9c0bd97589c8dc5b710880ddd9041c45d537e008fc2ccca6
[INFO] 25/Jun/2021 06:33:03 - OS: Windows
[INFO] 25/Jun/2021 06:33:04 - Platform: Windows-10-10.0.19041-SP0
[INFO] 25/Jun/2021 06:33:04 - Dist:
[INFO] 25/Jun/2021 06:33:04 - MobSF Basic Environment Check
[WARNING] 25/Jun/2021 06:33:04 - 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] 25/Jun/2021 06:33:04 - Checking for Update.
[INFO] 25/Jun/2021 06:33:04 - No updates available.
Download and Install wkhtmltopdf for PDF Report Generation - https://wkhtmltopdf.org/downloads.html
[INSTALL] Installation Complete
```

After successfully installing mobSF, you can run it by giving the command **“run.bat”**.

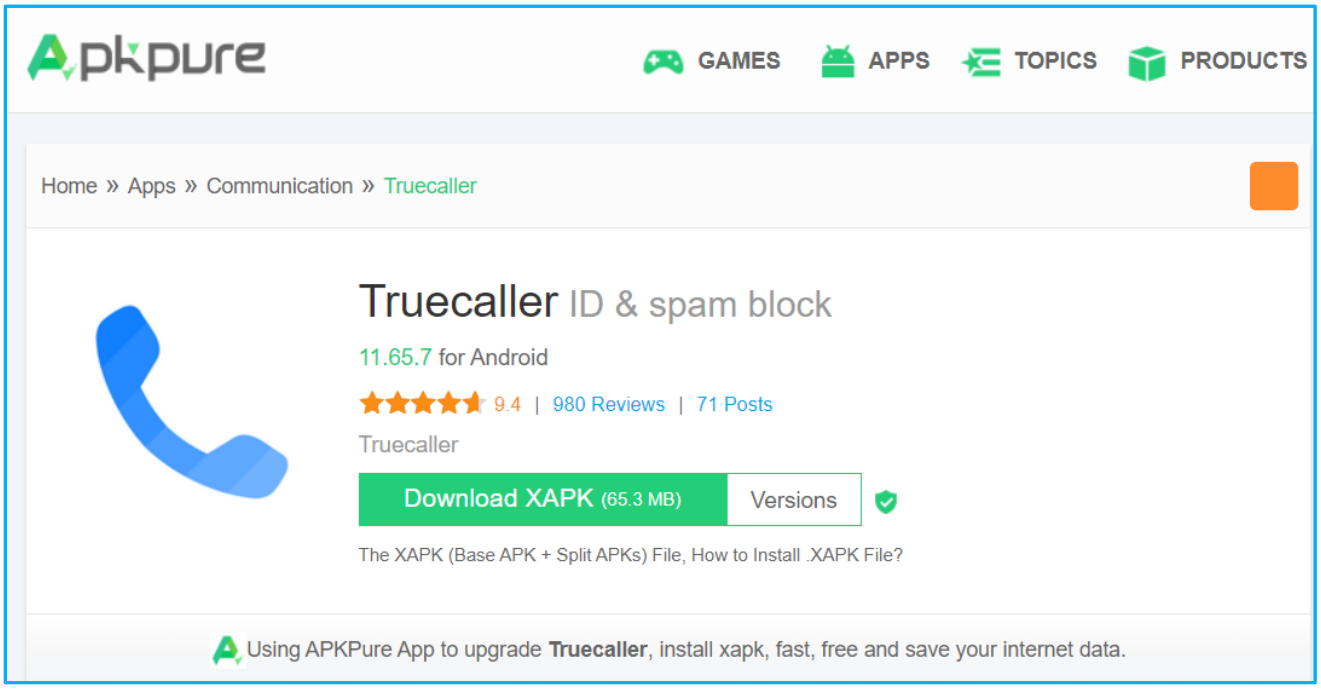
```
C:\Users\DILSHAN\Downloads\Programs\Mobile-Security-Framework-MobSF>run.bat
Running MobSF on 0.0.0.0:8000
[INFO] 25/Jun/2021 06:37:32 -
 __  __       _     ____  _____              _____ _  _
 |  \/  | ___ | |__/ ___||  ___| __   _|__ /| || |
 | |\/| |/ _ \| '_ \___ \| |_    \ \ / / |_ \| || |_
 | |  | | (_) | |_) |__) |  _|    \ V / ___) |__   _|
 |_|  |_|\___/|_.__/____/|_|       \_/ |____(_) |_|

[INFO] 25/Jun/2021 06:37:32 - Mobile Security Framework v3.4.4 Beta
REST API Key: 16b25d1eeff934cb9c0bd97589c8dc5b710880ddd9041c45d537e008fc2ccca6
[INFO] 25/Jun/2021 06:37:32 - OS: Windows
[INFO] 25/Jun/2021 06:37:32 - Platform: Windows-10-10.0.19041-SP0
[INFO] 25/Jun/2021 06:37:32 - Dist:
[INFO] 25/Jun/2021 06:37:32 - MobSF Basic Environment Check
[WARNING] 25/Jun/2021 06:37:33 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
[INFO] 25/Jun/2021 06:37:33 - Checking for Update.
[INFO] 25/Jun/2021 06:37:33 - No updates available.
```



Before proceeding further with the mobSF, first you need to set-up & configure the Genymotion to launch Virtual device. In here, you need to run a Genymotion Android VM before starting mobSF. Then every other thing will be configured automatically at the runtime. Usually, Android 5.0–9.0 versions use Frida and works out of the box with zero configuration or setup. So, in my case, I am going to use **Samsung Galaxy S9** virtual device as my android VM.

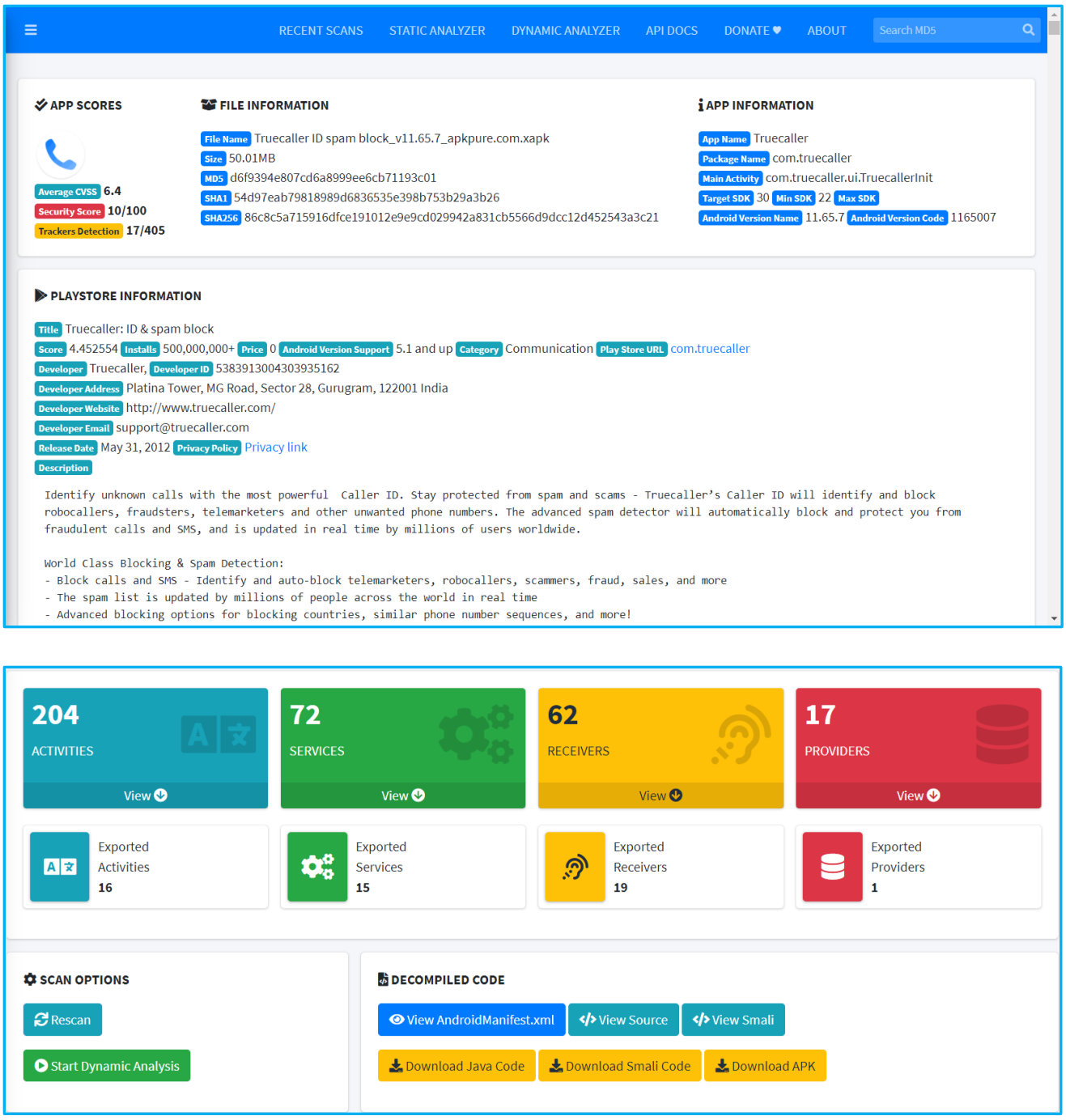


In order to perform the dynamic analysis, I have chosen the android application **“TrueCaller”** APK.

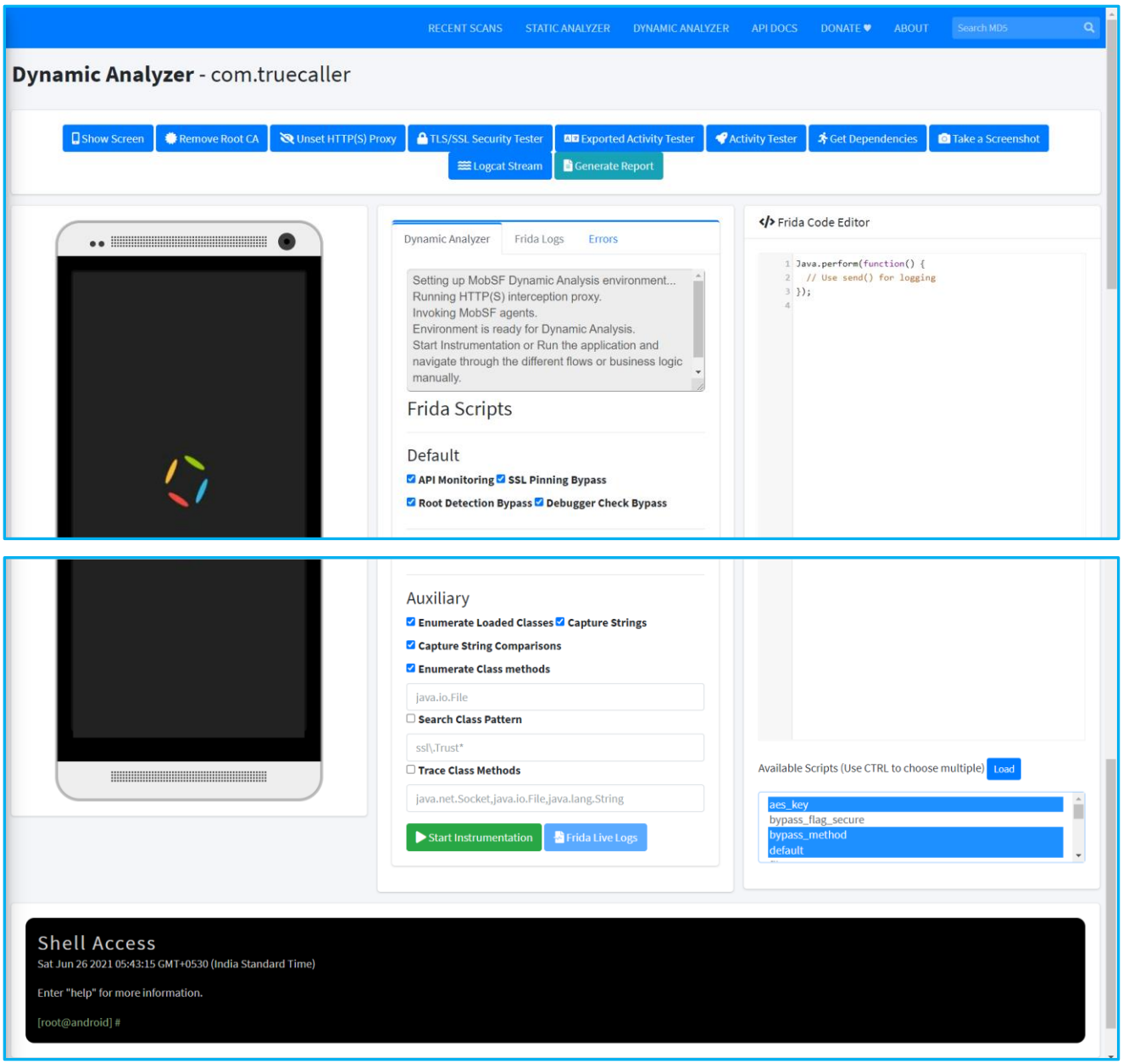


In here also, at the very beginning you need to follow the same procedure as in the static analysis.

So, as you can see in below images, the mobSF tool automatically performs the static analysis first for that particular android application.



In order to perform dynamic analysis, you need to press the **"Dynamic Analyzer"** option present on the top navigation pane. Then mobSF will automatically attach itself to currently running VM if mobSF and Genymotion are running on the same base machine.



As you can see, under the **"Dynamic Analyzer"** tab, there are various default **frida** scripts that are available.

Those scripts can be very useful specially when checking for vulnerabilities such as SSL Pinning bypass and Root detection checks.

What is Frida ?

Frida is a dynamic instrumentation toolkit that is used by various researchers to intercept IPC and modify it to make a function perform the desired function. This process is also known as **“android hooking”.**

Frida uses javascript to perform hooking since Android’s native code and javascript both run on JIT compilation techniques, it can intercept its inter-process communication, add the code specified in a script and completely change the function’s implementation.

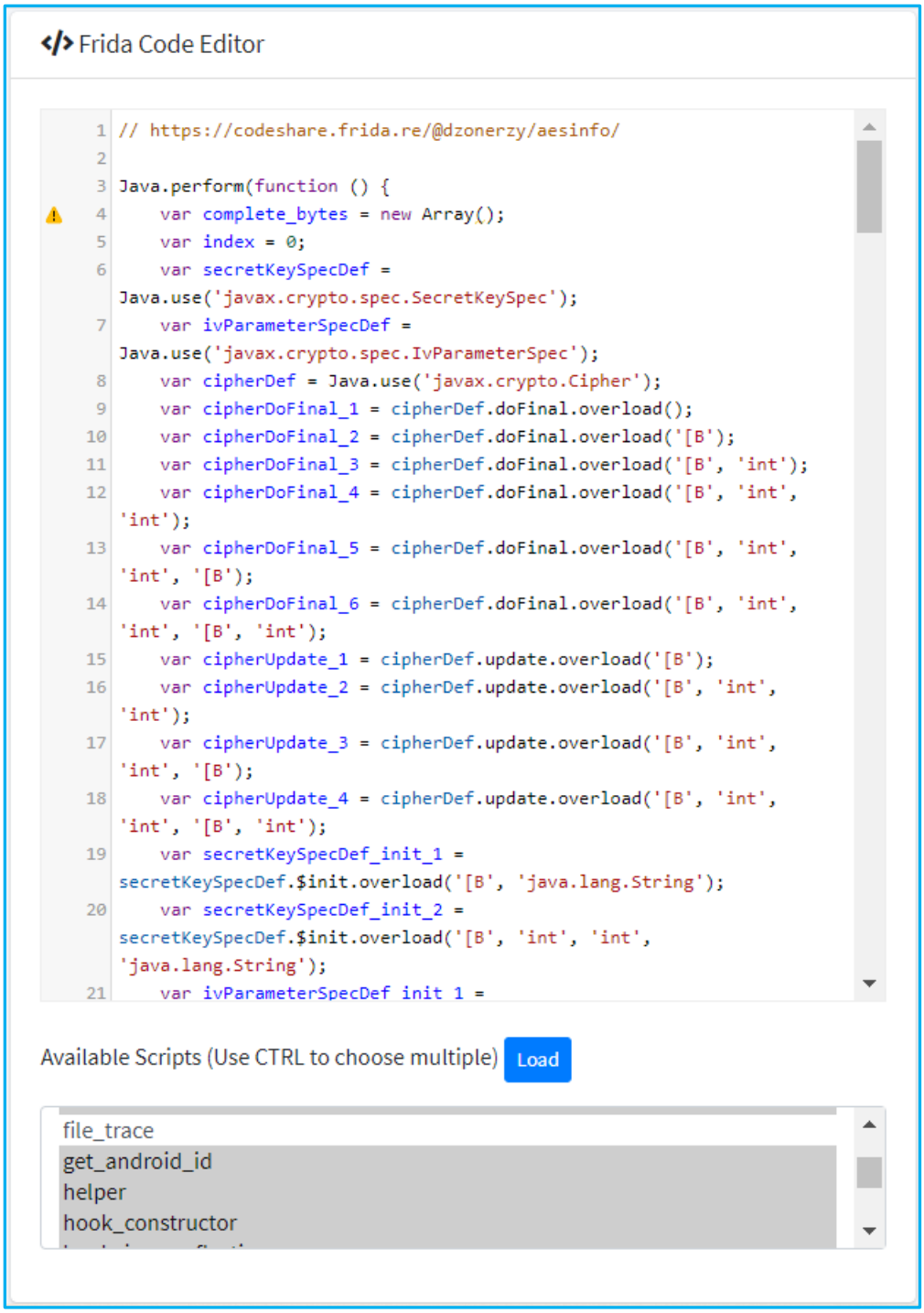
Some of the main use cases of Frida are as following.

1) Spy on Crypto APIs
2) Modify function’s output
3) Bypass AES encryption
4) Bypass SSL Pinning and Root detection
5) Trace private application code
6) Bypass various software sided locks (like Applock)

MobSF is not limited to those scripts only. There are other auxiliary scripts which allows to enumerate various classes and capture string comparisons in real time.

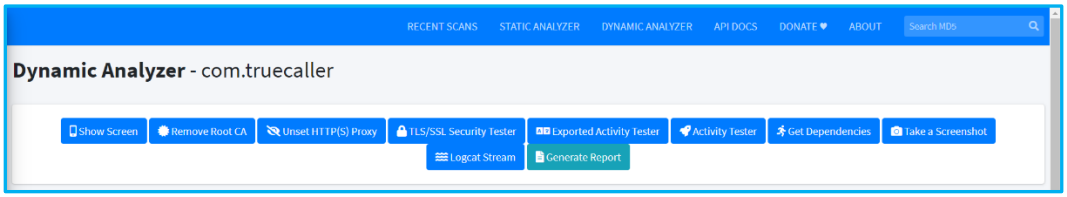
Then automatically the selected scripts will be attached to the application. So, in my case as you can see in the above images, I have chosen all of the Default Frida Scripts & Auxiliary Scripts.

Additionally, there are multiple available scripts too. We can load multiple scripts at once by holding the CTRL. So, you can see those loaded scripts in the **Frida Code Editor**.



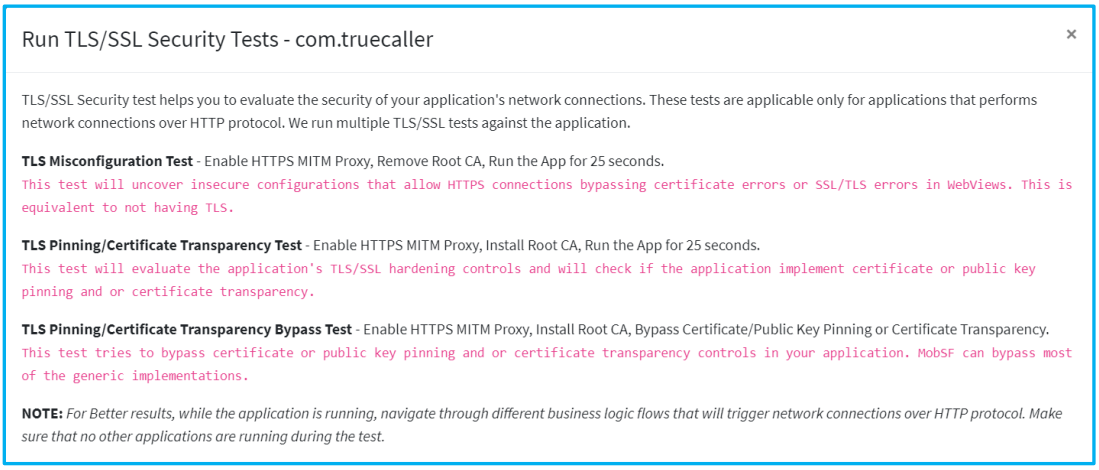
As the next step you need to click on **“Start Instrumentation”.**

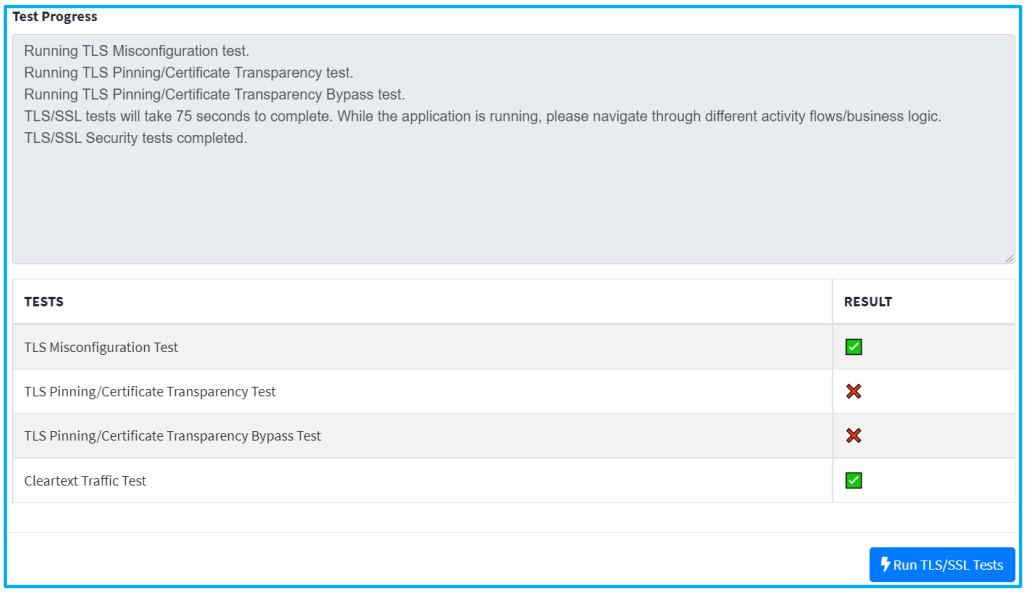
As I have mentioned previously, we can perform several additional tests on our chosen android application.



Now I will demonstrate some of those tests.

TLS/SSL Security Tester





As you can see in the above images, TLS/SSL Security test helps in evaluating the security of our application's network connections.

However, those tests are applicable only for applications that performs network connections over HTTP protocol.

Following are some of the TLS/SSL tests against the application.

1) TLS Misconfiguration Test

- This enables HTTPS MITM Proxy, Remove Root CA and Run the app for 25 seconds.
- Additionally, this test is capable of uncovering insecure configurations which allow HTTPS connections bypassing certificate errors or SSL/TLS errors in WebViews. In general, this is equivalent to not having TLS.

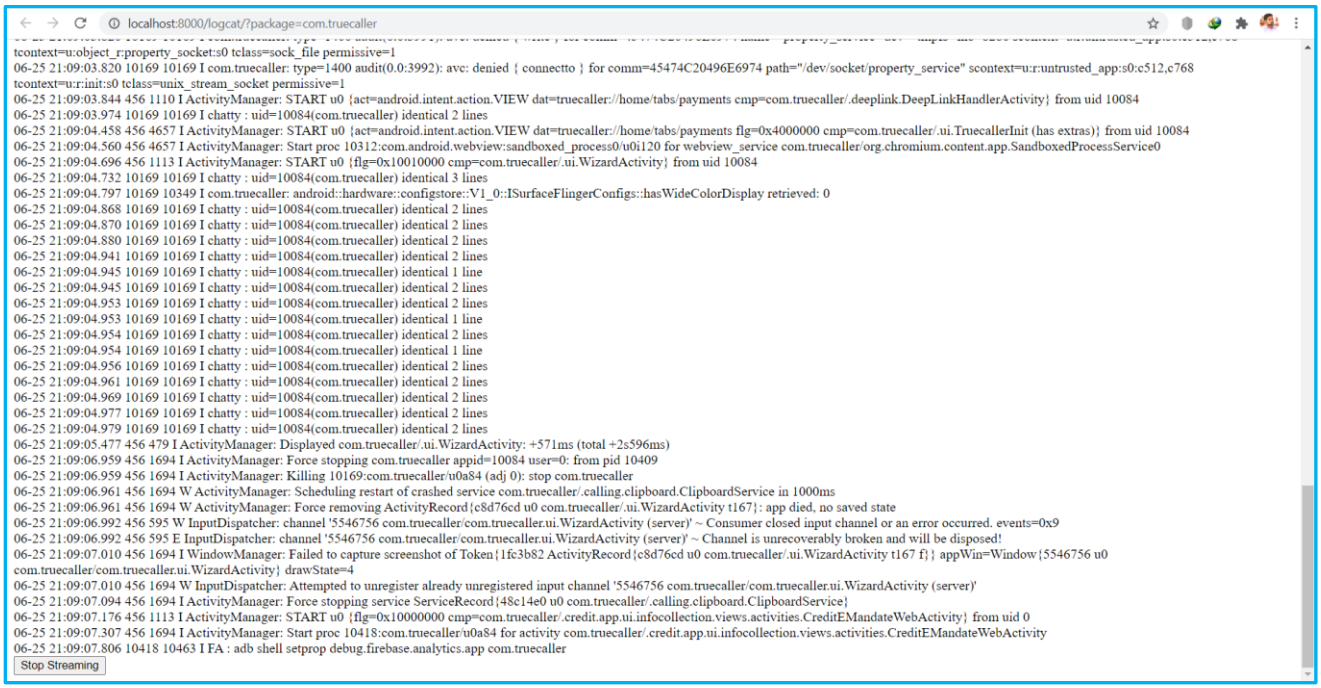
2) TLS Pinning/Certificate Transparency Test

- This enables HTTPS MITM Proxy, Install Root CA and Run the app for 25 seconds.
- Additionally, this test is capable of evaluating the application's TLS/SSL hardening controls and will check if the application implement certificate or public key pinning or certificate transparency.

3) TLS Pinning/Certificate Transparency Bypass Test

- This enables HTTPS MITM Proxy, Install Root CA and Bypass Certificate/Public Key Pinning or Certificate Transparency.
- Additionally, this test is capable of bypassing certificate or public key pinning and certificate transparency controlling. MobSF can bypass most of the generic implementations.

Logcat Stream



Logcat is used to dump a log of system messages, including stack traces when the device throws an error and messages that you have written from your app with the **Log** class. Logcat can also be viewed in MobSF's native environment.

Generating Reports

After you have done with the analysis, you can download the report by sliding the menu bar slider on the left-hand side and click generate the report.

So, let's examine & analyze the final results we have got.

Frida Logs

Frida Logs - com.truecaller

Data refreshed in every 10 seconds.

```
Loaded Frida Script - debugger_check_bypass
Loaded Frida Script - root_bypass
Loaded Frida Script - ssl_pinning_bypass
[SSL Pinning Bypass] okhttp CertificatePinner not found
[SSL Pinning Bypass] okhttp3 CertificatePinner not found
[SSL Pinning Bypass] DataTheorem trustkit not found
[SSL Pinning Bypass] Appcelerator PinningTrustManager not found
[SSL Pinning Bypass] Apache Cordova SSLCertificateChecker not found
[SSL Pinning Bypass] Wultra CertStore.validateFingerprint not found
[SSL Pinning Bypass] verifyCertificateChain() bypassed
[SSL Pinning Bypass] verifyCertificateChain() bypassed
[SSL Pinning Bypass] verifyCertificateChain() bypassed
[RootDetection Bypass] test-keys check
```

Frida log shows the activities that are done by their scripts.

HTTP(S) Traffic

```
========
REQUEST
========
GET https://graph.facebook.com/v6.0/126694440681943?
fields=supports_implicit_sdk_logging%2Cgdpv4_nux_content%2Cgdpv4_nux_enabled%2Candroid_dialog_configs%2Candroid_sdk_error_categories%2Capp_events_session_timeout%2Capp_events_feature_bitmask%2Cauto_event_mapping_android%2Cseamless_login%2Csmart_login_bookmark_icon_url%2Csmart_login_menu_icon_url%2Crestrictive_data_filter_params%2Caam_rules%2Csuggested_events_setting&format=json&sdk=android HTTP/1.1
User-Agent: FBAndroidSDK.7.1.0
Accept-Language: en_US
Content-Type: application/x-www-form-urlencoded
Content-Encoding: gzip
Host: graph.facebook.com
Connection: Keep-Alive
Accept-Encoding: gzip
========
RESPONSE
========
HTTP/1.1 200 OK
Content-Encoding: gzip
ETag: "ac5afa3e09fe85f2311df864e5657df008963421"
Content-Type: text/javascript; charset=UTF-8
Vary: Origin, Accept-Encoding
x-fb-rlafr: 0
Access-Control-Allow-Origin: *
facebook-api-version: v6.0
Strict-Transport-Security: max-age=15552000; preload
Pragma: no-cache
Cache-Control: private, no-cache, no-store, must-revalidate
Expires: Sat, 01 Jan 2000 00:00:00 GMT
x-fb-request-id: AdC9XciZi3SbreIoy-n_M5B
x-fb-trace-id: BVXHrew4s0t
x-fb-rev: 1004041549
X-FB-Debug: wcxiuvsJv7XaOOHqZLF9l/NZzUH24fi9swIPgxPqTOh/Nu/6g9AU2MlZXg0A4dV0aPWi6pY0vkNqsxa1xFB6mw==
Date: Sat, 26 Jun 2021 00:59:27 GMT
Alt-Svc: h3-29=":443"; ma=3600,h3-27=":443"; ma=3600
Connection: keep-alive
Content-Length: 7476


b'{"supports_implicit_sdk_logging":true,"gdpv4_nux_enabled":false,"android_sdk_error_categories":[{"name":"login_recoverable","items":[{"code":102},{"code":190}],"recovery_message":"Please log into this app
again to reconnect your Facebook
account."}],"app_events_session_timeout":60,"app_events_feature_bitmask":67077,"seamless_login":1,"smart_login_bookmark_icon_url":"https:\\/\\/static.xx.fbcdn.net\\/rsrc.php\\/v3\\/ys\\/r\\/C6ZutYDSaaV.png","sm
art_login_menu_icon_url":"https:\\/\\/static.xx.fbcdn.net\\/rsrc.php\\/v3\\/ys\\/r\\/0iarpnudmEx.png","restrictive_data_filter_params":"{\\"AttestationNonceRequested\\":{\\"deprecated_param\\":
[\\"MobileService\\",\\"Latenc9l\\",\\"_inBackground\\",\\"OlogTime\\",\\"_eventName_ed5l\\",\\"_uI\\",\\"_inBackGround\\",\\"_inBackgroqnd\\"]},\\"AttestationVerified\\":{\\"deprecated_param\\":
[\\"MobileService\\",\\"_lmgTime\\",\\"_inBackground\\",\\"OlogTime\\",\\"FullQatency\\",\\"_eventName_ed5l\\",\\"_uI\\",\\"_inBackGround\\",\\"_inBackgroqnd\\"]},\\"AttestationPerformed\\":
{\\"deprecated_param\\":[\\"MobileService\\",\\"Wui\\",\\"_inBackGround\\",\\"OlogTime\\",\\"_eventName_ed5l\\",\\"_uI\\",\\"_inBackGround\\",\\"_inBackgroqnd\\"]},\\"ViewVisited\\":{\\"deprecated_param\\":
[\\"_5l\\",\\"_inBackcround\\",\\"_logThme\\",\\"RiewId\\",\\"_eventName_m15l\\",\\"_se\\",\\"_inBaciground\\",\\"_uy\\",\\"Cojtext\\",\\"_eventName_m
5l\\",\\"_ua\\",\\"_logTiie\\",\\"_ioBackground\\",\\"_sessioF_id\\",\\"_inBackgrotnd\\",\\"RegistrationState\\"]},\\"SettingState\\":{\\"deprecated_param\\":
[\\"IncallUI\\",\\"MissgdCallNotification\\",\\"MissedCallNotification\\"]},\\"PermissionState\\":{\\"deprecated_param\\":
```

This shows all the HTTPS(S) traffic that has been captured.

Dumpsys Logs

```
Currently running services:
  DockObserver
  Genyd
  SurfaceFlinger
  SystemPatcher
  accessibility
  account
  activity
  alarm
  android.security.keystore
  android.service.gatekeeper.IGateKeeperService
  appops
  appwidget
  audio
  backup
  battery
  batteryproperties
  batterystats
  carrier_config
  clipboard
  commontime_management
  connectivity
  connmetrics
  consumer_ir
  content
  contexthub
  country_detector
  cpuinfo
  dbinfo
  device_identifiers
  device_policy
  deviceidle
  devicestoragemonitor
  diskstats
  display
  dreams
  drm.drmManager
  dropbox
  ethernet
  gfxinfo
  gpu
  graphicsstats
  hardware_properties
  imms
  input
  input_method
  installd
  iphonesubinfo
  isms
```

```
DUMP OF SERVICE DockObserver:
Current Dock Observer Service state:
  reported state: 0
  previous state: 0
  actual state: 0
--------- 0.001s was the duration of dumpsys DockObserver, ending at: 2021-06-25 21:10:25
-------------------------------------------------------------------------------
DUMP OF SERVICE Genyd:
--------- 0.001s was the duration of dumpsys Genyd, ending at: 2021-06-25 21:10:25
-------------------------------------------------------------------------------
DUMP OF SERVICE SurfaceFlinger:
Build configuration: [sf HAS_CONTEXT_PRIORITY=0 DISABLE_TRIPLE_BUFFERING PRESENT_TIME_OFFSET=0 FORCE_HWC_FOR_RBG_TO_YUV=0 MAX_VIRT_DISPLAY_DIM=0 RUNNING_WITHOUT_SYNC_FRAMEWORK=0
NUM_FRAMEBUFFER_SURFACE_BUFFERS=2] [libui] [libgui]
Sync configuration: [using: EGL_KHR_fence_sync]
DispSync configuration: app phase 1000000 ns, sf phase 1000000 ns, present offset 0 ns (refresh 16666666 ns)

Static screen stats:
  < 1 frames: 103.024 s (7.9%)
  < 2 frames: 176.241 s (13.5%)
  < 3 frames: 38.195 s (2.9%)
  < 4 frames: 20.680 s (1.6%)
  < 5 frames: 18.643 s (1.4%)
  < 6 frames: 17.835 s (1.4%)
  < 7 frames: 12.516 s (1.0%)
  7+ frames: 922.086 s (70.4%)

Buffering stats:
  [Layer name] <Active time> <Two buffer> <Double buffered> <Triple buffered>
  [NavigationBar#0] 177.04 0.046 0.112 0.888
  [com.android.vending/com.google.android.finsky.activities.MainActivity#0] 28.65 0.000 0.920 0.080
  [com.truecaller/com.truecaller.ui.WizardActivity#0] 26.58 0.156 0.487 0.513
  [com.android.vending/com.android.vending.AssetBrowserActivity#0] 17.88 0.017 0.886 0.114
  [StatusBar#0] 16.30 0.545 0.906 0.094
  [com.google.android.gms/com.google.android.gms.auth.uiflows.minutemaid.MinuteMaidActivity#0] 7.73 0.425 0.980 0.020
  [com.google.android.gms/com.google.android.gms.auth.uiflows.addaccount.PreAddAccountActivity#0] 5.55 0.000 0.000 1.000
  [com.android.launcher3/com.android.launcher3.Launcher#0] 4.61 0.000 0.619 0.381
  [BootAnimation#0] 4.42 1.000 1.000 0.000
  [com.android.systemui/com.android.systemui.recents.RecentsActivity#0] 3.52 0.161 0.205 0.795
  [com.android.packageinstaller/com.android.packageinstaller.permission.ui.GrantPermissionsActivity#0] 2.52 0.000 0.728 0.272
  [com.truecaller/com.whizdm.okycverificationsdk.ui.activities.OkycVerificationActivity#0] 1.82 0.000 0.000 1.000
  [com.truecaller/com.truecaller.bizmon.newBusiness.onboarding.ui.OnboardingSuccessActivity#0] 1.49 0.000 0.000 1.000
  [com.google.android.gms/com.google.android.gms.setupservices.GoogleServicesActivity#0] 1.46 0.188 1.000 0.000
  [com.google.android.gms/com.google.android.gms.auth.uiflows.addaccount.AddAccountActivity#0] 1.43 0.000 1.000 0.000
  [com.truecaller/com.truecaller.tagger.TagPickActivity#0] 0.95 0.000 0.293 0.707
  [com.truecaller/com.truecaller.ui.WizardActivity#1] 0.88 0.924 0.924 0.076
  [com.android.vending/com.google.android.finsky.unauthenticated.UnauthenticatedMainActivity#0] 0.86 0.000 0.000 1.000
  [com.truecaller/com.truecaller.ui.WizardActivity#2] 0.79 1.000 1.000 0.000
  [com.truecaller/com.truecaller.whoviewedme.WhoViewedMeActivity#0] 0.76 0.144 0.660 0.340
  [com.truecaller/com.truecaller.tagger.tagPicker.TaggerActivity#0] 0.59 0.000 0.000 1.000
```

This shows all the currently running services & then dump them.

API Monitor

API Monitor - com.truecaller

Data refreshed in every 10 seconds.

Data Snip: 100

Search:

NAME	CLASS	METHOD	ARGUMENTS	RESULT	RETURN VALUE
Binder	android.app.Activity	startActivity	["",null]		
Binder	android.app.Activity	startActivity	[""]		
Binder	android.app.ContextImpl	registerReceiver	["",""		
Binder	android.app.ContextImpl	registerReceiver	[null,"",null,null]	"Intent { act=android.net.wifi.STATE_CHANGE flg=0x4000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	[null,""]	"Intent { act=android.net.wifi.STATE_CHANGE flg=0x4000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	[""		
Binder	android.app.ContextImpl	registerReceiver	[""		
Binder	android.app.ContextImpl	registerReceiver	["",""	"Intent { act=android.intent.action.PROXY_CHANGE flg=0x24000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	["",""	"Intent { act=android.intent.action.PROXY_CHANGE flg=0x24000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	[""	"Intent { act=android.net.conn.CONNECTIVITY_CHANGE flg=0x4000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	[""	"Intent { act=android.net.conn.CONNECTIVITY_CHANGE flg=0x4000010 (has extras) }"	
Binder	android.app.ContextImpl	registerReceiver	["",""		
Binder	android.app.ContextImpl	registerReceiver	["",""		
Binder	android.app.ContextImpl	registerReceiver	[""		
Binder	android.app.ContextImpl	registerReceiver	[""		
Crypto - Hash	java.security.MessageDigest	update	[[65,112,112,86,105,115,105,116,101,100],0,10]		
Crypto - Hash	java.security.MessageDigest	update	[[49,-123,-70,-40,-95,60,-54,81,81,-18,-67,25,-71,-114,7,97,-108,10,-93,-57,11,-58,11,17,-126,67,24,...		
Crypto - Hash	java.security.MessageDigest	update	[[69,84,65,45,73,78,70,47,100,101,98,117,103,45,108,111,103,103,105,110,103,95,114,101,108,101,97,11...		
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/databases/com.google.android.datatransport.events"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/databases/com.google.android.datatransport.events"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/no_backup/androidx.work.workdb"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	openDatabase	["/data/user/0/com.truecaller/no_backup/androidx.work.workdb",""	"SQLiteDatabase: /data/user/0/com.truecaller/no_backup/androidx.work.workdb"	
Database	android.database.sqlite.SQLiteDatabase	compileStatement	["PRAGMA user_version;"]	"SQLiteProgram: PRAGMA user_version;"	
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/databases/insights.db"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	openDatabase	["/data/user/0/com.truecaller/databases/insights.db",""	"SQLiteDatabase: /data/user/0/com.truecaller/databases/insights.db"	
Database	android.database.sqlite.SQLiteDatabase	compileStatement	["PRAGMA user_version;"]	"SQLiteProgram: PRAGMA user_version;"	
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/databases/truecaller.data.Notifications.s3db"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	openDatabase	["/data/user/0/com.truecaller/databases/truecaller.data.Notifications.s3db",""	"SQLiteDatabase: /data/user/0/com.truecaller/databases/truecaller.data.Notifications.s3db"	
Database	android.database.sqlite.SQLiteDatabase	compileStatement	["PRAGMA user_version;"]	"SQLiteProgram: PRAGMA user_version;"	
Database	android.database.sqlite.SQLiteDatabase	rawQueryWithFactory	[null,"SELECT * FROM preferences WHERE [time] > ?",["0"],"preferences",null]	"[object Object]"	
Database	android.database.sqlite.SQLiteDatabase	queryWithFactory	[null,false,"preferences",null,"[time] > ?",["0"],null,null,null,null,null]	"[object Object]"	
Database	android.database.sqlite.SQLiteDatabase	query	["preferences",null,"[time] > ?",["0"],null,null,null]	"[object Object]"	
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/databases/truecaller.data.Notifications.s3db"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	rawQueryWithFactory	["","SELE...	"[object Object]"	
Database	android.database.sqlite.SQLiteDatabase	rawQueryWithFactory	["","SELE...	"[object Object]"	
Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/no_backup/androidx.work.workdb"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	rawQueryWithFactory	["","SELE...	"[object Object]"	

Database	android.database.sqlite.SQLiteDatabase	getPath	[]	"/data/user/0/com.truecaller/no_backup/androidx.work.workdb"	"/data/user/0/com.truecalle
Database	android.database.sqlite.SQLiteDatabase	rawQueryWithFactory	["","SELE...	"[object Object]"	
Device Data	android.accounts.AccountManager	getAccountsByType	["com.truecaller.account"]		
Device Data	android.accounts.AccountManager	getAccountsByType	["com.truecaller.account"]		
Device Data	android.content.ContentResolver	query	["",null,null,null]	"[object Object]"	
Device Data	android.content.ContentResolver	query	["",null,null,null,null]	"[object Object]"	
Device Data	android.accounts.AccountManager	getAccountsByType	["com.truecaller.account"]		
Device Data	android.accounts.AccountManager	getAccountsByType	["com.truecaller.account"]		
Device Info	android.telephony.TelephonyManager	getNetworkOperatorName	[1]	"Android"	"Android"
Device Info	android.telephony.TelephonyManager	getNetworkOperatorName	[]	"Android"	"Android"
Dex Class Loader	dalvik.system.BaseDexClassLoader	findLibrary	["conscrypt_gmscore_jni"]	"/data/app/com.google.android.gms-pax2ngVZx0kNCufO4L0djQ==/lib/x86/libconscrypt_gmscore_jni.so"	"/data/app/com.google.and pax2ngVZx0kNCufO4L0djQ=
Dex Class Loader	dalvik.system.DexClassLoader	$init	["/data/data/com.truecaller/cache/frida9129098276355782929.dex","/data/data/com.truecaller/code_cach...		
Dex Class Loader	dalvik.system.BaseDexClassLoader	findResource	["com/google/android/gms/org/conscrypt/conscrypt.properties"]		
Dex Class Loader	dalvik.system.BaseDexClassLoader	findLibrary	["conscrypt_gmscore_jni"]	"/data/app/com.google.android.gms-pax2ngVZx0kNCufO4L0djQ==/lib/x86/libconscrypt_gmscore_jni.so"	"/data/app/com.google.and pax2ngVZx0kNCufO4L0djQ=

Dex Class Loader	dalvik.system.BaseDexClassLoader	findLibrary	["webviewchromium"]	"/system/app/webview/webview.apk!/lib/x86/libwebviewchromium.so"	"/system/app/webview/wet
Dex Class Loader	dalvik.system.BaseDexClassLoader	findLibrary	["webviewchromium_plat_support"]	"/system/lib/libwebviewchromium_plat_support.so"	"/system/lib/libwebviewchr
File IO	android.content.ContextWrapper	openFileInput	["frc_1:2237880283 2:android:d040f8b97ff358e8_firebase_activate.json"]	"java.io.FileInputStream@fd5c701"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@c6e39a6"	
File IO	android.content.ContextWrapper	openFileInput	["frc_1:2237880283 2:android:d040f8b97ff358e8_firebase_defaults.json"]	"java.io.FileInputStream@2cb9794"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@b83d373"	
File IO	libcore.io.IoBridge	open	["/data/misc/keychain/pubkey_blacklist.txt",0]	"java.io.FileDescriptor@de9ecde"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@1ef0dcb"	
File IO	libcore.io.IoBridge	open	["/data/misc/keychain/serial_blacklist.txt",0]	"java.io.FileDescriptor@a6ca6ee"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@a5fa76e"	
File IO	android.content.ContextWrapper	openFileOutput	["frc_1:2237880283 2:android:d040f8b97ff358e8_firebase_defaults.json",0]	"java.io.FileOutputStream@471a007"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@43f99e2"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@77ef9fa"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@1747787"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@3accc9"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@2aeeada"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@eeda423"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@b6c5f7f"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@1c7f0aa"	
File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/files/generatefid.lock",66]	"java.io.FileDescriptor@af1749b"	

File IO	libcore.io.IoBridge	open	["/data/user/0/com.truecaller/app_webview/webview_data.lock",66]	"java.io.FileDescriptor@50c6d76"	
IPC	android.content.ContextWrapper	startService	[""]	"ComponentInfo{com.truecaller/com.truecaller.analytics.EventsTrackerService}"	
IPC	android.content.ContextWrapper	startService	[""]	"ComponentInfo{com.truecaller/com.truecaller.analytics.EventsTrackerService}"	
IPC	android.content.ContextWrapper	registerReceiver	["",""		
IPC	android.content.ContextWrapper	registerReceiver	[null,""]	"Intent { act=android.net.wifi.STATE_CHANGE flg=0x4000010 (has extras) }"	
IPC	android.content.ContextWrapper	registerReceiver	[""		
IPC	android.content.ContextWrapper	registerReceiver	["",""	"Intent { act=android.intent.action.PROXY_CHANGE flg=0x24000010 (has extras) }"	
IPC	android.content.ContextWrapper	registerReceiver	[""	"Intent { act=android.net.conn.CONNECTIVITY_CHANGE flg=0x4000010 (has extras) }"	
IPC	android.content.ContextWrapper	registerReceiver	["",""		
IPC	android.content.ContextWrapper	registerReceiver	[""		
Network	com.android.okhttp.internal.huc.HttpURLConnectionImpl	getInputStream	[]	"buffer(com.android.okhttp.okio.GzipSource@46c36b4).inputStream()"	
Network	java.net.URL	openConnection	[]	"com.android.okhttp.internal.huc.HttpURLConnectionImpl:https://scontent.fcmb2-1.fna.fbcdn.net/v/t39....	

NAME	CLASS	METHOD	ARGUMENTS	RESULT	RETURN VALUE

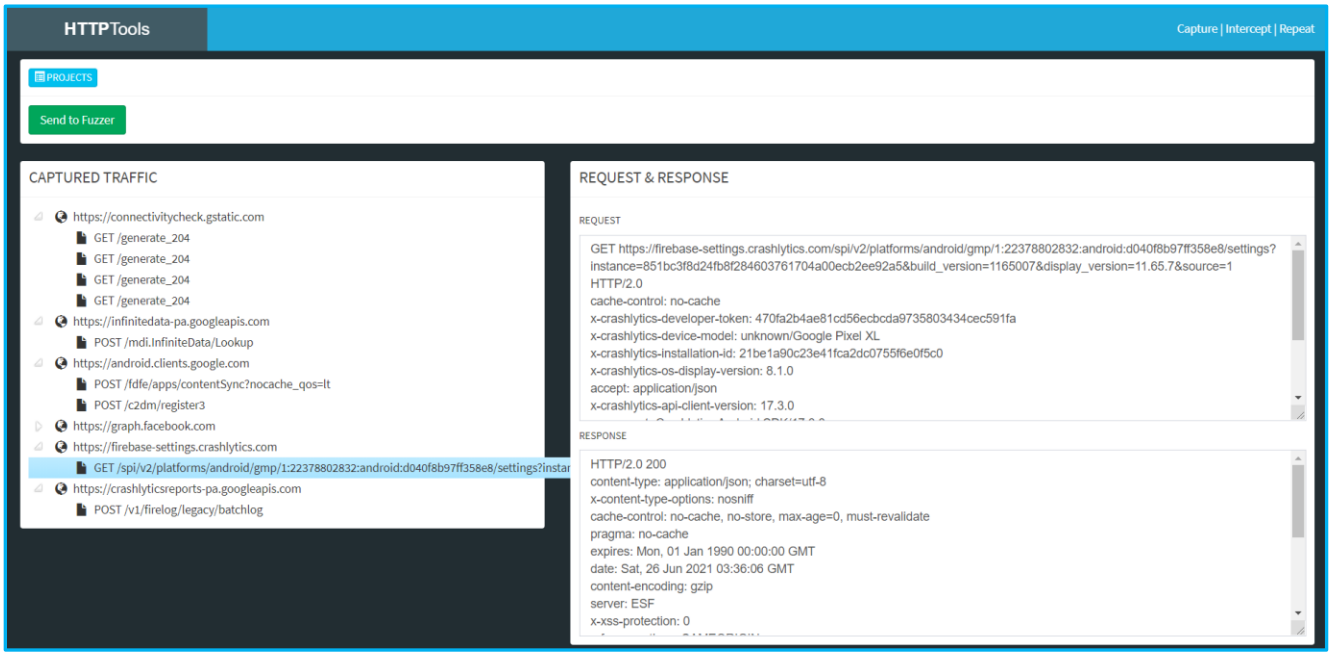
Showing 1 to 278 of 278 entries

Just like in the logcat stream logs, in here the APIs can also be monitored. APKs use various APIs in real-time to perform various functions, for example, the Base64 library.

So according to my case scenario, there are various types of APIs available.

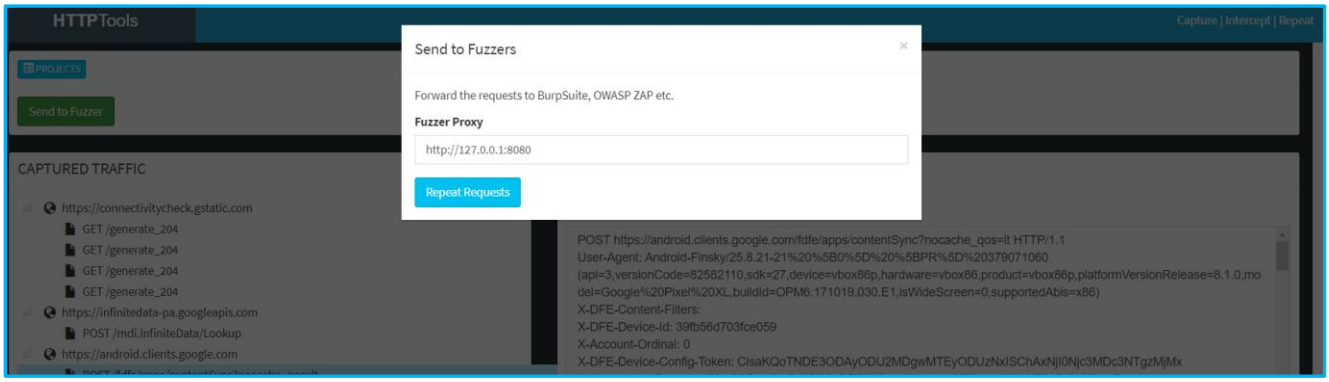
1) Binder
2) Crypto-Hash
3) Database
4) Device Data
5) Dex Class Loader
6) File IO
7) IPC
8) Network

HTTP Tools

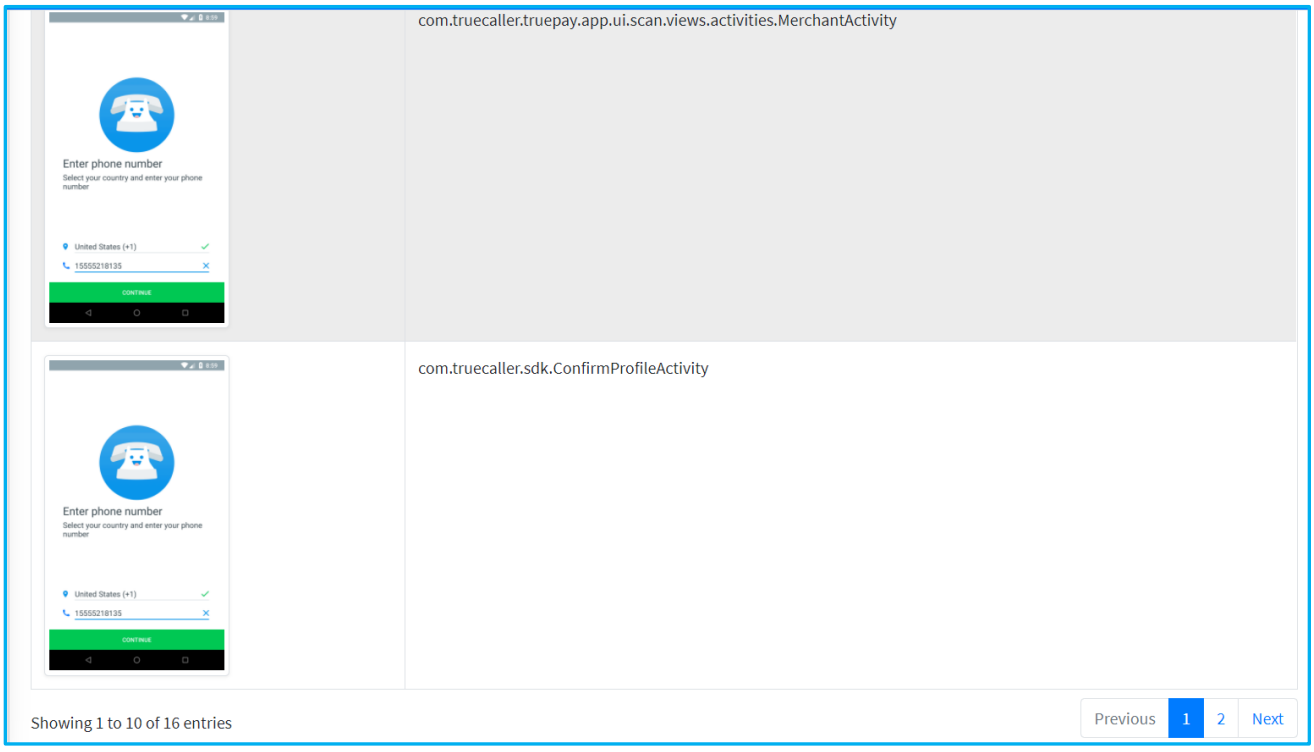


So, as you can see in the above image, there are various types of HTTP traffic has been captured.

Additionally, if you like you can send those traffic towards a fuzzer such as BurpSuite, OWASP ZAP etc.

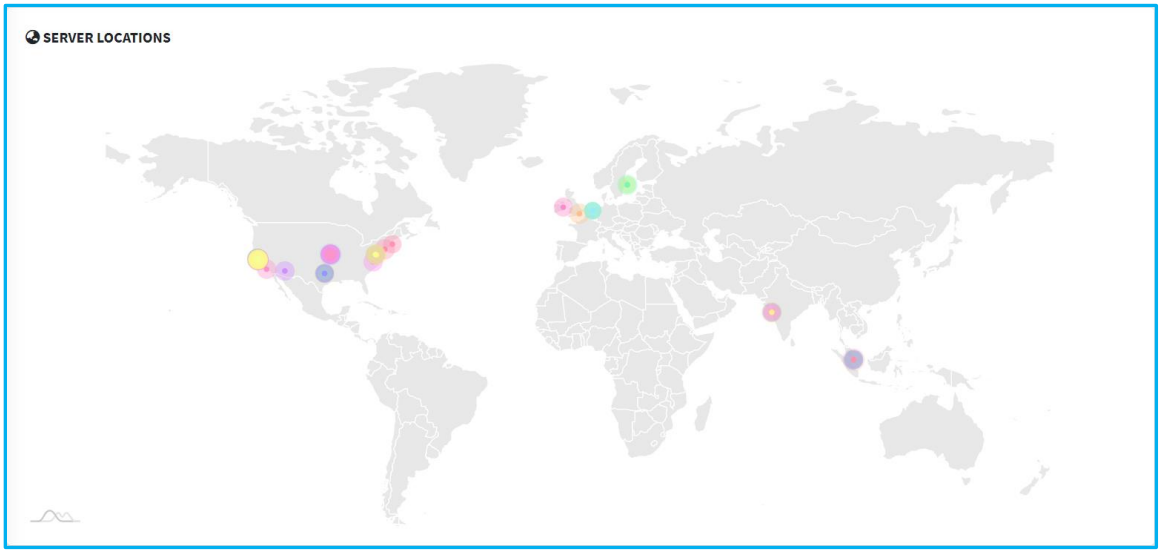


Exported Activity Tester



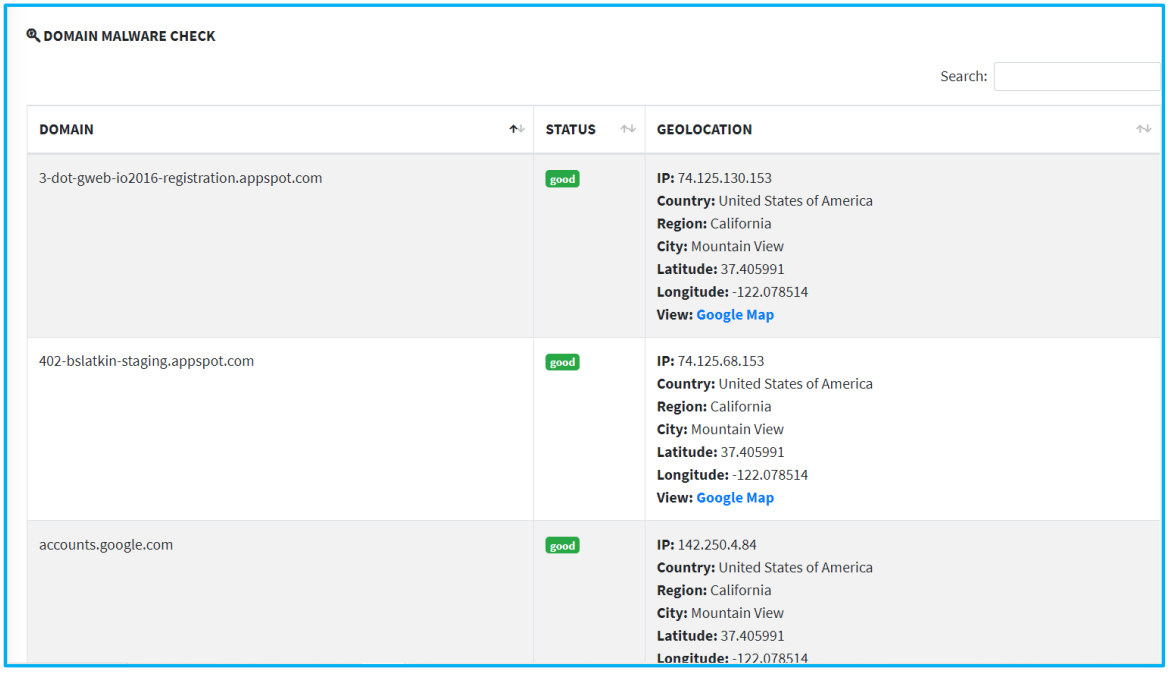
As you can see in the above image, this shows the various activities that are performed inside the app with the relevant screenshots.

Server Locations



It uses ip2location to give out its geolocation of the server. So, as you can see in the above image, TrueCaller App has got multiple servers situated in multiple countries.

Domain Malware Check



		Longitude: -77.487488 **View:** Google Map
ampcid.google.com	good	**IP:** 172.217.194.102 **Country:** United States of America **Region:** California **City:** Mountain View **Latitude:** 37.405991 **Longitude:** -122.078514 **View:** Google Map
apis.google.com	good	**IP:** 172.217.194.101 **Country:** United States of America **Region:** California **City:** Mountain View **Latitude:** 37.405991 **Longitude:** -122.078514 **View:** Google Map
apk.truecaller.com	good	**IP:** 34.117.64.87 **Country:** United States of America **Region:** Missouri **City:** Kansas City **Latitude:** 39.099731 **Longitude:** -94.578568 **View:** Google Map

Showing 1 to 10 of 148 entries

Previous 1 2 3 4 5 … 15 Next

In here, mobSF extracts all the URLs / IP Addresses which are hard-coded or being used in the application. Then it displays its malware status and it uses ip2location to find out the geo location.

Emails

EMAILS

screen-caller-id@2x.3479f58e
screen-caller-id@2x.68c92cb3
screen-messaging@2x.15db7010
badge-truecaller@2x.1be118ac
badge-app-store@2x.6d5ec0f2
press@truecaller.com
screen-spam-id@2x.0c511adb
badge-google-play@2x.65b16433
screen-dialer@2x.dba25bd2
screen-dialer@2x.13ed9edf
screen-spam-id@2x.327dbe88
screen-messaging@2x.ea042397
robert@broofa.com
support@truecaller.com
nsupport@truecaller.com

As you can see in the above image, there are several hardcoded email addresses in this app. Those emails are derived using the decompiled source code.

Trackers

TRACKERS

Search:

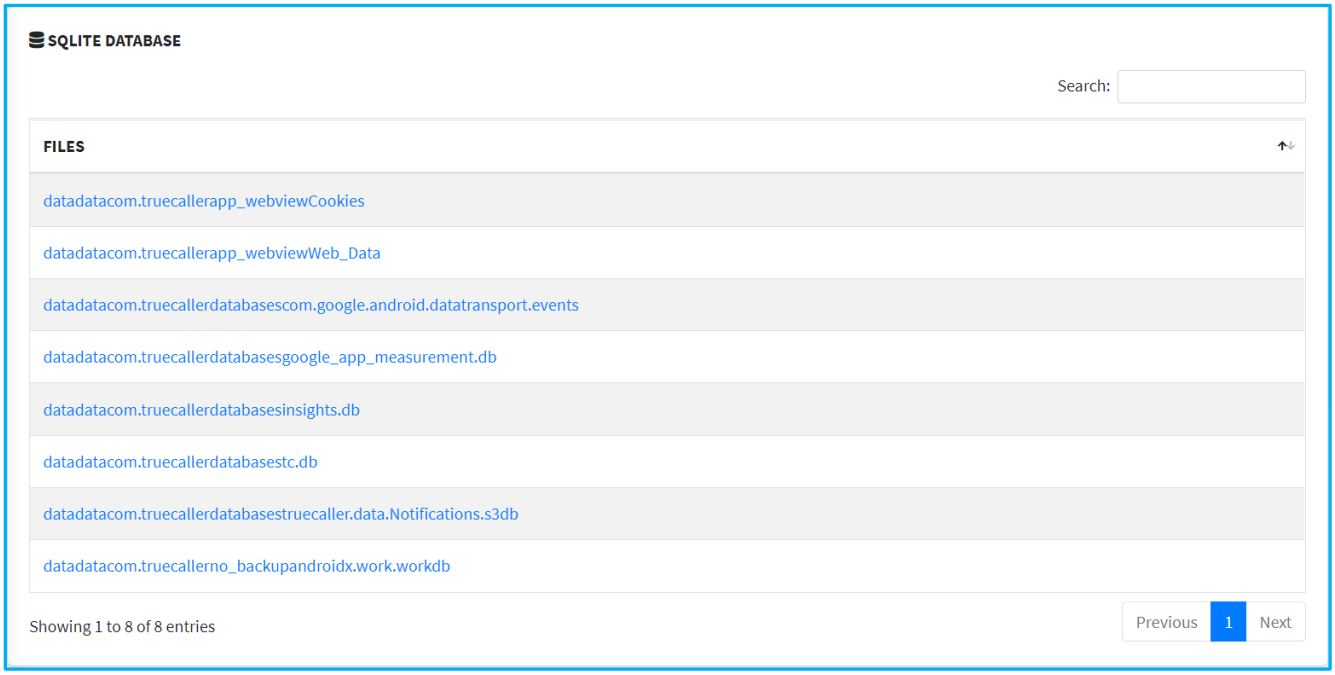
TRACKER NAME	CATEGORIES	URL
Facebook Ads	Advertisement	https://reports.exodus-privacy.eu.org/trackers/65
Facebook Analytics	Analytics	https://reports.exodus-privacy.eu.org/trackers/66
Facebook Audience	Analytics	https://reports.exodus-privacy.eu.org/trackers/47
Facebook Login	Identification	https://reports.exodus-privacy.eu.org/trackers/67
Facebook Notifications		https://reports.exodus-privacy.eu.org/trackers/68
Facebook Places		https://reports.exodus-privacy.eu.org/trackers/69
Facebook Share		https://reports.exodus-privacy.eu.org/trackers/70
Google AdMob	Advertisement	https://reports.exodus-privacy.eu.org/trackers/312
Google Ads		https://reports.exodus-privacy.eu.org/trackers/71
Google Analytics	Analytics	https://reports.exodus-privacy.eu.org/trackers/48

Showing 1 to 10 of 16 entries

Previous 1 2 Next

As you can see in the above image, trackers are used to perform several activities such as showing ads, authentication process, showing analytics etc.

SQLITE Database



Most of the applications will use internal SQLite databases to save information. In most of the cases, a person could easily find out sensitive information in here. Even though the database saves confidential information in an encrypted format, several passwords can be still found out inside the application.

References

https://book.hacktricks.xyz/mobile-apps-pentesting/android-app-pentesting

https://resources.infosecinstitute.com/topic/android-penetration-tools-walkthrough-series-mobsf/

https://www.hackingarticles.in/android-pentest-automated-analysis-using-mobsf/

Part B

What are Decentralized Applications ?

- **Decentralized applications** or else ***dApps*** are digital applications or programs which exist & run on a **blockchain** or **P2P network (**peer-to-peer**)** of computers instead of a single computer. Most of the time, those are under the purview of a blockchain.

Clear explanation on the concept

- When we consider a particular standard web app (Facebook, PickMe), most of the time those applications run on a computer system which is owned & operated by an organization. Those organizations provide the full authority over the app and its behavior. Although there are multiple users on 1 side, the backend is generally controlled by a single organization.

- However decentralized applications don‘t necessarily needs to be run on top of a blockchain network. For examples, Tor web browser, BitTorrent & BitMessage can be considered as decentralized applications that run on a P2P network, but not on a blockchain.

- Decentralized applications are a piece of tech that communicate with the blockchain while managing the state of all network actors. When the interface of an decentralized application is concerned, it does not look any different than any website or mobile app today.

- The smart contract is considered to be the **core logic** behind the decentralized applications. Smart contracts are the integral building blocks of blockchains which process information from external sensors or events & help the blockchain manage the state of all network actors.

- Just like in a normal web-based application, decentralized applications also consist with a frontend & a backend. In here the frontend of a decentralized application represents what you see and the backend represents the entire business logic.

- The frontend is used to host files like a photo, a video or an audio on decentralized storage networks such as Swarm or IPFS. This contains a **"wallet"** that communicates with the blockchain. This wallet manages cryptographic keys and the blockchain address. Additionally, Public-key infrastructure **(PKI)** is used for user identification and user authentication. In here, instead of an API connecting to a database, a wallet triggers activities of a smart contract, which interacts with a blockchain.

- In the backend, the Web3 adds a whole new infrastructure layer for decentralized applications to interact with the decentralized protocol stack. Decentralized apps need to have a component that manages a user's private keys with which one can sign transactions on the state layer, the blockchain.

Building a suitable case scenario

“Hadahan” is an open source, decentralized prediction market platform built on the Ethereum blockchain. Hadahan allows users to match their own horoscopes online in order to find a matching life partner.

A person can gain lot of advantages when using this particular service.

1) This allows people to upload their own horoscope to a P2P decentralized network.

2) After the creation of a particular record, practically it cannot be changed or tampered with during the execution.

3) A specific single party does not own the power to influence Hadahan service’s behavior.

4) Since this Hadahan platform is open source, the transparency of the procedure is very high.

5) There is no **single point of failure** at all because the machines of individual users don’t rely on a single central server to handle processes.

6) When the security is concerned, all the Hadahan users’ information is stored on a shared database which no authority has control over. Additionally, the very sensitive horoscope details can only be decrypted by the user itself.

7) In this Hadahan platform, there is no specific central authority that executes censorship. In here, the users cannot be blocked from submitting their horoscopes. So, this makes the users more responsible specially when sharing content to this platform.

When it comes to the development process of this Hadahan app, it's not just a simple plug-n-play project. Developing such kind of a DApp requires an immense amount of programming knowledge.

So, now let's move on for developing our Hadahan DApp.

Step 1 :- Learn Ethereum

Our Hadahan app is required to follow a specific conceptual framework. But in order to achieve that requirement, it's necessary to learn Ethereum.

Ex:- *Our Hadahan app needs to be open-source and the operations should be performed autonomously without any entity controlling majority of the crypto tokens. Additionally, the backend code must comprise smart contracts and must run on a decentralized blockchain. In this occasion it's very much important to use a crypto token generated using a standard cryptographic algorithm when storing in blockchain.*

Step 2 :- Get our Blockchain

"Testrpc" is a very easy-to-use command line interface, that a developer can choose. This tool can be used to specify the block intervals.

Step 3 :- Communicate with Blockchain

In order to communicate with the blockchain, **"web3.js"** tool can be used. In here we need to configure the necessary parameters such as acongig.js file & web3 API functions.

Step 4 :- Learn Solidity

Solidity is considered as the proprietary language of Ethereum to write smart contracts. This feature-rich language is specifically designed for this particular purpose. Although our Hadahan App's frontend code can be written in any language, the backend code must comprise smart contracts.

Step 5 :- Code Smart Contracts

In order to make our Hadahan app more effective, the following guidelines needs to be followed.

- Computing logic and storage requirements should be kept to minimal. Otherwise, it's going to consume lot of computing power.

- The code has to be simpler as possible. Otherwise, it's more likely to occur some severe errors. Since the outcome of a smart contract is irreversible, maintain the simplicity of code is very much vital.

Step 6 :- Deploy Smart Contracts

If we use the **"Truffle"** tool for this process, we can gain several advantages from this.

- A single directory allows the developer to maintain all her smart contracts.

- As the scripts can deploy the contracts in the test environment, this tool can easily blend into the testing framework.

Step 7 :- Invoke Smart Contracts

In here, it's necessary to have the smart contracts in hexadecimal strings. However, there are Ethereum contract "application binary interface" *(ABI)* libraries which can be used as helping hands.

Step 8 :- Setting-Up an Ethereum Account

In order to execute the smart contracts, we need to create Ethereum account for that. Then we need to update the config.js file with the key details. After that we can go back to the truffle test and see Ether moving between accounts. However, it's very much important not to share the private key with anyone.

Step 9 :- Transact with the Smart Contract

As the last step in our Hadahan App development, we need to transact using the Ether. In order to carry out this process, there are 3 main options.

1) Transfer to another address as a value

2) Call a contract function that will update the state of the network and spend the Ether to pay the fees to the miner.

3) Involve a contract that updates the state of the network and accepts Ether as payment. In here the developer also needs to pay the fees to the miner.

References

https://101blockchains.com/dapp-development-guide/

https://research.aimultiple.com/dapps/

https://research.aimultiple.com/dapps/

https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp