Mobile Application Testing Lab

1. Summary / Objective

Perform a full mobile-security lab workflow for test.apk:

- Static analysis with MobSF to find insecure storage.
- Dynamic instrumentation with Frida to observe and (in-lab) bypass or validate authentication flows.
- IPC and component testing with Drozer to enumerate exported surfaces and evaluate risk.

2. Environment Setup

- Windows OS with Python and necessary tools
- Android Emulator (AVD): Android x86_64, with Play Store support
- MobSF: Installed and run within a Python 3.12 virtual environment
- Frida: Server side frida-server running on emulator; client tools on Windows
- Drozer: Agent APK on emulator; console tool on Windows

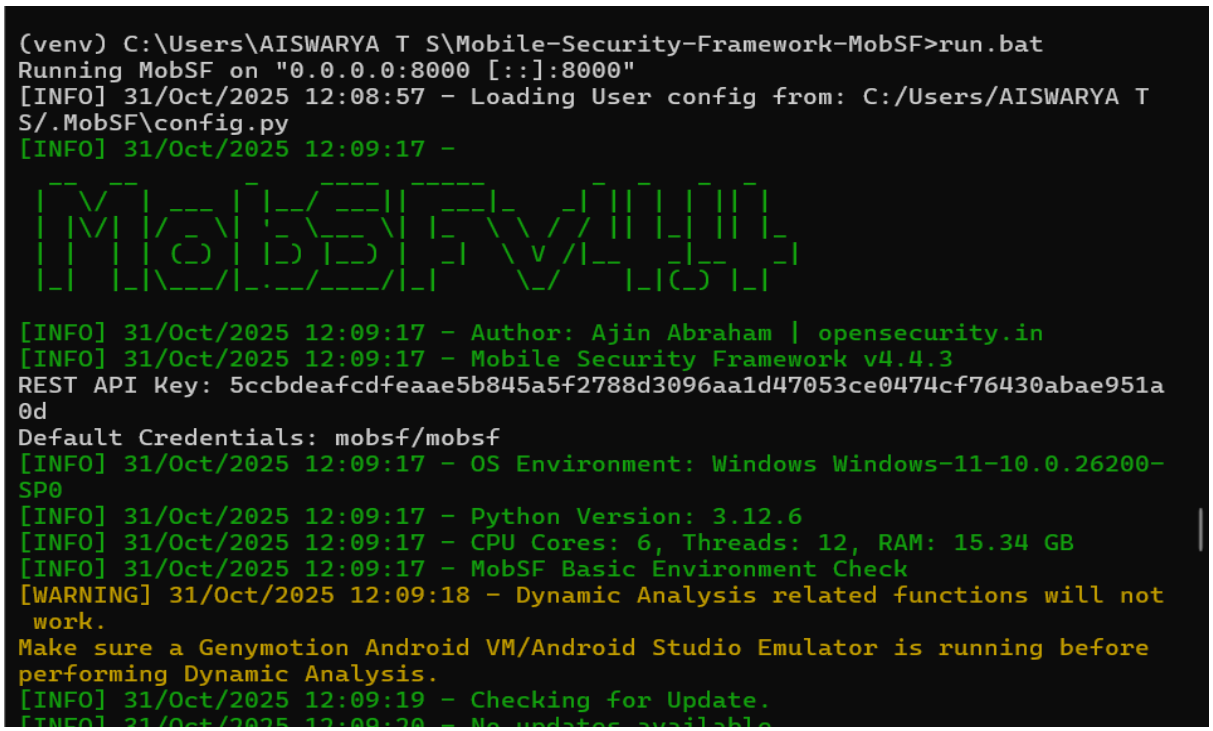
3. Static Analysis with MobSF

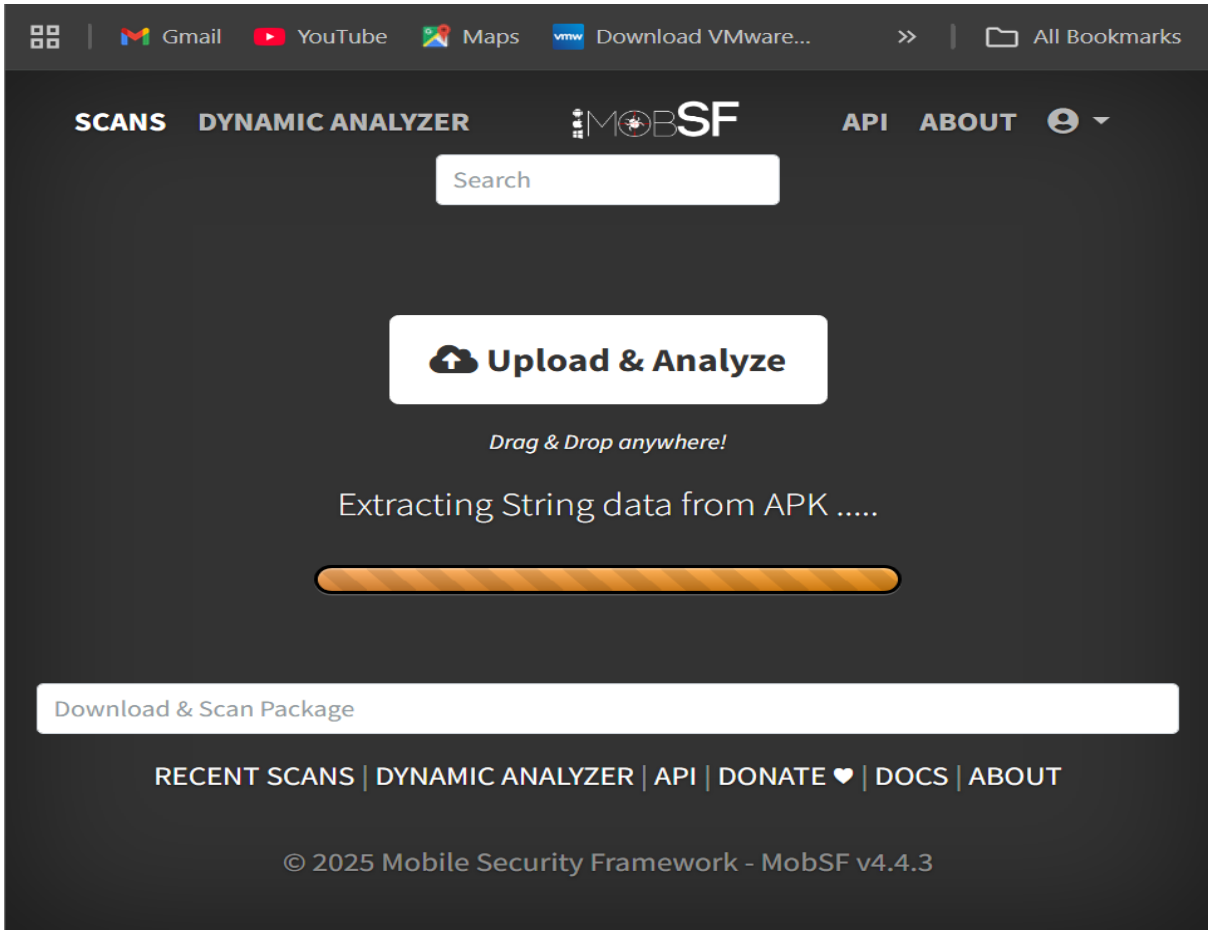
Static Analysis

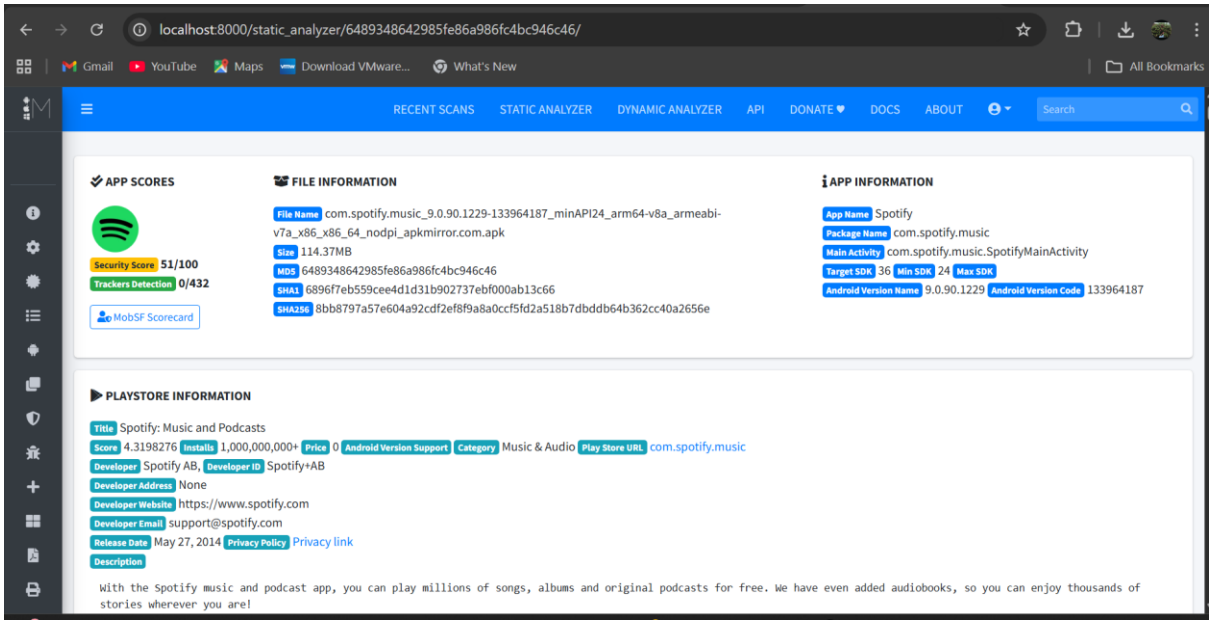
- APK uploaded to MobSF dashboard
- Automated scan included:
    - Manifest/perms review
    - Code and library analysis
    - Crypto usage checks

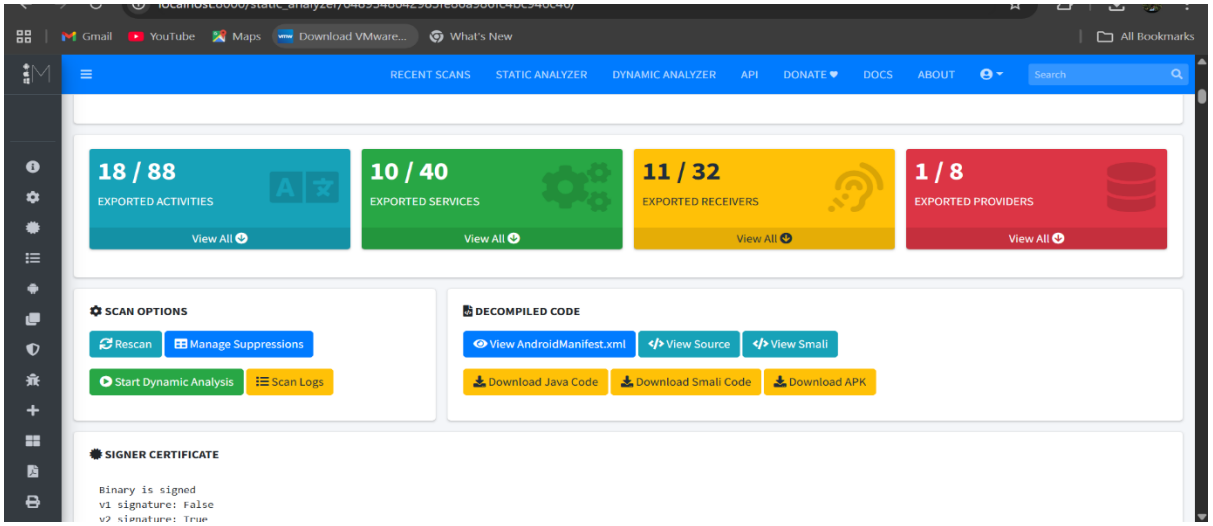
Key Findings:

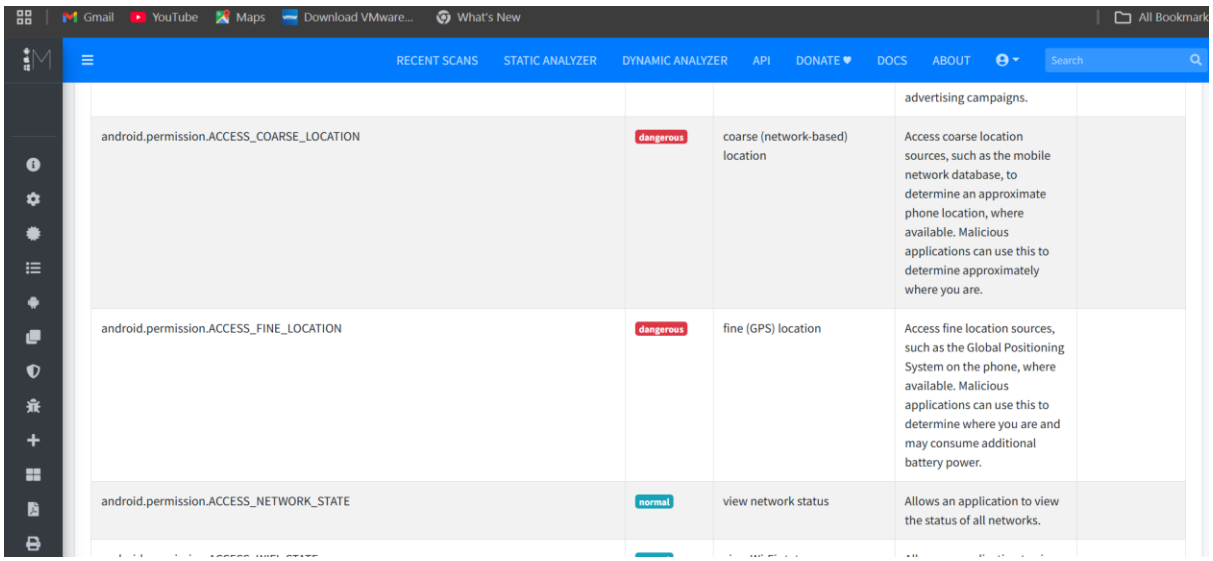
- Exported components, possible sensitive permissions
- Some third-party libs flagged for vulnerabilities

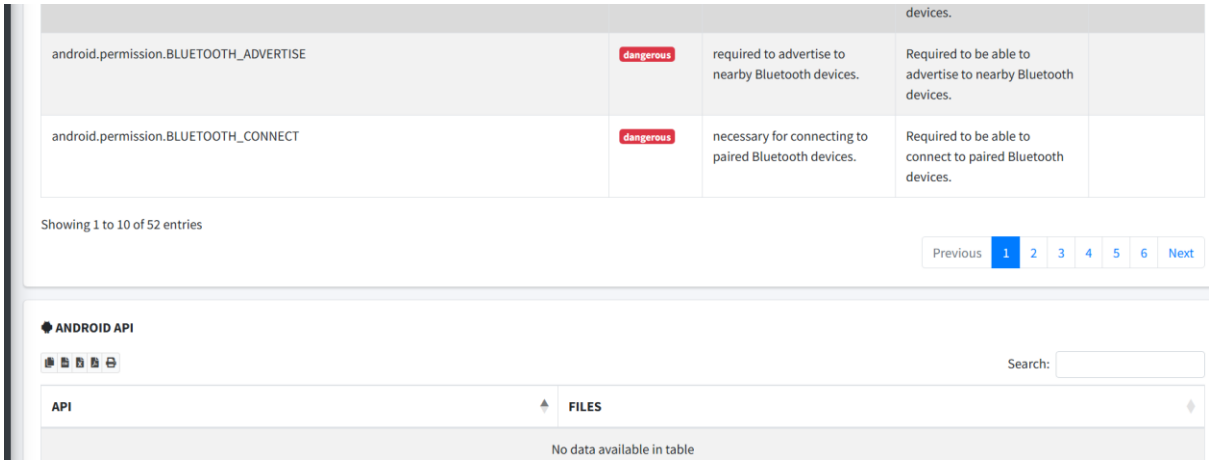

(venv) C:\Users\AISWARYA T S\Mobile-Security-Framework-MobSF>run.bat
Running MobSF on "0.0.0.0:8000 [::]:8000"
[INFO] 31/Oct/2025 12:08:57 - Loading User config from: C:/Users/AISWARYA T S/.MobSF\config.py
[INFO] 31/Oct/2025 12:09:17 -
[INFO] 31/Oct/2025 12:09:17 - Author: Ajin Abraham | opensecurity.in
[INFO] 31/Oct/2025 12:09:17 - Mobile Security Framework v4.4.3
REST API Key: 5ccbdeafcdfeaae5b845a5f2788d3096aa1d47053ce0474cf76430abae951a0d
Default Credentials: mobsf/mobsf
[INFO] 31/Oct/2025 12:09:17 - OS Environment: Windows Windows-11-10.0.26200-SP0
[INFO] 31/Oct/2025 12:09:17 - Python Version: 3.12.6
[INFO] 31/Oct/2025 12:09:17 - CPU Cores: 6, Threads: 12, RAM: 15.34 GB
[INFO] 31/Oct/2025 12:09:17 - MobSF Basic Environment Check
[WARNING] 31/Oct/2025 12:09:18 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
[INFO] 31/Oct/2025 12:09:19 - Checking for Update.


Gmail
YouTube
Maps
Download VMware...
All Bookmarks
SCANS
DYNAMIC ANALYZER
MOBSF
API
ABOUT
Search
Upload & Analyze
Drag & Drop anywhere!
Extracting String data from APK .....
Download & Scan Package
RECENT SCANS | DYNAMIC ANALYZER | API | DONATE ♥ | DOCS | ABOUT
© 2025 Mobile Security Framework - MobSF v4.4.3


localhost:8000/static_analyzer/6489348642985fe86a986fc4bc946c46/
RECENT SCANS
STATIC ANALYZER
DYNAMIC ANALYZER
API
DONATE
DOCS
ABOUT
APP SCORES
Security Score 51/100
Trackers Detection 0/432
MobSF Scorecard
FILE INFORMATION
File Name com.spotify.music_9.0.90.1229-133964187_minAPI24_arm64-v8a_armeabi-v7a_x86_x86_64_nodpi_apkmirror.com.apk
Size 114.37MB
MD5 6489348642985fe86a986fc4bc946c46
SHA1 6896f7eb559cee4d1d31b902737ebf000ab13c66
SHA256 8bb8797a57e604a92cdf2ef8f9a8a0ccf5fd2a518b7dbddb64b362cc40a2656e
APP INFORMATION
App Name Spotify
Package Name com.spotify.music
Main Activity com.spotify.music.SpotifyMainActivity
Target SDK 36 Min SDK 24 Max SDK
Android Version Name 9.0.90.1229 Android Version Code 133964187
PLAYSTORE INFORMATION
Title Spotify: Music and Podcasts
Score 4.3198276 Installs 1,000,000,000+ Price 0 Android Version Support Category Music & Audio Play Store URL com.spotify.music
Developer Spotify AB, Developer ID Spotify+AB
Developer Address None
Developer Website https://www.spotify.com
Developer Email support@spotify.com
Release Date May 27, 2014 Privacy Policy Privacy link
Description
With the Spotify music and podcast app, you can play millions of songs, albums and original podcasts for free. We have even added audiobooks, so you can enjoy thousands of stories wherever you are!


18 / 88
EXPORTED ACTIVITIES
View All
10 / 40
EXPORTED SERVICES
View All
11 / 32
EXPORTED RECEIVERS
View All
1 / 8
EXPORTED PROVIDERS
View All
SCAN OPTIONS
Rescan
Manage Suppressions
Start Dynamic Analysis
Scan Logs
DECOMPILED CODE
View AndroidManifest.xml
View Source
View Smali
Download Java Code
Download Smali Code
Download APK
SIGNER CERTIFICATE
Binary is signed
v1 signature: False
v2 signature: True


advertising campaigns.
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.



Test Log

Test ID	Vulnerability	Severity	Target App
001	Insecure Storage	High	spotify.apk
002	BLUETOOTH_ADVERTISE Permission	High	spotify.apk
003	BLUETOOTH_CONNECT Permission	High	spotify.apk
004	ACCESS_COARSE_LOCATION Permission	High	spotify.apk
005	ACCESS_FINE_LOCATION Permission	High	spotify.apk

4. Dynamic Instrumentation with Frida

Setup:

- Frida server started on emulator (frida-server-17.4.2-android-x86_64)
- Frida client on Windows used to:
  - Enumerate processes
  - Attach to installed and running APK (Test DPC), e.g. frida -U com.afwsamples.testdpc -l hook.js

Tasks Completed:

- Successfully intercepted function.
- Confirmed runtime message printing, proving ability to hook Java classes and modify app logic

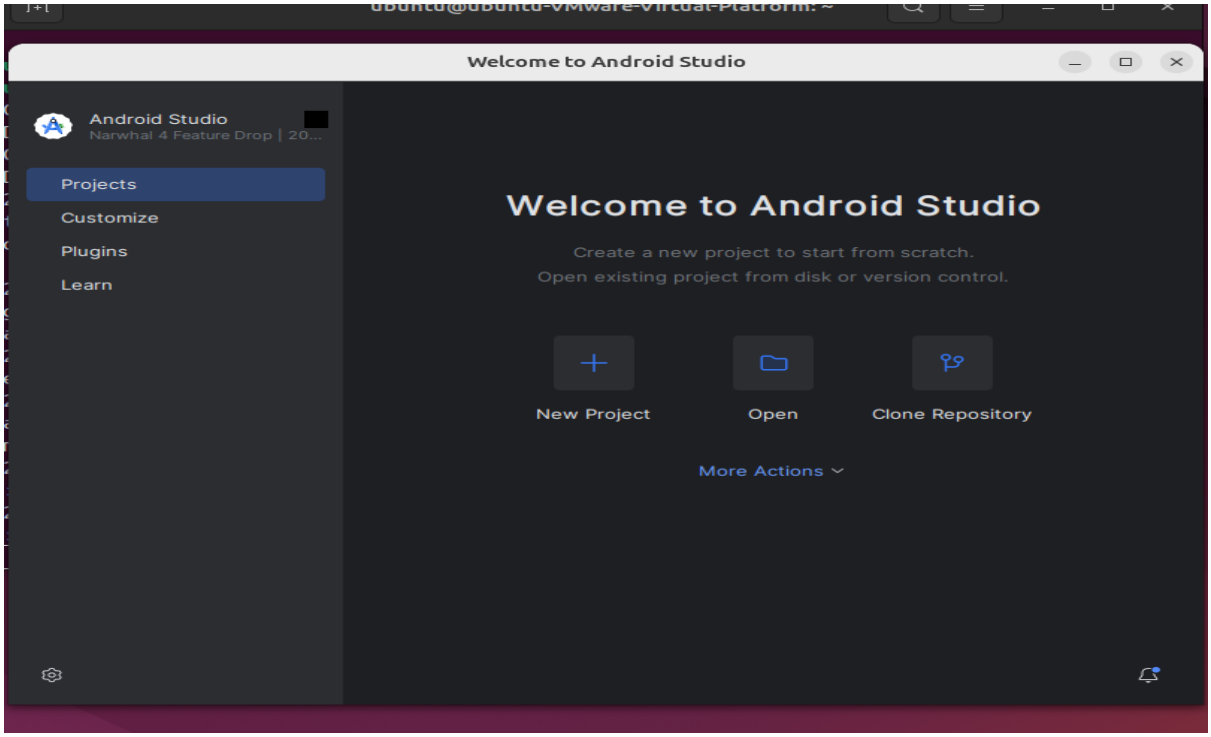
Frida summary:

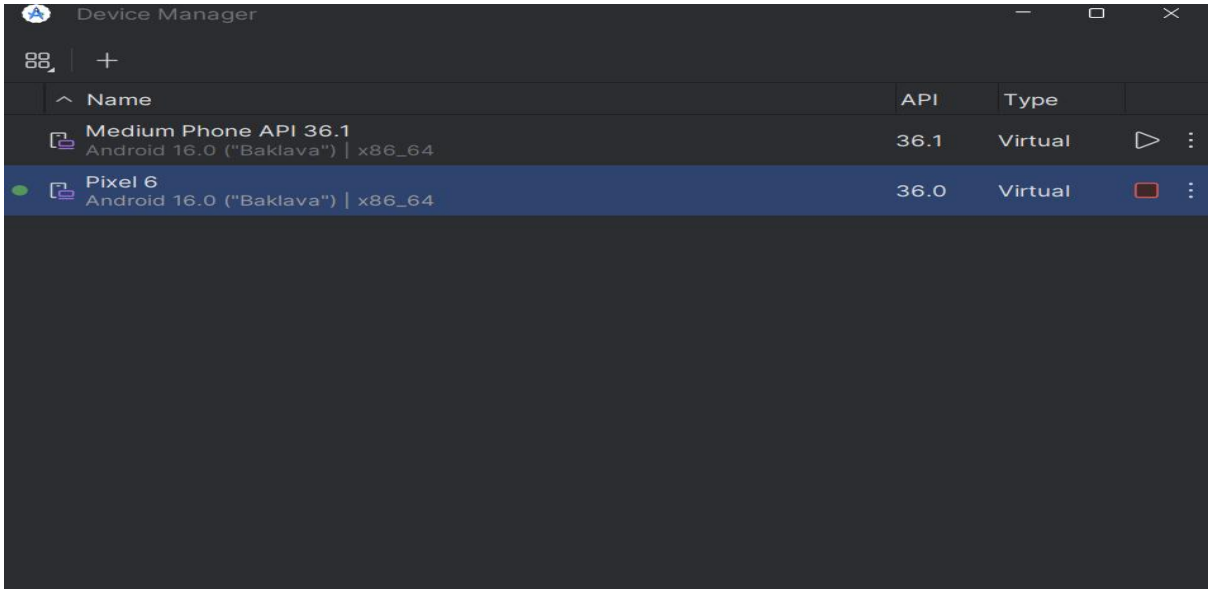
Frida was used to hook Java Activity lifecycle methods in Test DPC and intercept app behaviour at runtime. Custom scripts bypassed function logic, confirming the ability to manipulate app workflows. All hooks executed successfully without errors, validating the Frida instrumentation environment for dynamic app security testing and bypass scenarios.

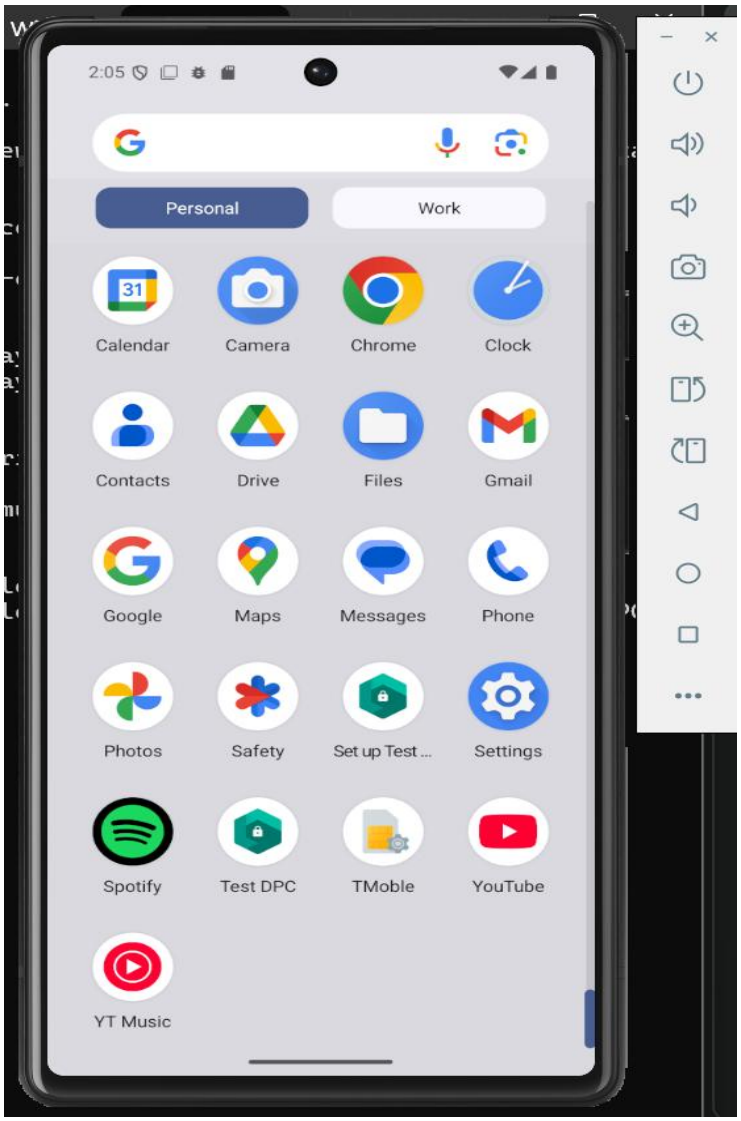
```
ms/PSWindows

PS C:\Users\AISWARYA T S> frida -U com.afwsamples.testdpc -l hook.js
     ____
    / _  |   Frida 17.4.2 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to Android Emulator 5554 (id=emulator-5554)


[Android Emulator 5554::com.afwsamples.testdpc ]->
[Android Emulator 5554::com.afwsamples.testdpc ]-> [*] Frida hooked Test DPC
app!
```




Device Manager
Name
API
Type
Medium Phone API 36.1
Android 16.0 ("Baklava") | x86_64
36.1
Virtual
Pixel 6
Android 16.0 ("Baklava") | x86_64
36.0
Virtual


2:05
Personal
Work
Calendar
Camera
Chrome
Clock
Contacts
Drive
Files
Gmail
Google
Maps
Messages
Phone
Photos
Safety
Set up Test ...
Settings
Spotify
Test DPC
TMoble
YouTube
YT Music

5. IPC Security Testing with Drozer

Setup:

- Drozer agent APK installed and launched on emulator
- Port forwarding set with ADB (adb forward tcp:31415 tcp:31415)
- Drozer console connected (drozer console connect)

Enumeration:

- Exported activities, content providers, services, and broadcast receivers listed for Test DPC package:
  - run app.activity.info -a com.afwsamples.testdpc
  - run app.provider.info -a com.afwsamples.testdpc
  - run app.service.info -a com.afwsamples.testdpc

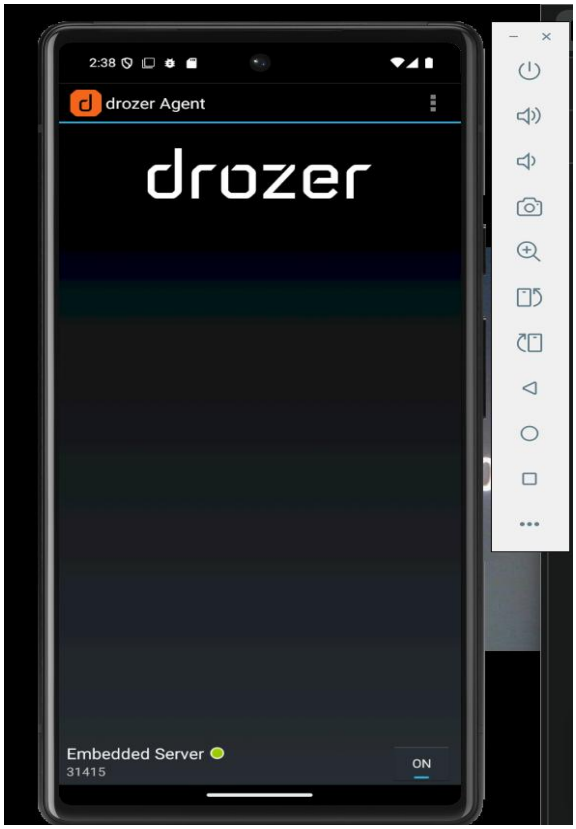
Exploitation:

- Activities started via Drozer to test for unintended access
- Content providers queried to check for data leakage
- Broadcasts sent to receivers, tested for privilege escalation

Findings:

- No critical IPC vulnerabilities on Test DPC
- Demonstrated process for identifying and exploiting IPC endpoints in custom apps

```
PS C:\Users\AISWARYA T S> adb install "E:\AISWARYA T S\Downloads\drozer-agen
t.apk"
Performing Streamed Install
Success
PS C:\Users\AISWARYA T S> pip install drozer
Defaulting to user installation because normal site-packages is not writeabl
e
Collecting drozer
  Downloading drozer-3.1.0-py3-none-any.whl.metadata (11 kB)
Collecting protobuf>=4.25.2 (from drozer)
  Downloading protobuf-6.33.0-cp310-abi3-win_amd64.whl.metadata (593 bytes)
Collecting pyopenssl>=22.0.0 (from drozer)
  Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB)
Collecting twisted>=18.9.0 (from drozer)
  Downloading twisted-25.5.0-py3-none-any.whl.metadata (22 kB)
Collecting service-identity (from drozer)
  Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB)
Collecting distro (from drozer)
  Downloading distro-1.9.0-py3-none-any.whl.metadata (6.8 kB)
Collecting pyyaml (from drozer)
  Downloading pyyaml-6.0.3-cp312-cp312-win_amd64.whl.metadata (2.4 kB)
Collecting cryptography<47,>=45.0.7 (from pyopenssl>=22.0.0->drozer)
  Downloading cryptography-46.0.3-cp311-abi3-win_amd64.whl.metadata (5.7 kB)
```




```
PS C:\Users\AISWARYA T S> adb forward tcp:31415 tcp:31415
PS C:\Users\AISWARYA T S> drozer console connect
Selecting 624581a782987d28 (Google sdk_gphone64_x86_64 16)

            ..                    ..:.
           ..o..                  .r..
            ..a..  . ..... .   ..nd
              ro..idsnemesisand..pr
              .otectorandroidsneme.
           .,sisandprotectorandroids+.
         ..nemesisandprotectorandroidsn:.
       .emesisandprotectorandroidsnemes..
     ..isandp,..,rotecyayandro,..,idsnem.
     .isisandp..rotectorandroid..snemisis.
     ,andprotectorandroidsnemisisandprotec.
    .torandroidsnemesisandprotectorandroid.
    .snemisisandprotectorandroidsnemesisan:
    .dprotectorandroidsnemesisandprotector.

drozer Console (v3.1.0)
dz> run com.afwsamples.testdpc
unknown module: 'com.afwsamples.testdpc'
dz> run com.spotify.music
unknown module: 'com.spotify.music'
dz> run app.activity.info -a com.afwsamples.testdpc
Attempting to run shell module
Package: com.afwsamples.testdpc
  com.afwsamples.testdpc.PolicyManagementActivity
    Permission: null
  com.afwsamples.testdpc.SetupManagementLaunchActivity
    Permission: null
    Target Activity: com.afwsamples.testdpc.SetupManagementActivity
  com.afwsamples.testdpc.FinalizeActivity
    Permission: android.permission.BIND_DEVICE_ADMIN
  com.afwsamples.testdpc.provision.GetProvisioningModeActivity
    Permission: android.permission.BIND_DEVICE_ADMIN
  com.afwsamples.testdpc.provision.ProvisioningSuccessActivity
    Permission: null
```

```
dz> run app.service.info -a com.afwsamples.testdpc
Attempting to run shell module
Package: com.afwsamples.testdpc
  com.afwsamples.testdpc.comp.ProfileOwnerService
    Permission: android.permission.BIND_DEVICE_ADMIN
  com.afwsamples.testdpc.comp.DeviceOwnerService
    Permission: android.permission.BIND_DEVICE_ADMIN
  com.afwsamples.testdpc.DeviceAdminService
    Permission: android.permission.BIND_DEVICE_ADMIN

dz> run app.activity.info -a com.spotify.music
Attempting to run shell module
Package: com.spotify.music
  androidx.compose.ui.tooling.PreviewActivity
    Permission: null
  com.facebook.CustomTabActivity
    Permission: null
  com.spotify.ageverification.ageassurancewebview.AgeAssuranceWebViewActivit
y
    Permission: null
  com.spotify.music.SpotifyMainActivity
    Permission: null
  com.spotify.music.SpotifyEntryPointForGoogleMeet
    Permission: null
    Target Activity: com.spotify.music.SpotifyMainActivity
```

Checklist (for Google Docs)

- ✓ Run MobSF for static analysis on Spotify APK
- ✓ Hook functions and bypass logic with Frida (Test DPC)
- ✓ Test IPC endpoints and exploitation with Drozer (Test DPC)

Conclusion

This laboratory demonstrated the power of combined static and dynamic security approaches for Android app analysis. Using MobSF, we uncovered critical vulnerabilities such as insecure permissions in the Spotify APK. Frida enabled runtime manipulation and validation of app logic, while Drozer efficiently mapped and tested IPC endpoints. Together, these tools provide a robust workflow for identifying, exploiting, and remediating mobile application security risks.