

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):

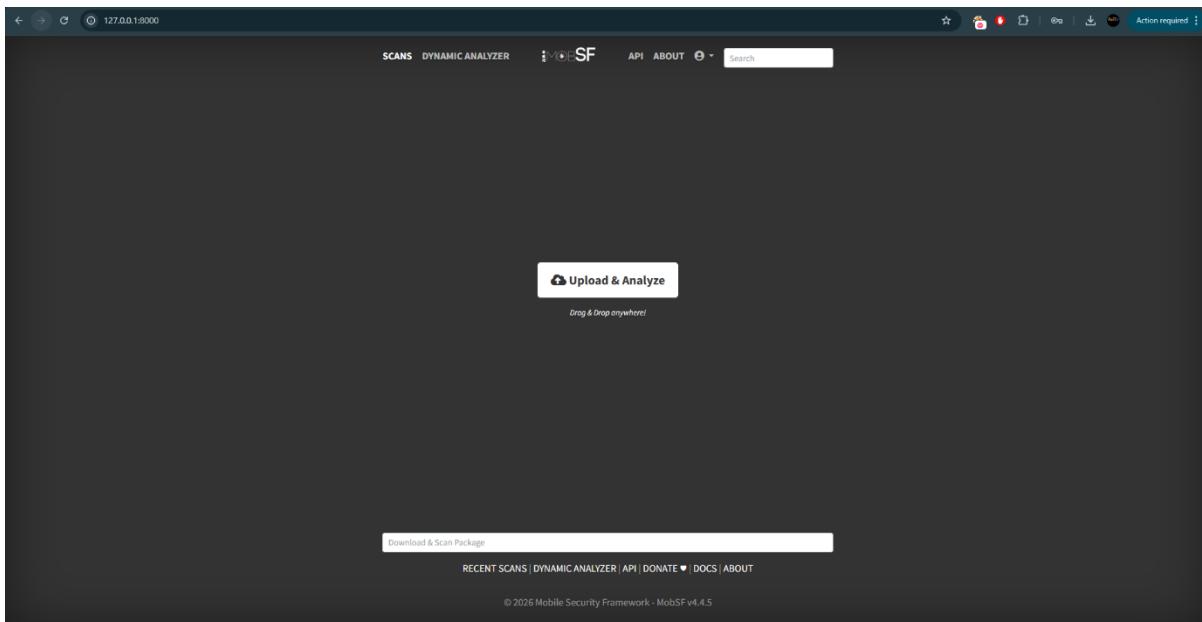
run.bat 127.0.0.1:8000



```
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 - [██████████] 100% [██████████]
[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: 220615dc2b673d95e72271d6edebac42d5dc2b071df4fe4b0996ffff141337ea3
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



The screenshot shows the MobSF static analyzer interface. The top navigation bar includes links for RECENT SCANS, STATIC ANALYZER, DYNAMIC ANALYZERS, API, DONATE, DOCS, ABOUT, and SUPPORT. The main content area is divided into several sections:

- APP SCORES**: Shows a score of 48/100 with a green progress bar.
- FILE INFORMATION**: Displays the APK file name as androidGoat.apk and its SHA1 hash as 1a30254ba65fb4ad4c6982f69fae72666.
- APP INFORMATION**: Provides details like App Name (AndroidGoat), Package Name (com.wings.cat.agot), and Target SDK (33).
- EXPORTED ACTIVITIES**: Score 1/30, with a teal button to "View All".
- EXPORTED SERVICES**: Score 1/1, with a green button to "View All".
- EXPORTED RECEIVERS**: Score 2/2, with a yellow button to "View All".
- EXPORTED PROVIDERS**: Score 1/2, with a red button to "View All".
- SCAN OPTIONS**: Includes buttons for Recan, Manage Suppressions, Start Dynamic Analysis, and Scan Logs.
- DECOMPILATION**: Shows Java code snippets and download links for Download Java Code, Download Small Code, and Download APK.
- SIGNER CERTIFICATE**: Displays certificate details including Subject, Valid To, Issuer, Serial Number, Hash, and MD5.

4. Review the Security Analysis tab

127.0.0.1:8000/static_analyzer/8a38254ba65fa4d4cb982f69ac72666/*code_analysis

RECENT SCANS **STATIC ANALYZER** **DYNAMIC ANALYZER** **API** **DONATE** **DOCS** **ABOUT** **Team** **Action required**

CODE ANALYSIS

HIGH	WARNING	INFO	SECURE	SUPPRESSED
0	7	2	2	0

Search:

NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	App creates temp file. Sensitive information should never be written into a temp file.	Warning	CWE: CWE-278: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	overapp1/app/main/ResourceStorageActivity.java overapp1/app/main/ResourceStorageTempActivity.java	
2	The App logs information. Sensitive information should never be logged.	Info	CWE: CWE-352: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3	overapp1/app/main/LogActivity.java	
3	This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.	Secure	OWASP MASVS: MSTG-NETWORK-4	overapp1/app/main/HttpsActivity.java	
4	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	CWE: CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: A7: Cryptographic Failures	overapp1/app/main/ContentDetailsActivity.java overapp1/app/main/ContentDetailsRawActivity.java overapp1/app/main/ContentDetailsRawActivity.java	
5	Filters may contain hardcoded sensitive information like usernames, passwords, keys etc.	Warning	CWE: CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M8: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	overapp1/app/main/FilterActivity.java	
6	MDS is a weak hash known to have hash collisions.	Warning	CWE: CWE-327: Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-4	overapp1/app/main/AccessControlActivity.java	
7	Ensure that user controlled URLs never reaches the WebView. Enabling file access from URLs in WebView can leak sensitive information from the file system.	Warning	CWE: CWE-200: Information Exposure OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: MSTG-PLATFORM-7	overapp1/app/main/InputTranslationWebViewActivity.java	
8	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	overapp1/app/main/CopyDataActivity.java	
9	App can read/write to External Storage. Any App can read data written to External Storage.	Warning	CWE: CWE-278: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	overapp1/app/main/ResourceStorageActivity.java	
10	This App may request root (Super User) privileges.	Warning	CWE: CWE-250: Execution with Unnecessary Privileges OWASP MASVS: MSTG-RESILIENCE-1	overapp1/app/main/RootDetectorActivity.java	

<https://github.com/MobSF/MobSF> | [Report a Bug](#) | [Feature Requests](#) | [Code Review](#) | [Feedback](#) | [Network Communication Reflections](#)



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.