



## 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**



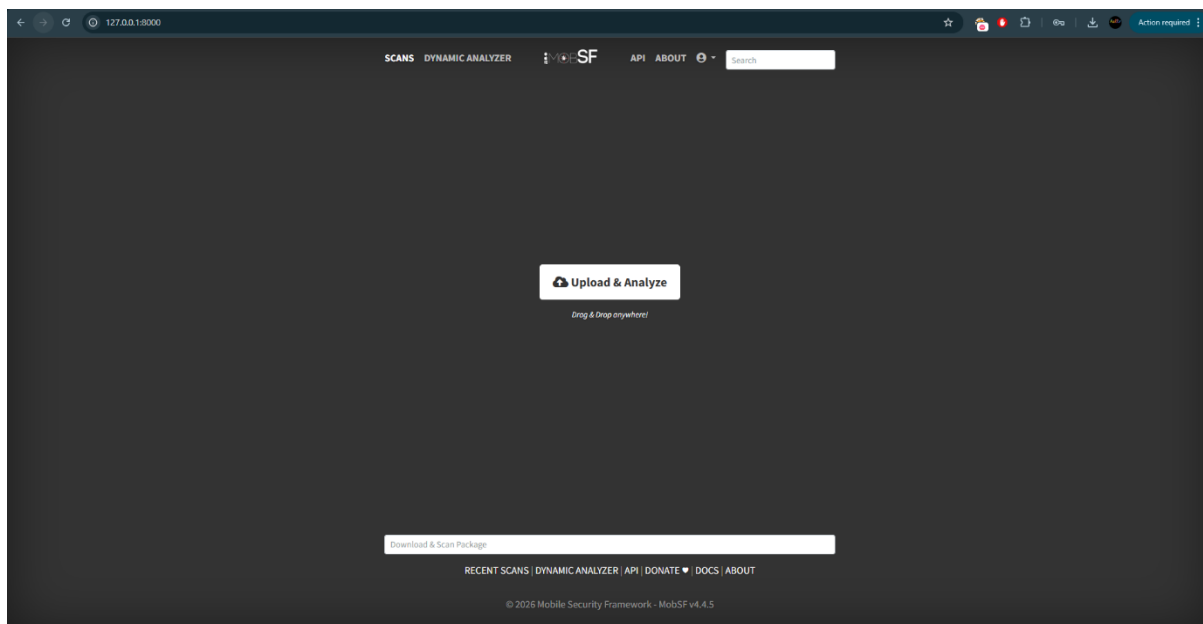
```
C:\Users\91977\OneDrive\Desktop\Applications\Cybersecurity\Android Pen Testing\MobSF\Mobile-Security-Framework-MobSF-master>run.bat 127.0.0.1:8000
Running MobSF on 127.0.0.1:8000
[INFO] 27/Feb/2026 07:26:09 - Loading User config from: C:/Users/91977/.MobSF/config.py
[INFO] 27/Feb/2026 07:26:27 -

  MOBSF

[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: 220615dc2b673495e72271d6debac42d5dc2b071df4fe4b0996fff141337ea3
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 ANALYZER, API, DONATE, DOCS, and ABOUT. The left sidebar lists various analysis options: Information, Scan Options, Signer Certificate, Permissions, Android API, Browsable Activities, Security Analysis, Malware Analysis, Reconnaissance, Components, PDF Report, Print Report, and Start Dynamic Analysis. The main content area displays the following sections:

- APP SCORES:** Security Score 48/100, Features Score 0/432.
- FILE INFORMATION:** File Name: AndroGoat.apk, Size: 6.77MB, SHA1: 3a3254abf55e4a4d4092c68ac72656, SHA256: 7b6736a389c0de5acc5351a28b9373887b59, MD5: 3edf8b538b2874d494c0dc0cadd5b05a544b59d2c7eb055073892e6a4f0e0.
- APP INFORMATION:** App Name: AndroGoat - Insecure App (Kotlin), Package Name: twarp.sst.agcat, Manifest Name: twarp.sst.agcat.SplashActivity, Target SDK: 33, Min SDK: 19, API Level: 33, Android Version Name: 13.0, Android Version Code: 1.
- EXPORTED ACTIVITIES:** 1/30 (View All).
- EXPORTED SERVICES:** 1/1 (View All).
- EXPORTED RECEIVERS:** 2/2 (View All).
- EXPORTED PROVIDERS:** 1/2 (View All).
- SCAN OPTIONS:** Buttons for Rescan, Manage Suppressors, Start Dynamic Analysis, and Scan Logs.
- DECOMPILED CODE:** Buttons for View AndroidManifest.xml, View Source, View Strings, Download Java Code, Download Smali Code, and Download APK.
- SIGNER CERTIFICATE:** Binary is signed, v1 signature: True, v2 signature: True, v3 signature: False, v4 signature: False, X.509 Subject: CN=Android Debug, OU=Android, C=US, Signature Algorithm: rsaesha\_sha256, Signature Algorithm: rsaesha\_sha256, Valid From: 2021-11-12 09:10:00+00:00, Valid To: 2025-11-12 09:10:00+00:00, Issuer: CN=Android Debug, OU=Android, C=US, Serial Number: 801, Hash Algorithm: sha256, MD5: 54172d5227f8f086e0113428baf5, SHA1: 4d5b2d2c43280c0150f91f09097332c0c, SHA256: 8f5e6e38c2021f40800150f91f09097332c0c, SHA512: 8f5e6e38c2021f40800150f91f09097332c0c, SHA512: 8f5e6e38c2021f40800150f91f09097332c0c.

## 4. Review the Security Analysis tab

The screenshot shows the MobSF Static Analyzer Security Analysis tab. The top navigation bar includes links for RECENT SCANS, STATIC ANALYZER, DYNAMIC ANALYZER, API, DONATE, DOCS, and ABOUT. The left sidebar lists various analysis options: Information, Scan Options, Signer Certificate, Permissions, Android API, Browsable Activities, Security Analysis, Malware Analysis, Reconnaissance, Components, PDF Report, Print Report, and Start Dynamic Analysis. The main content area displays the following sections:

- CODE ANALYSIS:** A table showing security issues with columns for NO, ISSUE, SEVERITY, STANDARDS, FILES, and OPTIONS.

NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	App creates temp file. Sensitive information should never be written into a temp file.	Warning	CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: HSTG-STORAGE-2	owasp\src\app\InsecureStorage\DefaultActivity.java owasp\src\app\InsecureStorage\TempActivity.java	[Icon]
2	The App logs information. Sensitive information should never be logged.	Info	CWE-532: Inclusion of Sensitive Information into Log File OWASP MASVS: HSTG-STORAGE-3	owasp\src\app\TrafficActivity\DefaultActivity.java	[Icon]
3	This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.	Secure	OWASP MASVS: HSTG-NETWORK-4	owasp\src\app\TrafficActivity\DefaultActivity.java	[Icon]
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-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M1: Client Code Quality	owasp\src\app\ContentProvider\DefaultActivity.java owasp\src\app\InsecureStorage\DefaultActivity.java owasp\src\app\SQLInjectionActivity.java	[Icon]
5	Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	Warning	CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: HSTG-STORAGE-14	owasp\src\app\ContentProvider\DefaultActivity.java	[Icon]
6	MD5 is a weak hash known to have hash collisions.	Warning	CWE-327: Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: HSTG-CRYPTO-4	owasp\src\app\ContentProvider\DefaultActivity.java	[Icon]
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-200: Information Exposure OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: HSTG-PLATFORM-7	owasp\src\app\WebView\WebViewDefaultActivity.java	[Icon]
8	This App copies data to clipboard. Sensitive data should not be copied to clipboard as other applications can access it.	Info	OWASP MASVS: HSTG-STORAGE-10	owasp\src\app\CliBoardActivity.java	[Icon]
9	App can read/write to External Storage. Any App can read data written to External Storage.	Warning	CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: HSTG-STORAGE-2	owasp\src\app\InsecureStorage\DefaultActivity.java	[Icon]
10	This App may request root (Super User) privileges.	Warning	CWE-250: Execution with Unnecessary Privileges OWASP MASVS: HSTG-RESILIENCE-1	owasp\src\app\RootAccessActivity.java	[Icon]



## 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.