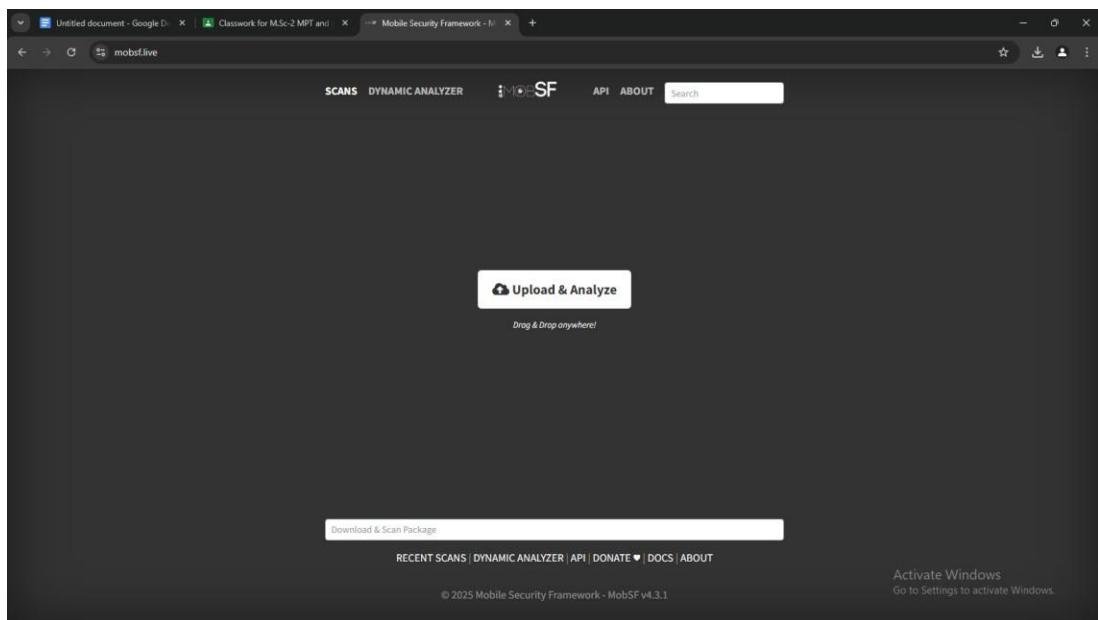


PRACTICAL 7

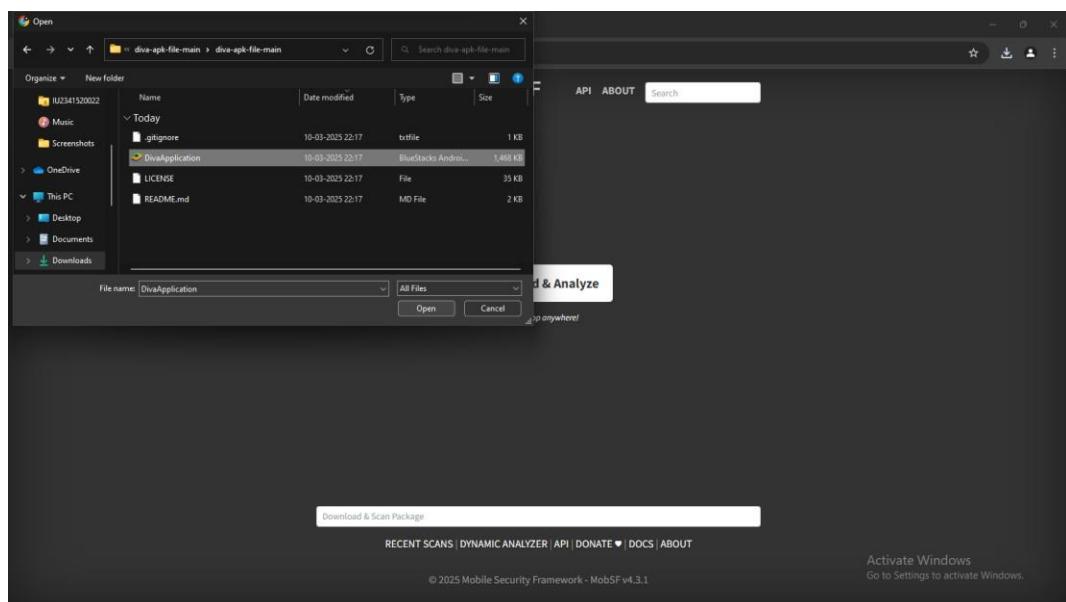
Aim: Install Mobsf and do Static/Dynamic Analysis

Mobsf: **Mobsf** (Mobile Security Framework) is an open-source automated mobile application security testing framework designed to perform static and dynamic analysis of mobile applications. It is primarily used for identifying security vulnerabilities in Android and iOS applications.

Step 1: Open Mobsf. Live Web Application and start static analysis for Android Analysis.



Step 2: Upload A base Andriod Application and analyze that the App is malicious or Not ?



Step 3: Now the Mobsf.live will give the Scan report of the android application.

The screenshot shows the 'Scan Queue' section of the Mobsf.live web interface. It lists several completed tasks with their details:

- Task ID:** ae61ad6c0625462eacd3823b3d6425d91
Checksum: fbd4a11b1343bfef97622923553cc170d
Filename: com.random.chat.app-4.0.0.apk
Timeline: Queued At: 3/11/2025, 10:12:03 AM; Started At: 3/11/2025, 10:12:03 AM; Completed At: 3/11/2025, 10:13:23 AM
Status: Success (View Report, RandoChat (com.random.chat.app))
- Task ID:** 65defcf5e59d4e3fb2c784849758970
Checksum: 71d534986d8df088d1244cac78e83cc
Filename: apk2.apk
Timeline: Queued At: 3/11/2025, 10:05:12 AM; Started At: 3/11/2025, 10:05:13 AM; Completed At: 3/11/2025, 10:07:18 AM
Status: Success (View Report, VLC (org.videolan.vlc))
- Task ID:** d25c7169e121479e93885ce058ae9ca
Checksum: 20d17739551af603b2506329b85e830
Filename: com.nhkorindo.mts_244_apps.evozi.com._apk
Timeline: Queued At: 3/11/2025, 10:04:58 AM; Started At: 3/11/2025, 10:04:58 AM; Completed At: 3/11/2025, 10:07:18 AM
Status: Success (View Report, Nalk (com.nhkorindo.mts))
- Task ID:** 2965b378e17d48ff883bd9709493843
Checksum: 94de7448eb34f446d513cb984467243
Filename: ebitb2c-debug.apk
Timeline: Queued At: 3/11/2025, 9:55:19 AM; Started At: 3/11/2025, 9:55:19 AM; Completed At: 3/11/2025, 9:58:43 AM
Status: Success (View Report, EbitCash-Debug (app.ebitcash.debug))
- Task ID:** 954a8528de494b12ace915618577db12
Checksum: 82ab8b2193b3cfbc1c737e3786be363a
Filename: DivaApplication.apk
Timeline: Queued At: 3/11/2025, 9:40:22 AM; Started At: 3/11/2025, 9:40:23 AM; Completed At: 3/11/2025, 9:41:00 AM
Status: Success (View Report, Diva (jakhar.aseem.diva))
- Task ID:** d728985f7fa2450b9eld3f1a2a15747
Checksum: y1mtieky.apk
Filename: y1mtieky.apk
Timeline: Queued At: 3/11/2025, 9:39:19 AM
Status: Success

Step 4: Now mobsf.live will give you the report of the application.

The screenshot shows the 'Static Analysis' page for the 'Diva' application. It displays various analysis results and options:

- APP SCORES:** Security Score: 36/100, Testers Score: 0/432, Mobsf Scorecard.
- FILE INFORMATION:** File Name: DivaApplication.apk, Size: 1.43MB, MD5: 82ab8b2193b3cfbc1c737e3786be363a, SHA1: 27e49d97d7b6da3a1357e3b980433a91d416801, SHA256: 5cef51fce9bd760b92ab2340477fdada84ae05d0a8c9493e4fe34fab7c5.
- APP INFORMATION:** App Name: Diva, Package Name: jakhar.aseem.diva, Main Activity: jakhar.aseem.diva.MainActivity, Target SDK: 23, Min SDK: 15, Max SDK: 1, Android Version Name: 1.0, Android Version Code: 1.
- EXPORTED ACTIVITIES:** 2 / 17 (A/A) View All.
- EXPORTED SERVICES:** 0 / 0 (Gears) View All.
- EXPORTED RECEIVERS:** 0 / 0 (Speaker) View All.
- EXPORTED PROVIDERS:** 1 / 1 (Database) View All.
- SCAN OPTIONS:** Rescan, Manage Suppressions, Start Dynamic Analysis, Scan Log.
- DECOMPILED CODE:** View AndroidManifest.xml, View Source, View Small, Download Java Code, Download Smali Code, Download APK.
- SIGNER CERTIFICATE:** Binary is signed, v1 signature: True, v2 signature: False, v3 signature: False, v4 signature: False, X.509 Subject: C=US, O=Android, CN=Android Debug, Signature Algorithm: rsassa_pkcs1v15, Valid From: 2015-11-02 08:32:11+00:00, Valid To: 2045-10-25 08:32:11+00:00, Issuer: C=US, O=Android, CN=Android Debug, Serial Number: 0x181330df, Hash Algorithm: sha256, MD5: d520162ac4ee974d7fd3a1862e7e4df, SHA1: ae4eadseba4e4efc928e7cf7f7d459f000881, SHA256: 35d7f7d35df62b67fb473187d47854932c080c5653b86568029fc5d5840, SHA512: e936165585937a244e3933d2b9d2101432de5b07dc05b277b46e443eff3f730d4fsa25b61f78c9078f54cc325cb086c17160b5bae13148d1e.

Step 5: Mobsf. Live Web application will provide you types of vulnerabilities in the application.

The screenshot shows the 'SIGNER CERTIFICATE' page for the Diva application. It displays detailed information about the certificate:

- BINARY IS SIGNED:**
- V1 SIGNATURE:** True
- V2 SIGNATURE:** False
- V3 SIGNATURE:** False
- V4 SIGNATURE:** False
- X.509 SUBJECT:** C=US, O=Android, CN=Android Debug
- SIGNATURE ALGORITHM:** rsassa_pkcs1v15
- VALID FROM:** 2015-11-02 08:32:11+00:00
- VALID TO:** 2045-10-25 08:32:11+00:00
- ISSUER:** C=US, O=Android, CN=Android Debug
- SERIAL NUMBER:** 0x181330df
- HASH ALGORITHM:** sha256
- MD5:** d520162ac4ee974d7fd3a1862e7e4df
- SHA1:** ae4eadseba4e4efc928e7cf7f7d459f000881
- SHA256:** 35d7f7d35df62b67fb473187d47854932c080c5653b86568029fc5d5840
- SHA512:** e936165585937a244e3933d2b9d2101432de5b07dc05b277b46e443eff3f730d4fsa25b61f78c9078f54cc325cb086c17160b5bae13148d1e
- FOUND:** 1 unique certificates

Step 6: The Permission of Application and Android API

The screenshot shows the MobSF interface under the 'Static Analyzer' tab. On the left, a sidebar lists various analysis options. The main panel displays two tables: 'APPLICATION PERMISSIONS' and 'ANDROID API'.

APPLICATION PERMISSIONS:

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

ANDROID API:

API	FILES
Content Provider	Show File
Inter Process Communication	Show File
Loading Native Code (Shared Library)	Show File
Local File I/O Operations	Show File
Starting Activity	Show File

Step 7: The manifest analysis of the application

The screenshot shows the MobSF interface under the 'Static Analyzer' tab. The main panel displays a table titled 'MANIFEST ANALYSIS' with four severity categories: HIGH, WARNING, INFO, and SUPRESSED.

NO	ISSUE	SEVERITY	DESCRIPTION	OPTIONS
1	App can be installed on a vulnerable uppatched Android version	High	This application can be installed on an older version of android that has multiple unfixed vulnerabilities. These devices won't receive reasonable security updates from Google. Support an Android version >= 10, API 29 to receive reasonable security updates.	[Icon]
2	Debug Enabled For App	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.	[Icon]
3	Application Data can Be backed up	warn	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.	[Icon]
4	Activity (<code>jakhar.aseem.diva.APIcredsActivity</code>) is not Protected. An intent filter exists.	warn	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.	[Icon]
5	Activity (<code>jakhar.aseem.diva.APIcreds2Activity</code>) is not Protected. An intent-filter exists.	warn	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.	[Icon]
6	Content Provider (<code>jakhar.aseem.diva.NotesProvider</code>) is not Protected.	warn	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.	[Icon]

Step 8: The Application code Analysis:

The screenshot shows the MobSF interface under the 'Static Analyzer' tab. The main panel displays a table titled 'CODE ANALYSIS' with five severity categories: HIGH, WARNING, INFO, SECURE, and SUPRESSED.

NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	Debug configuration enabled. Production builds must not be debuggable.	High	CWE: CWE-919: Weaknesses in Mobile Applications OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: MSTG-RESILIENCE-2	<code>jakhar.aseem.diva/BuildConfig.java</code>	[Icon]
2	The App logs information. Sensitive information should never be logged.	Info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP Top 10: M2: Insecure Data Storage	<code>jakhar.aseem.diva/InsecureDataStorageActivity.java</code>	[Icon]
3	App can read/write to External Storage. Any App can read data written to External Storage.	warn	CWE: CWE-776: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	<code>jakhar.aseem.diva/InsecureDataStorageActivity.java</code>	[Icon]
4	App creates temp file. Sensitive information should never be written into a temp file.	warn	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	<code>jakhar.aseem.diva/InsecureDataStorageActivity.java</code>	[Icon]
5	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.	warn	CWE: CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	<code>jakhar.aseem.diva/InsecureDataStorageActivity.java</code> <code>jakhar.aseem.diva/NotesProvider.java</code> <code>jakhar.aseem.diva/SQLInjectionActivity.java</code>	[Icon]

Step 9: Shared library Binary Analysis

SHARED LIBRARY BINARY ANALYSIS									
NO.	SHARED OBJECT	NX	PIE	STACK CANARY	RELRO	RPATH	RUNPATH	FORTIFY	SYMBOLS STRIPPED
1	mips/libdivajni.so	False [Q] Analyze [W]	Dynamic Shared Object (DSO) [Info] [High]	False [Info] [High]	Full RELRO [Info]	None [Info]	None [Info]	False [Warning]	True [Info] Symbols are stripped.
2	armeabi-v7a/libdivajni.so	False [Q] Analyze [W]	Dynamic Shared Object (DSO) [Info] [High]	False [Info] [High]	Full RELRO [Info]	None [Info]	None [Info]	False [Warning]	True [Info] Symbols are stripped.
3	armeabi/libdivajni.so	False [Q] Analyze [W]	Dynamic Shared Object (DSO) [Info] [High]	False [Info] [High]	Full RELRO [Info]	None [Info]	None [Info]	False [Warning]	True [Info] Symbols are stripped.
4	x86/libdivajni.so	False [Q] Analyze [W]	Dynamic Shared Object (DSO) [Info] [High]	True [Info] [Low]	Full RELRO [Info]	None [Info]	None [Info]	False [Warning]	True [Info] Symbols are stripped.

Conclusion:

Once MobSF is running, upload your APK or IPA file for static analysis. After the analysis, review the report for key findings related to permissions, code issues, and security risks. The conclusion will depend on the specific vulnerabilities and security issues identified in the report.