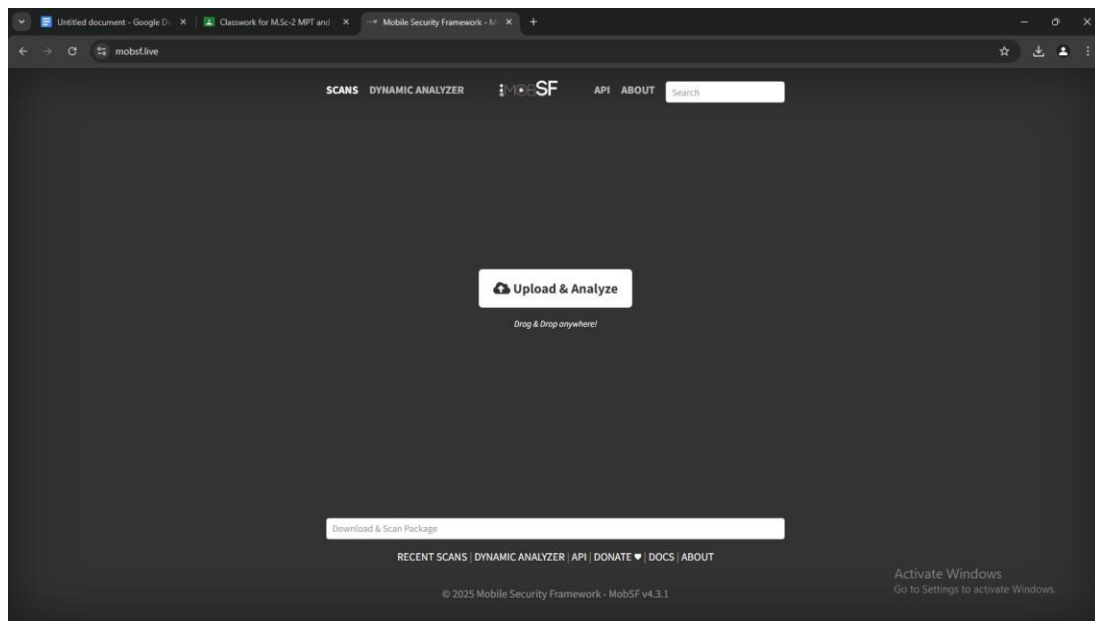


PRACTICAL 7

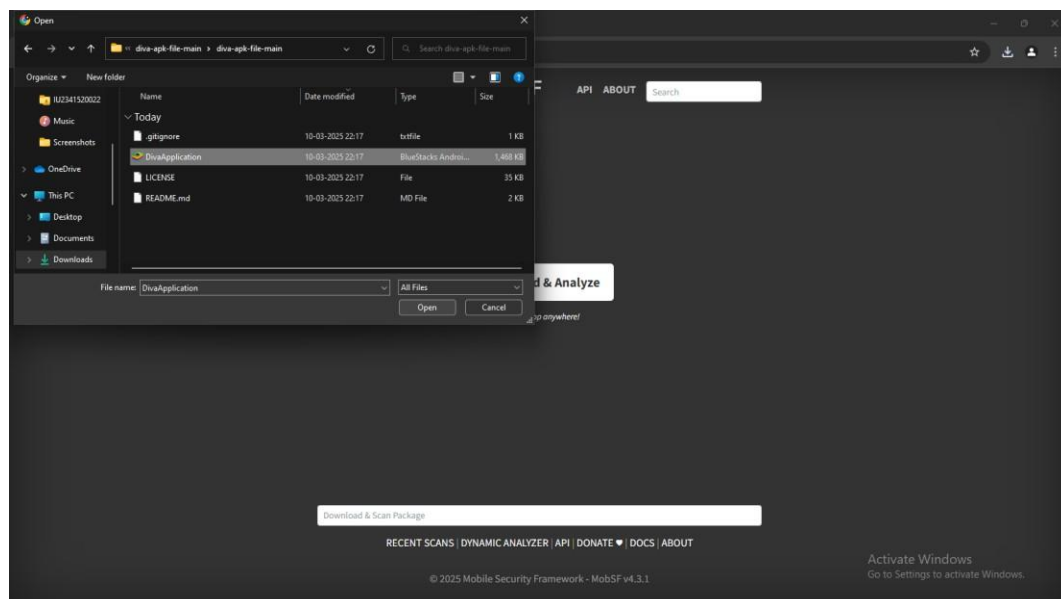
Aim: Install Mobsf and do Static/Dynamic Analysis

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.

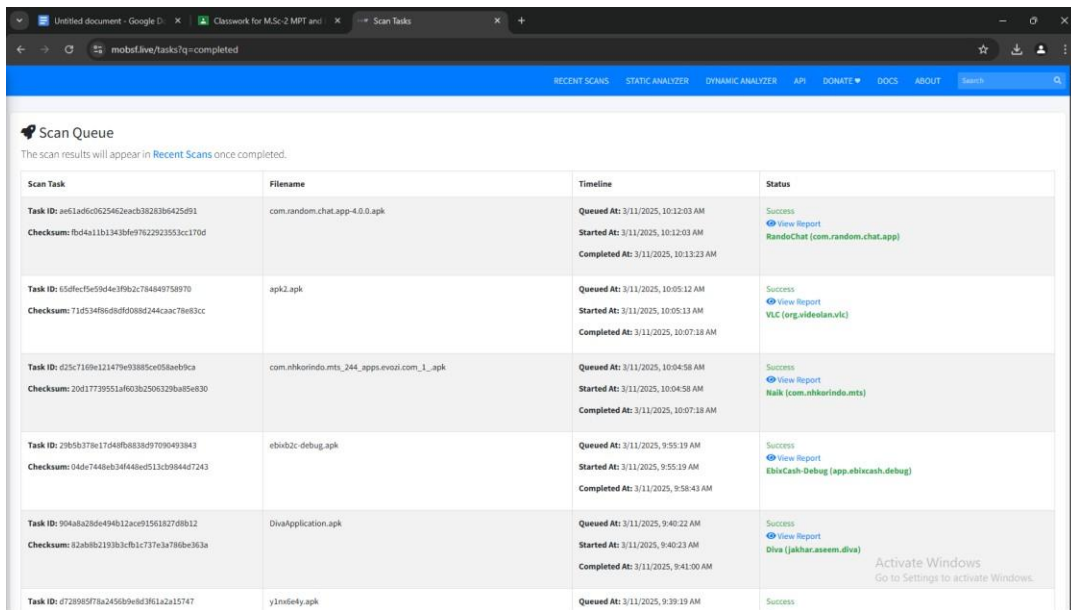
Step 1: Open Mobsf. Live Web Application and start static analysis for Android Analysis.



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

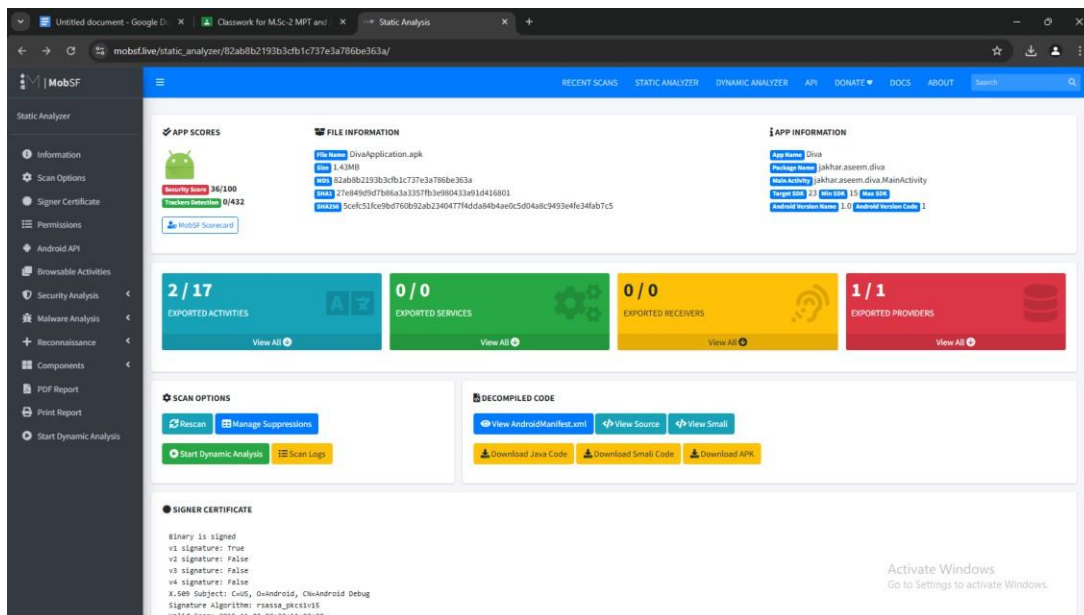


Step 3: Now the Mobsf.live will give the Scan report of the android application.



Task ID	Filename	Timeline	Status
Task ID: ae61ad6c625462eac38283b6425d91 Checksum: fbd4a11b1343bf9762292353cc170d	com.random.chat.app-4.0.0.apk	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	Success View Report RandeChat [com.random.chat.app]
Task ID: 65dfe959d4e3f962c784849758970 Checksum: 71d534f86dd8d088244caac78e93cc	apk2.apk	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	Success View Report VLC [org.videolan.vlc]
Task ID: d25c7169e121479e93885ce058a6b1ca Checksum: 20d17739551a603b2506329ba85e830	com.nhkorinda.mts_244_apps.evsnl.com_1.apk	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	Success View Report Nalik [com.nhkorinda.mts]
Task ID: 29b5b379e17d48fb683d97090493843 Checksum: 04de7448eb34f448e0513cb99467243	ebivb2c-debug.apk	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	Success View Report EbiivCash-Debug [app.ebiivcash.debug]
Task ID: 904ba286e49b12ace91561827d8b12 Checksum: 82ab8b2193b3cfc1c737e3a786be363a	DivaApplication.apk	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	Success View Report Diva [jakhar.assem.diva]
Task ID: d72895f78a2458b9e8d3f61a2a15747	y3nfs6ky.apk	Queued At: 3/11/2025, 9:39:19 AM	Success

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



Static Analysis

APP SCORES
Security Score: 34/100
Tracker Detection: 0/432
MobSF Scorecard

FILE INFORMATION
File Name: DivaApplication.apk
Size: 1.43MB
SHA1: 82ab8b2193b3cfc1c737e3a786be363a
SHA256: 71d534f86dd8d088244caac78e93cc
MD5: Scef51f6c9b0d760b92ab2340477f4dda84b4a6c5d048c94934f63fab7c5

APP INFORMATION
App Name: Diva
Package Name: jakhar.assem.diva
Main Activity: jakhar.assem.diva.MainActivity
Target SDK: 33 (Android 13)
Android Version Range: 1.0 (Required Version Code: 1)

2/17 EXPORTED ACTIVITIES
0/0 EXPORTED SERVICES
0/0 EXPORTED RECEIVERS
1/1 EXPORTED PROVIDERS

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: True
v2 signature: False
v3 signature: False
v4 signature: False
X.509 Subject: C=US, O=Android, CN=Android Debug
Signature Algorithm: rsaesha_pkcs1v15
X.509 Subject: C=US, O=Android, CN=Android Debug
Signature Algorithm: rsaesha_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: 0x218380df
Hash Algorithm: sha256
MDS: d628162ac34ee974f93d1862a7e4df
SHA1: ae4e0d5ae0b4e4e4efc926e7c7f7f6459f980851
SHA256: 35d7f7ad35dfb26b70f4b73187d478540e32c8b6c56b3b6568029fcd540
SHA512: e93619585893a7a248e393d02b2d1014320e5b07c3d7e2bdc0070c0b58277b46e443eff3730b4f5a25b61f78c907f54cc325cb86c17160b5bae1314801e
Found 1 unique certificates

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



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: rsaesha_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: 0x218380df
Hash Algorithm: sha256
MDS: d628162ac34ee974f93d1862a7e4df
SHA1: ae4e0d5ae0b4e4e4efc926e7c7f7f6459f980851
SHA256: 35d7f7ad35dfb26b70f4b73187d478540e32c8b6c56b3b6568029fcd540
SHA512: e93619585893a7a248e393d02b2d1014320e5b07c3d7e2bdc0070c0b58277b46e443eff3730b4f5a25b61f78c907f54cc325cb86c17160b5bae1314801e
Found 1 unique certificates

Step 6: The Permission of Application and Android API

The screenshot displays the MobSF Static Analyzer interface. The left sidebar contains navigation options: Information, Scan Options, Signer Certificate, Permissions, Android API, Browseable Activities, Security Analysis, Malware Analysis, Reconnaissance, Components, PDF Report, Print Report, and Start Dynamic Analysis. The main content area is divided into two sections: 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	dangerous	read external storage contents	Allows an application to read from external storage.	
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.	

Showing 1 to 3 of 3 entries

ANDROID API

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

Showing 1 to 5 of 5 entries

Step 7: The manifest analysis of the application

The screenshot displays the MobSF Static Analyzer interface, specifically the MANIFEST ANALYSIS section. The left sidebar is the same as in Step 6. The main content area shows a table of manifest analysis results with columns: NO, ISSUE, SEVERITY, DESCRIPTION, and OPTIONS.

Summary: HIGH 2, WARNING 4, INFO 0, SUPPRESSED 0

NO	ISSUE	SEVERITY	DESCRIPTION	OPTIONS
1	App can be installed on a vulnerable patched Android version Android 4.0.3-4.0.4, [minSdk=15]	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 >= 16, API 29 to receive reasonable security updates.	Show Files
2	Debug Enabled For App [android:debuggable=true]	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.	Show Files
3	Application Data can be Backed up [android:allowBackup=true]	warning	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.	Show Files
4	Activity [jakhar.aseem.diva.APICred3Activity] is not Protected. An intent filter exists.	warning	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.	Show Files
5	Activity [jakhar.aseem.diva.APICred3Activity] is not Protected. An intent filter exists.	warning	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.	Show Files
6	Content Provider [jakhar.aseem.diva.NotesProvider] is not Protected. [android:exported=true]	warning	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.	Show Files

Showing 1 to 6 of 6 entries

Step 8: The Application code Analysis:

The screenshot displays the MobSF Static Analyzer interface, specifically the CODE ANALYSIS section. The left sidebar is the same as in Step 6. The main content area shows a table of code analysis results with columns: NO, ISSUE, SEVERITY, STANDARDS, FILES, and OPTIONS.

Summary: HIGH 1, WARNING 3, INFO 1, SECURE 0, SUPPRESSED 0

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	jakhar.aseem.diva.BuildConfig.java	Show Files
2	The App logs information. Sensitive information should never be logged.	info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3	jakhar.aseem.diva/InsecureDataStorageActivity.java	Show Files
3	App can read/write to External Storage. Any App can read data written to External Storage.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	jakhar.aseem.diva/InsecureDataStorage2Activity.java	Show Files
4	App creates temp file. Sensitive information should never be written into a temp file.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	jakhar.aseem.diva/InsecureDataStorage2Activity.java	Show Files
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.	warning	CWE: CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality	jakhar.aseem.diva/InsecureDataStorage2Activity.java jakhar.aseem.diva/SQLInjectionActivity.java	Show Files

Showing 1 to 5 of 5 entries

Step 9: Shared library Binary Analysis

The screenshot shows the MobSF Static Analyzer interface. The main panel displays a table titled "SHARED LIBRARY BINARY ANALYSIS" with columns for NO, SHARED OBJECT, NX, PIE, STACK CANARY, RELRO, RPATH, RUNPATH, FORTIFY, and SYMBOLS STRIPPED. The table lists four shared objects: mips/libdvajni.so, arm64-v7a/libdvajni.so, arm64/libdvajni.so, and x86/libdvajni.so. Each row provides a detailed analysis of the binary's security features and potential vulnerabilities.

NO	SHARED OBJECT	NX	PIE	STACK CANARY	RELRO	RPATH	RUNPATH	FORTIFY	SYMBOLS STRIPPED
1	mips/libdvajni.so Q.Ansari	False The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. Use option -noexecstack or -z noexecstack to mark stack as non-executable.	Dynamic Shared Object (DSO) The shared object is built with-EPIC flag which enables Position Independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.	False This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries. Not applicable for Dart/Flutter libraries unless Dart FFI is used.	Full RELRO This shared object has full RELRO enabled. RELRO ensures that the GOT cannot be overwritten in vulnerable ELF binaries. In Full RELRO, the entire GOT (.got and .got.plt both) is marked as read-only.	None The binary does not have run-time search path or RPATH set.	None The binary does not have RUNPATH set.	False warning The binary does not have any fortified functions. Fortified functions provides buffer overflow checks against glibc's common insecure functions like strcpy, gets etc. Use the compiler option -D_FORTIFY_SOURCE=2 to fortify functions. This check is not applicable for Dart/Flutter libraries.	True Symbols are stripped.
2	arm64-v7a/libdvajni.so Q.Ansari	False The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. Use option -noexecstack or -z noexecstack to mark stack as non-executable.	Dynamic Shared Object (DSO) The shared object is built with-EPIC flag which enables Position Independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.	False This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries. Not applicable for Dart/Flutter libraries unless Dart FFI is used.	Full RELRO This shared object has full RELRO enabled. RELRO ensures that the GOT cannot be overwritten in vulnerable ELF binaries. In Full RELRO, the entire GOT (.got and .got.plt both) is marked as read-only.	None The binary does not have run-time search path or RPATH set.	None The binary does not have RUNPATH set.	False warning The binary does not have any fortified functions. Fortified functions provides buffer overflow checks against glibc's common insecure functions like strcpy, gets etc. Use the compiler option -D_FORTIFY_SOURCE=2 to fortify functions. This check is not applicable for Dart/Flutter libraries.	True Symbols are stripped.
3	arm64/libdvajni.so Q.Ansari	False The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable. Use option -noexecstack or -z noexecstack to mark stack as non-executable.	Dynamic Shared Object (DSO) The shared object is built with-EPIC flag which enables Position Independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.	False This binary does not have a stack canary value added to the stack. Stack canaries are used to detect and prevent exploits from overwriting return address. Use the option -fstack-protector-all to enable stack canaries. Not applicable for Dart/Flutter libraries unless Dart FFI is used.	Full RELRO This shared object has full RELRO enabled. RELRO ensures that the GOT cannot be overwritten in vulnerable ELF binaries. In Full RELRO, the entire GOT (.got and .got.plt both) is marked as read-only.	None The binary does not have run-time search path or RPATH set.	None The binary does not have RUNPATH set.	False warning The binary does not have any fortified functions. Fortified functions provides buffer overflow checks against glibc's common insecure functions like strcpy, gets etc. Use the compiler option -D_FORTIFY_SOURCE=2 to fortify functions. This check is not applicable for Dart/Flutter libraries.	True Symbols are stripped.
4	x86/libdvajni.so Q.Ansari	False The binary does not have NX bit set. NX bit offer protection against exploitation of memory corruption vulnerabilities by marking memory page as non-executable.	Dynamic Shared Object (DSO) The shared object is built with-EPIC flag which enables Position Independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.	True This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows	Full RELRO This shared object has full RELRO enabled. RELRO ensures that the GOT cannot be overwritten in vulnerable ELF	None The binary does not have run-time search path or	None The binary does not have RUNPATH set.	False warning The binary does not have any fortified functions. Fortified functions provides buffer overflow checks against glibc's common insecure functions like strcpy, gets etc. Use the compiler option	True 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.