

Mobsf vs our tool

Activity detection :

The screenshot displays the MobSF static analysis interface. The left sidebar shows the 'APP SCORES' section with a Security Score of 33/100. The main area is divided into 'FILE INFORMATION' and 'APP INFORMATION'. The 'APP INFORMATION' section lists details such as App Name (AES), Package Name (com.apphacking.aes), Main Activity (com.apphacking.aes.MainActivity), Target SDK (30), Min SDK (23), and Android Version Name (1.0). Below this, there are four colored boxes representing different components: EXPORTED ACTIVITIES (0/1), EXPORTED SERVICES (0/0), EXPORTED RECEIVERS (0/0), and EXPORTED PROVIDERS (0/0). The right sidebar shows a terminal window with the command 'python basic-activity_test12.py AES_androidVersion12.apk' and its output, which lists the package name and the main activity.

Deep links :

The screenshot displays the MobSF static analysis interface with a list of issues. The table below shows the issues:

NO	ISSUE	SEVERITY
1	App can be installed on a vulnerable unpatched Android version Android 7.0, [minSdk=24]	high
2	Clear text traffic is Enabled For App [android:usesCleartextTraffic=true]	high
3	App has a Network Security Configuration [android:networkSecurityConfig=@xml/network_security_config]	info
4	Debug Enabled For App [android:debuggable=true]	high
5	Application Data can be Backed up [android:allowBackup=true]	warning
6	Activity (com.apphacking.deeplinks.WebLink) is not Protected. [android:exported=true]	warning
7	Activity (com.apphacking.deeplinks.DeepLink) is not Protected. [android:exported=true]	warning

The right sidebar shows a detailed view of the issue for 'Activity (com.apphacking.deeplinks.WebLink) is not Protected'. The terminal output shows the command 'python basic-activity_test12.py AES_androidVersion12.apk' and its output, which lists the package name and the main activity.

Mobsf only detect that these activities (which handle the links) are exported and what intent invoke them but never tells if the deep links are hijackable or not which matters in both bug hunter's perspective and penetration tester

Permission check

The screenshot displays the Static Analyzer web interface on the left and a terminal window on the right. The web interface shows a table of Android permissions with columns for PERMISSION, STATUS, INFO, DESCRIPTION, and CODE MAPPINGS. The terminal window shows the output of a script analyzing permissions, with specific entries for `android.permission.READ_EXTERNAL_STORAGE` and `android.permission.WRITE_EXTERNAL_STORAGE` highlighted.

PERMISSION	STATUS	INFO	DESCRIPTION	CODE MAPPINGS
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.	Show Files
android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people.	Show Files
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows an application to read from external storage.	Show Files
android.permission.WAKE_LOCK	normal	prevent phone from sleeping	Allows an application to prevent the phone from going to sleep.	Show Files
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.	Show Files

Showing 1 to 5 of 5 entries

Previous 1 Next

```
(bughunt)kali@kali: ~/Desktop/automation-mobile/permissions
File Actions Edit View Help
[+] Pentest Analysis of Permissions
[+] Permission: android.permission.INTERNET
+ Risk : Can send or receive data over the internet.
+ Test :
  + Run app with proxy (Burp, mitmproxy) to inspect requests.
  + Look for plaintext credentials, tokens, or APIs.
+ Tools :
  + Burp Suite, mitmproxy, jadx
+ Note :
  + Check for insecure HTTP usage or broken certificate pinning.
[+] Permission: android.permission.READ_EXTERNAL_STORAGE
+ Risk : Can read user files on SD card (images, downloads, etc).
+ Test :
  + Look for data leaks by browsing files the app accesses.
  + Monitor runtime file access via 'logcat' or dynamic analysis.
+ Tools :
  + Frida, adb, logcat
+ Note :
  + May access unrelated apps' files on older Android versions.
[+] Permission: android.permission.WRITE_EXTERNAL_STORAGE
+ Risk : App can write files to shared external storage.
+ Test :
  + Use adb to inspect /sdcard/ and /storage/emulated/0/ for sensitive files.
  + Look for logs, tokens, DBs, or exported files.
+ Tools :
  + adb, file explorers, grep
+ Note :
  + External storage is world-readable on older Android versions.
[+] Permission: android.permission.READ_CONTACTS
+ No specific guidance found. Consider researching manually.
[+] Permission: android.permission.WAKE_LOCK
```

Our tool is more focused to guide the hunter to what test paths he can take (potentially can be ai agent in future) instead of analyzing the permission to check if it is malicious or not which is more related to malware analysis not penetration testing or bug hunting