



DOP: / /2023

DOS: / /2023

Experiment No:

Title: Android Manifest File Analysis and SDK Misuse detection using MobSF tool

Theory:

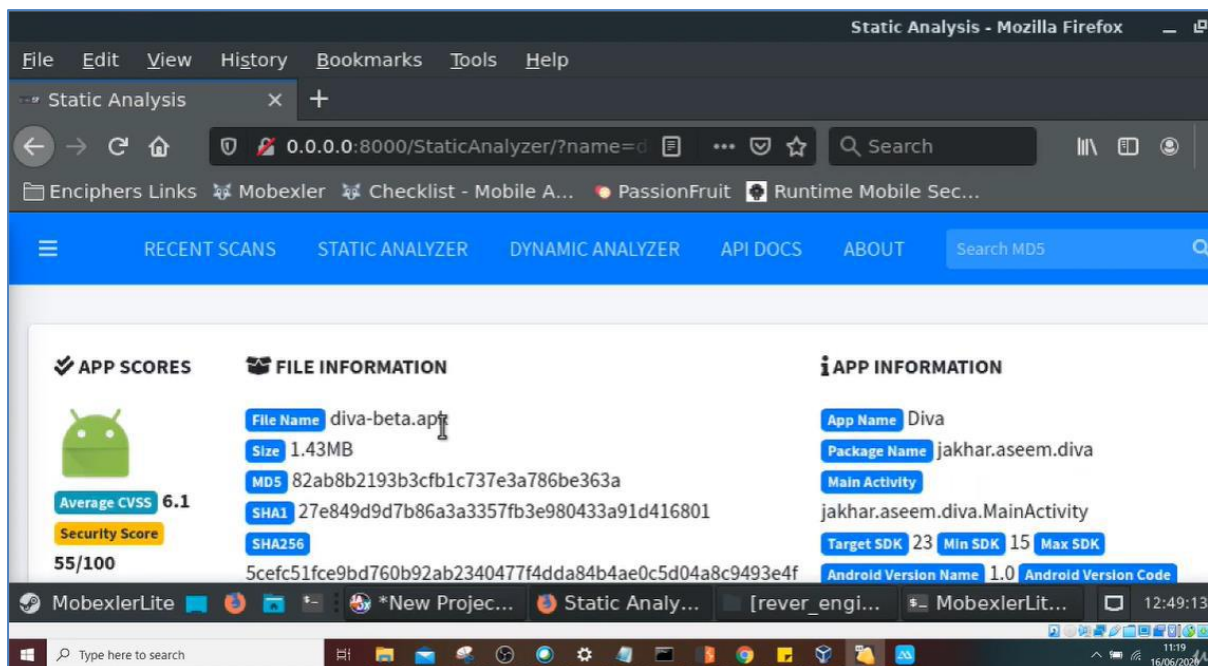
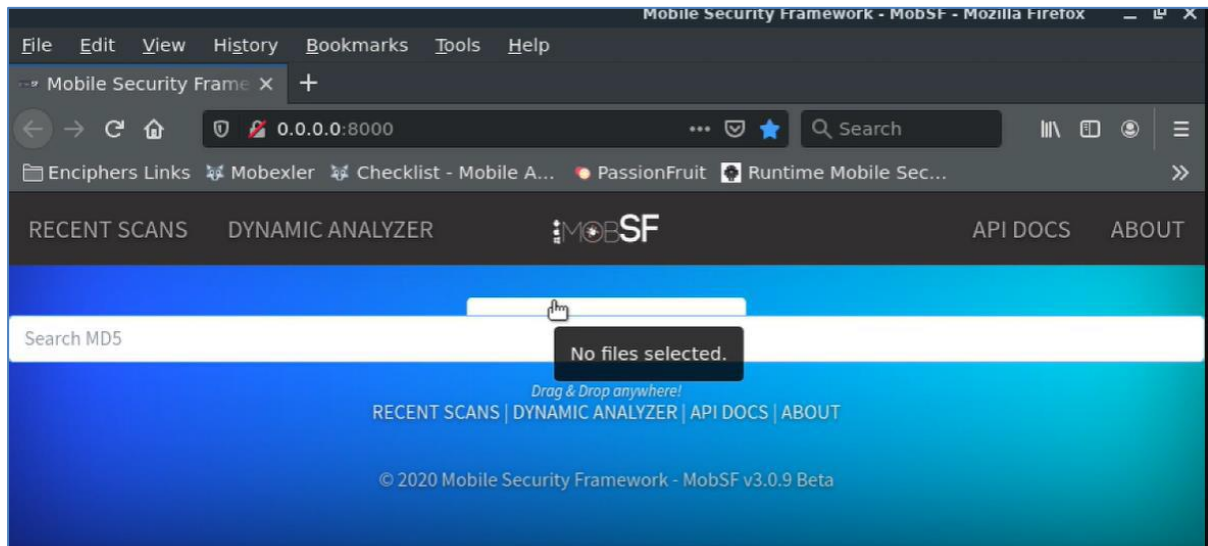
Mobile Security Framework (MobSF)

Performing Pentesting on Mobile Application by the Red Team means you are dealing with either APK (Android), IPA (iOS), or EXE (Windows), you need to have a vast knowledge of how you can perform not only automated tests and attacks but also Dynamic, which requires a special setup and specific tools, with special knowledge on how you can look under the hood within the source code itself, trying to find various range of vulnerabilities.

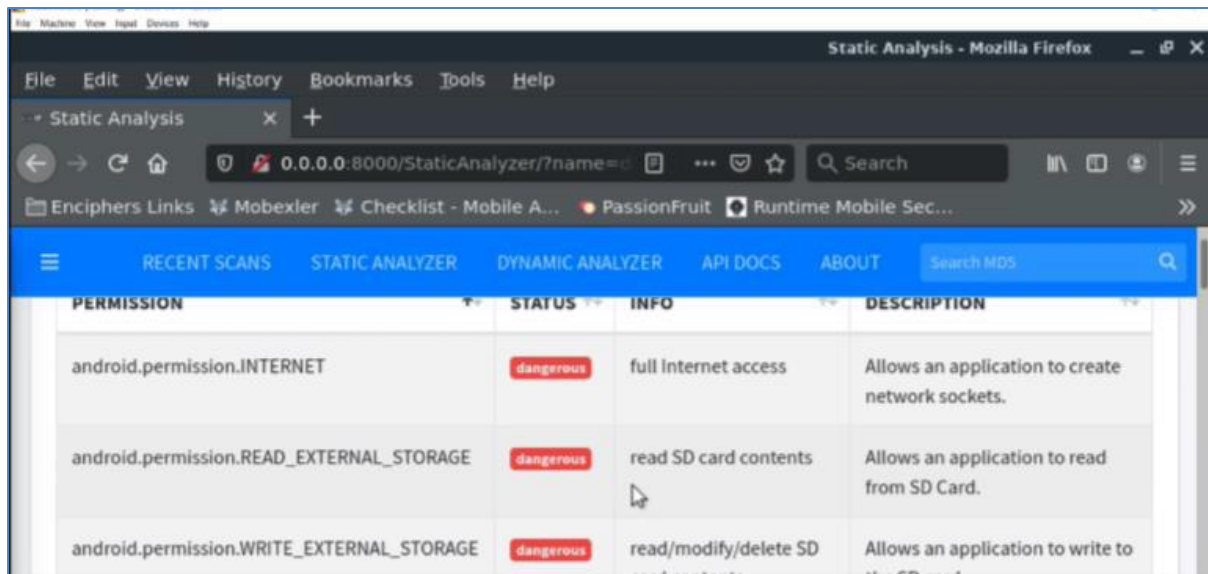
Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis. MobSF supports mobile app binaries (APK, XAPK, IPA & APPX) along with zipped source code and provides REST APIs for seamless integration with your CI/CD or DevSecOps pipeline. The Dynamic Analyzer helps you to perform runtime security assessment and interactive instrumented testing.

Some of the common security issues that can be detected by MobSF in an Android manifest file include:

- Excessive or unnecessary permissions: MobSF can identify permissions that are not needed for the functionality of the application or that can be abused by malicious actors to gain access to sensitive data or resources.
- Insecure export of components: MobSF can detect activities, services, and broadcast receivers that are exported with insecure or overly permissive settings, which can allow other applications to interact with them without proper authorization.
- Use of deprecated or vulnerable APIs: MobSF can flag the use of APIs that are no longer supported or have known security vulnerabilities, which can put the application and the device at risk of exploitation.

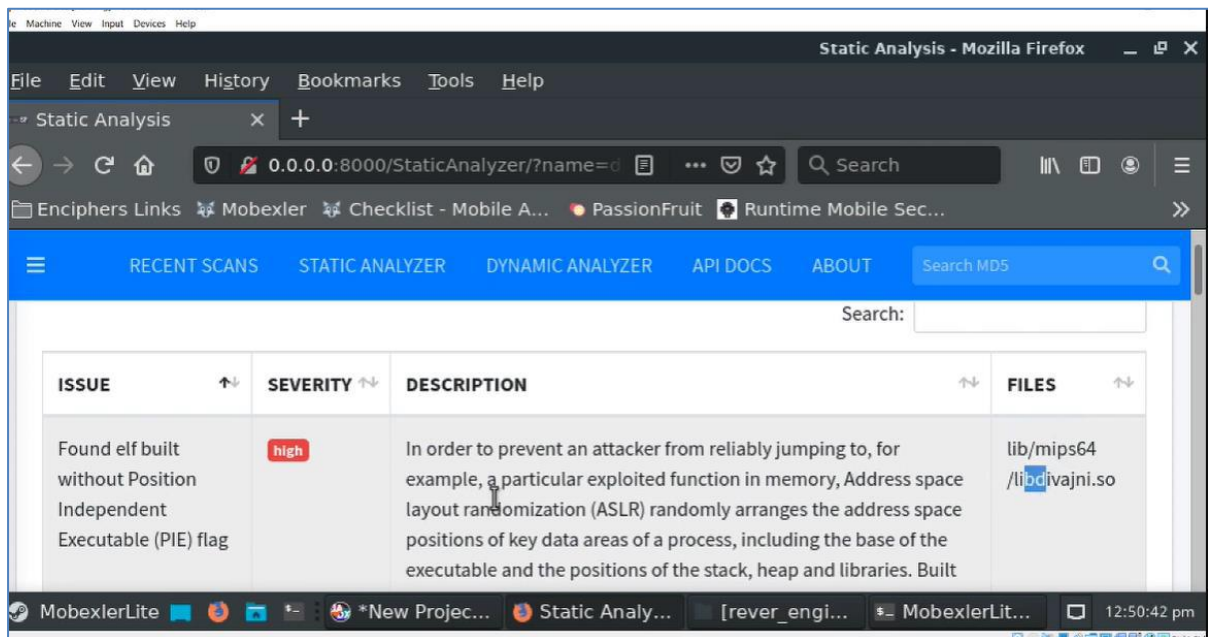


SDK misuse or abuse: MobSF can identify instances where the application is using third-party SDKs inappropriately or exposing sensitive data or functionality to them without proper safeguards or permissions.



PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.INTERNET	dangerous	full Internet access	Allows an application to create network sockets.
android.permission.READ_EXTERNAL_STORAGE	dangerous	read SD card contents	Allows an application to read from SD Card.
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete SD	Allows an application to write to the SD Card.

Android manifest file analysis is an essential part of the mobile application security assessment process, and tools like MobSF can help automate and streamline this process while providing valuable insights and actionable recommendations to developers and security teams.



ISSUE	SEVERITY	DESCRIPTION	FILES
Found elf built without Position Independent Executable (PIE) flag	high	In order to prevent an attacker from reliably jumping to, for example, a particular exploited function in memory, Address space layout randomization (ASLR) randomly arranges the address space positions of key data areas of a process, including the base of the executable and the positions of the stack, heap and libraries. Built	lib/mips64/libdivajni.so



To analyze an Android Manifest file and detect SDK misuse using MobSF, you can follow these steps:

- **Install MobSF:** MobSF is a mobile security framework that can be used to analyze mobile applications. You can download and install it from its official website (<https://mobsf.github.io/docs/installation/>).
- **Add the Android Manifest file:** To analyze the Android Manifest file of an Android application, you need to add the APK file to MobSF. You can either upload the APK file to the MobSF web interface or use the MobSF CLI to analyze the file.
- **Analyze the Android Manifest file:** Once the APK file is added to MobSF, you can click on the "Analyze" button to analyze the Android Manifest file. MobSF will parse the Manifest file and generate a report that includes information about the app's permissions, activities, services, and more.
- **Check for SDK Misuse:** MobSF can also detect SDK misuse in an Android application. To do this, you need to navigate to the "Vulnerabilities" section of the report and look for any entries related to SDK misuse. MobSF checks for common SDK-related vulnerabilities, such as using insecure SSL/TLS connections, storing sensitive information in shared preferences, and more.
- **Review the Report:** Once the analysis is complete, you can review the report generated by MobSF to identify any issues with the Android Manifest file or SDK misuse. The report will include a summary of the vulnerabilities found and recommendations on how to fix them.

Overall, MobSF is a powerful tool for analyzing Android applications and identifying security issues. By analyzing the Android Manifest file and checking for SDK misuse, you can identify potential security issues before they are exploited by attackers.

Conclusion: - Thus we have successfully studied the Android manifest file analysis and sdk misuse detecting using MobSF tool.