CS678 – Topics in Smartphone Security & Reliability

Project Proposal : MobileAudit

Submitted by – Kush Borikar (kb97)

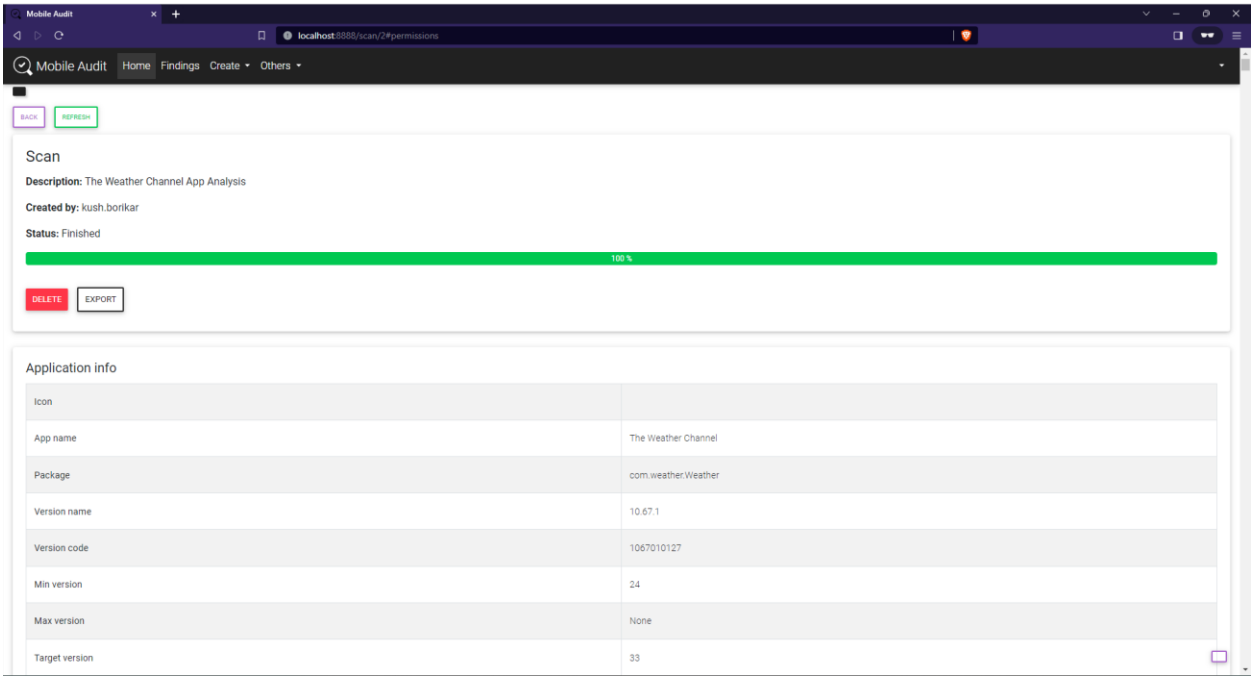
Project Idea: Analyzing the Security of an App set of various Android Apps using MobileAudit.

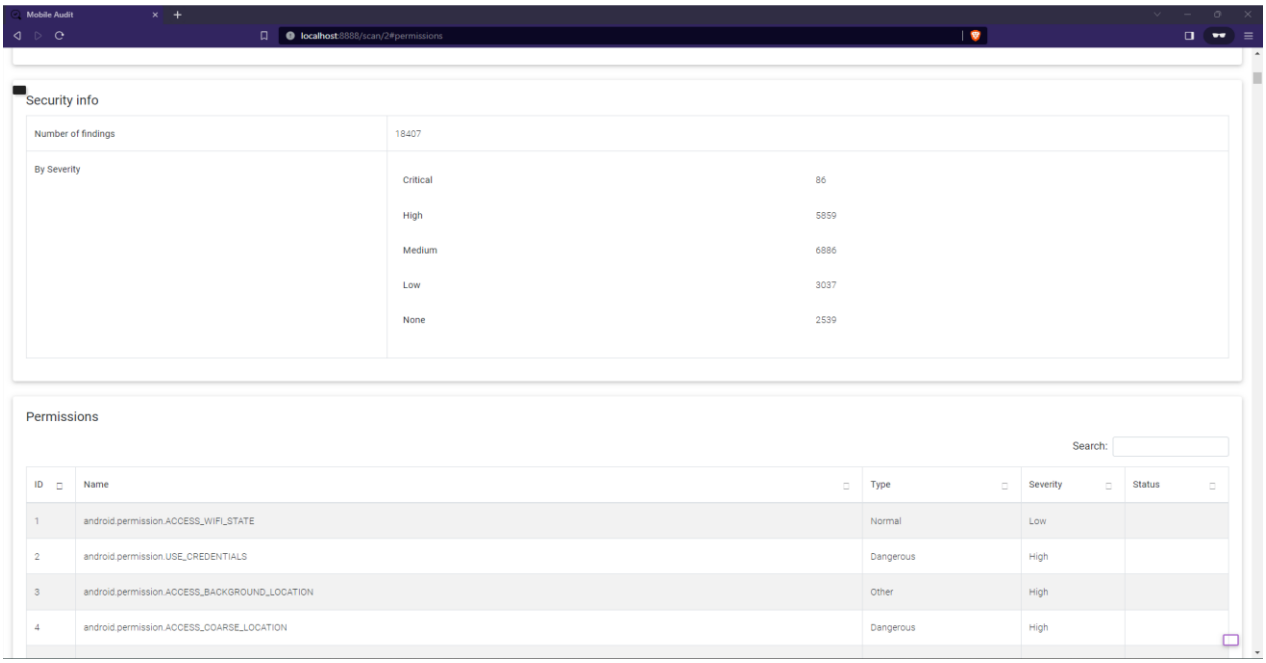
URL for the Tool: MobileAudit

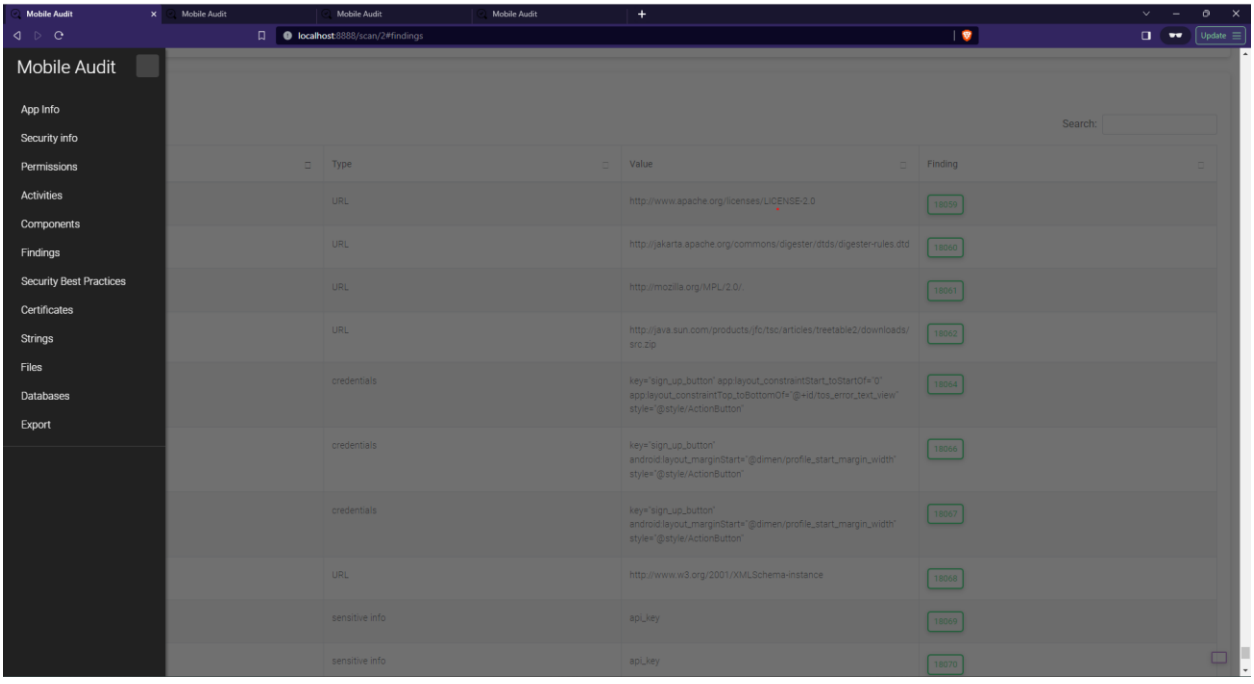
Project Overview: In this project, I will use MobileAudit to assess the security and reliability of selected Android apps. MobileAudit is an open-source Android app analysis tool developed by the Open Web Application Security Project (OWASP). MobileAudit can be used to perform static analysis of Android apps. Static analysis involves analyzing the app's code and resources. I will select a set of widely used Android applications and perform security analysis to identify potential vulnerabilities, misconfigurations, and reliability issues by conducting a static analysis. The goal is to gain insights into the state of security in the commonly used Android apps of the present.

Project Steps:

- **App Selection:** I will choose a set of Android applications from various categories such as social media, messaging, banking, or productivity from the Google Play Store or other trusted sources.
- **Environment Setup:** I have installed MobileAudit using Docker which runs the MobileAudit container on port 8888. I can access the dashboard by using the following URL on my web browser : http://localhost:8888







Analysis Overview:

- **Code Analysis:** The tool checks the byte code to review app logic and identify security issues in code like - checking if the device is debug-able, looking for exported components and root detection abilities. The tool identifies the code for patterns like Hardcoded API Keys, tokens, IP addresses.

- **Permission Set Analysis:** Analyze the categorized (Dangerous & Normal) type of permissions the application requests and uses, to see if these permissions pertain to the application's functionality or are they overprivileged and may cause security risks. Check using the tool for hardcoded credentials and if application is requesting superuser privileges, this is an indication of permission violation to forcibly take control of data and resources.

- **Data Access Analysis:** Check for sensitive data exposure. The tool identifies patterns like logging of sensitive information, operations like reading and writing of sensitive information to and from the device. These may include actions like getting device information (IMEI, MAC Address, etc.), reading clipboard data, and downloading files. These are concerning privacy threats.

- **Check for Secure Coding Practices:** Check for SQL injection vulnerabilities and if the app properly sanitizes SQL queries before executing them. The tool identifies Insecure functions and deserializations since this can lead to arbitrary code execution leading the attacker to take control of the program flow.

- **Network State Analysis:** Check the type and level of network permissions the application requests and whether or not apps adhere to standard network security configurations like using HTTPS traffic, and certificate verification using SSL pinning.

- **Cryptography Suite Analysis:** Check if app deploys secure cryptography practices. The tool identifies ciphers with no padding, ciphers with ECB, weak cryptography and hashing algorithms and insecure initialization vectors.

- **Check for XSS and TapJacking attacks:** Check for WebViews in applications, as they are vectors for cross site scripting attacks, that allow attackers to carry out any actions that the user is able to perform. Check android methods for fake actions to prevent TapJacking, as it leads to outcomes the user does not intend to do (like sharing sensitive data to a 3rd party website).

- **App-Set:** I will perform this analysis on an app-set of approximately 20 applications for the analysis outcome to be relevant and make sense of the overall security standing of these apps and also of how the tool functions on a high level.