**Week 8 – Lab: Android Application Static Analysis**

Learning objectives

* Comprehend and deploy the basic steps for penetration testing by utilising appropriate tools and techniques.
* Deploy, analyse and interpret the outcomes of vulnerability assessment tasks.
* Comprehend and use a pentesting framework such as MobSF (Mobile Security Framework).

**Introduction**

This lab aims to introduce you with the basics of static analysis of Android applications. You will use JADX-GUI to inspect an APK file, investigate its components (e.g., Manifest file, code structures, and network configurations), and identify vulnerabilities based on the **OWASP Mobile Top 10**.

Prerequisites: For this lab you are required to pull the **Injured Android application** (using Android studio) and use JADX-GUI to analyse it.

**Setup and open the APK**

* Launch JADX-GUI and load the **Injured Android** APK file
* Explore the file tree structure in JADX-GUI and locate files AndroidManifest.xml, .java files and resources.

**Task 1**: Investigate APK signature and determine the type of signature (e.g. V1, V2, V3) used, and write the **signature type**. **(maximum 5 marks)**

**Task 2:** Investigate the Manifest File

**Questions**:

* + **Permissions:** What permissions are declared in the manifest file? List them. Are any permissions excessive or unnecessary? Explanation must be provided. Please include a screenshot for the permission list. **(maximum 5 marks)**
  + **Exported components**: Check for exported activities, is the ***android:exported*** attribute set to true or false for each activity? Briefly explain the reason for setting true or false for each activity. What are the names of the exported activities? **(maximum 8 marks)**
  + **Provider**: identify the ***provider****,* what is the purpose of the ***android:exported*** attribute for this provider. What security implications could arise from improper configuration? (**maximum 10 marks)**

**Task 3:** Locate the hardcoded Information. Investigate ***resources.arsc*** and review the contents of the **res/values** directory.

* **Questions**:
  1. Review the ***strings.xml*** file. Are there any hardcoded credentials (e.g., API keys, usernames, passwords)? If you find any, please list them and explain potential risks associated with their presence. Map the risk with the OWASP top 10 Mobile Risks and identify mitigation mechanism to address these risks. **(maximum 10 marks)**
  2. Search the codebase for hardcoded URLs or IP addresses. Check whether any HTTP URLs are used instead of HTTPS. If so, please list them and discuss their implications. What security implications could arise from using HTTP based on the **OWASP top 10 Mobile** Risks. **(maximum 7 marks)**

**Task 4:** Explore the source code to locate ***FlagNine***. Specifically search for ***FirebaseActivity***, are there any base64 encoded value present in the ***FirebaseActivity***? If so, decode the encoded value and provide details, e.g., provide the steps. Additionally, explain whether this value presents a potential risk. **(maximum 10 marks)**

**Task 5: Automated Analysis using MobSF** **tool (Maximum 20 marks)**

1. Install the MobSF tool and use it to perform a static analysis of your APK file
2. Generate a static analysis report for your APK file and include a copy of your report with this coursework submission.
3. Review the MobSF report to identify any security risks or vulnerabilities in the APK.
4. If there is any security risks, map them to the OWASP Mobile Top 10 Risks.
5. Present your MobSF tool and report during one of the labs to your instructor.

Resources:

* **MobSF Github:** <https://github.com/MobSF/Mobile-Security-Framework-MobSF>
* MobSF Documentation: <https://mobsf.github.io/docs/#/>
* **MobSF Dependencies:**<https://mobsf.github.io/docs/#/requirements>
* **To generate PDF Reports:**<https://wkhtmltopdf.org/downloads.html>

End of Lab 😊