SIL771: SPECIAL MODULE IN CYBER SECURITY

Assignment No. 2: SECURITY TESTING OF ANDROGOAT MOBILE APP

Introduction: We have learnt Android app security, the Android Security Model and performed security testing of DIVA app. To reinforce the learning, you have to perform security testing of AndroGoat mobile app as assignment 2. This assignment carries 100 marks and is due on 25 Apr 24.

Task: Download AndroGoat.apk from the zip file at https://mobisec.in/frida.zip. Create an Android Device in Genymotion with Google Pixel 3 device with Android 6.0 API 23 (you may use other configurations as well). Install AndroGoat.apk and verify that it runs. You may use tools such as adb (https://developer.android.com/tools/releases/platform-tools), jadx-gui (https://sourceforge.net/projects/jadx.mirror/), mobSF (https://mobsf.live/), apktool (https://apktool.org/), DB Browser for SQLite (https://sqlitebrowser.org/dl/), Burp-suite (https://portswigger.net/burp/documentation/desktop/getting-started/download-and-install), Frida (https://frida.re/docs/home/), Objections (https://github.com/sensepost/objection), etc. to perform vulnerability analysis and penetration testing of the app.

Your task is to perform the security testing of the given app and exploit the vulnerabilities (built intentionally in the app). The list of vulnerabilities covered in the app are listed below.

1. Network intercepting – HTTP
2. Network intercepting – HTTPS
3. Network intercepting – Certificate Pinning
4. Unprotected Android Components – Activity
5. Unprotected Android Components –Service
6. Unprotected Android Components – Broadcast Receivers
7. Unprotected Android Components – Custom URL Scheme
8. Insecure Data Storage – Shared Prefs - 1
9. Insecure Data Storage - Shared Prefs - 2
10. Insecure Data Storage - SQLite
11. Insecure Data Storage – Temp Files
12. Insecure Data Storage – SD Card
13. Input Validations – XSS
14. Input Validations – SQLi
15. Input Validations – WebView
16. Keyboard Cache
17. Insecure Logging
18. Clipboard - Copy & paste
19. Hard coding issues
20. Root Detection
21. Emulator Detection

22. Binary Patching

Identify and exploit 20 vulnerabilities out of 22 listed above. You will earn **5 points** for successfully solving the challenge by identifying/ exploiting each vulnerability of the app.

Submission: Prepare a report clearly defining the following for each vulnerability you identify/ exploit:

- Vulnerability (What is the vulnerability?)
- Root Cause of the Vulnerability
- Snapshot of the commands you executed
- Snapshot of the Screens of relevant parts of the app to show success

Submit a report in pdf format having your roll no and name as filename (e.g. A1_2021JCS2290_AmitSingh). The submission is **due on 25 Apr 24 2359 Hrs**.

Note: The assignment has to be solved individually and relevant snapshots must be attached to demonstrate the results you obtained.