TOOLS

* adb
* Dex2jar,dex2smali,back2smali
* Jadx
* Apktool
* Android Studio emulator
* Burp pro/community
* Strings
* Burp certificate
* Objection
* MobSF
* Frida (verifica versione client cosi da prendere Frida server della stessa versione. Attenzione all’emulatore in runnerà Frida server). Runnare prima Frida-server poi objection

Per Frida e objection:

python -m pip install objection

python -m pip install frida-tools

d2j 2.0 o version inferiori con le nuove app può dare problemi. Viene quindi usata d2j 2.1 che li resolve C:\Users\crimensi\Desktop\PERSEUS\mobile\_tools\mobile\dex2jar-2.1\dex-tools-2.1\d2j-dex2jar2.1.bat

Usare python versione 3.7

MobSF

Da wsl in Windows:

* sudo apt-get install git
* sudo apt-get install python3.
* sudo apt-get install openjdk-8-jdk
* sudo apt install python3-dev python3-venv python3-pip build-essential libffi-dev libssl-dev libxml2-dev libxslt1-dev libjpeg8-dev zlib1g-dev wkhtmltopdf (se non funziona sudo apt install libjpeg62-turbo-dev)
* sudo apt install wkhtmltopdf
* sudo apt install python3-pip
* sudo apt install python3.9-venv
* git clone <https://github.com/MobSF/Mobile-Security-Framework-MobSF.git>
* cd Mobile-Security-Framework-MobSF
* sudo chmod +x setup.sh
* sudo ./setup.sh (ci vuole del tempo)
* sudo chmod +x run.sh
* ./run.sh

TESTING VULNS

* Componenti esportate (Activity,Service,Broadcast Receiver, Content Provider)
* Analisi del traffico (http o https) simile ai test sulle web application/web services
* Client-Side vulnerabilities/Server-side vulnerabilities
* Analisi statica del codice di default tramite JADX
* Insecure Data Storage (usando emulatore Android 8 x64 bit)
* Insecure Logging
* Copy/Paste Issues
* Client-Side Protection (Obfuscation,Root Detection, SSL Pinning, Detecting Debuggers, End-to-end encryption)

Client-Side Protection (SSL Pinning Bypass)

frida -U -f <package\_name> -l sslpinningbypass.js --no-paus

(il certificate di burp deve essere già presente dentro il device)

<https://infosecwriteups.com/hail-frida-the-universal-ssl-pinning-bypass-for-android-e9e1d733d29>

Tramite SAST si cerca il metodo della classe che fa il controllo della root detection. Si crea un customize script in JS per frida per modificare il valore di ritorno

Client\_Side Protection (Root Detection)

Si cerca il metodo che controlla la root detection. Se booleano ad esempio (in diversi casi è cosi) basta modificare a runtime il valore di ritorno

1. frida -U -f <package\_name>
2. objection –gadget <package\_name> explore -s
3. Ritornando su frida scrivere %resume

Client\_Side Protection (End-To-end encryption)

Usare code review e runtime tracing per ottenere le chiavi di encryption