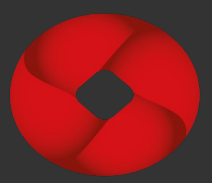
# Pentesting Android Apps

Sneha Rajguru (@Sneharajguru)



n|u

# About Me

- Penetration Tester
- Web, Mobile and Infrastructure applications,
- Secure coding ( part time do secure code analysis),
- CTF challenge writer (at HackIM- Nullcon & Winja),
- A wannabe guitarist !
- ...

# What are we talking about ...

- Pentesting Environment setup
- Arsenal
- App Analysis
- With #Respect to the Developers!!

# Pentesting Environment setup

- So, you have a choice to select from
  - A. Physical Device
  - B. Android SDK Emulator
  - C. Android x86 (VM).

# Pentesting Environment setup

- If a physical device is used, remember “root” is needed.
- Also, check “ allow from unknown sources”
- You may install Cydia substrate;
- Also, AndroidSSLTrustKiller by iSEC Partners
- Install any proxy app / or setup the connection with Burp proxy, fiddler or *any of your choice..*

# Arsenal

dex2jar



JD-GUI

Smali/Baksmali

keytool

Introspy

zipalign

SignAPK

jarsigner

drozer


# Arsenal

- Android SDK
  - Software Development Kit containing api libraries and developer tools to build, test and debug Android apps.
  - Well what we need the most are :

    adb,aapt,ddms and the emulator.

http://developer.android.com/sdk/

# Arsenal
# adb

- Command-line tool to communicate with emulator instance or connected physical/virtual device.
  - Most needed commands :
    - adb connect
    - adb devices
    - adb install
    - adb push
    - adb pull
    - adb shell

# DDMS
# Dalvik debug monitor server

- Debugging tool that provides port-forwarding, screen capture, heap dump, logcat, file manager and many other features.

http://developer.android.com/sdk/

# Arsenal

## dex2jar

- Converts from dex to smali or dex2jar- an approximate representation of the original source code.

# Arsenal
# jd-gui

- standalone graphical utility that displays Java source codes of “.class” files.

# Smali/Baksmali

- Assembler/disassembler for the Dex format used by Dalvik.

# Arsenal
## Introspy

- Tool to analyze app behavior during runtime and help to identify potential security issues.
- Tool to Generate HTML reports based on the database generated by Introspy-Android.

https://github.com/iSECPartners/Introspy-Android

- Tool to bypass SSL certificate pinning for most applications *
- To get this install ; Cydia substrate + AndroidSSLTrustKiller
- Well this can be done manually as well!!

# Certificates and validating the pinnig

- Proxy server CA certificate
  - Make use of burp, generate a host machine certificate

# Arsenal
# Burp suite

- Integrated platform for security testing of web applications.
- The most interesting part is to generate the certificate and intercept and inspect the requests and responses between the app and its backend...inshort uncovering the treasure

# Arsenal


- Security testing framework, great to determine app attack surface and interact with it.

https://www.mwrinfosecurity.com/products/drozer/

# App Analysis
# - The apk components

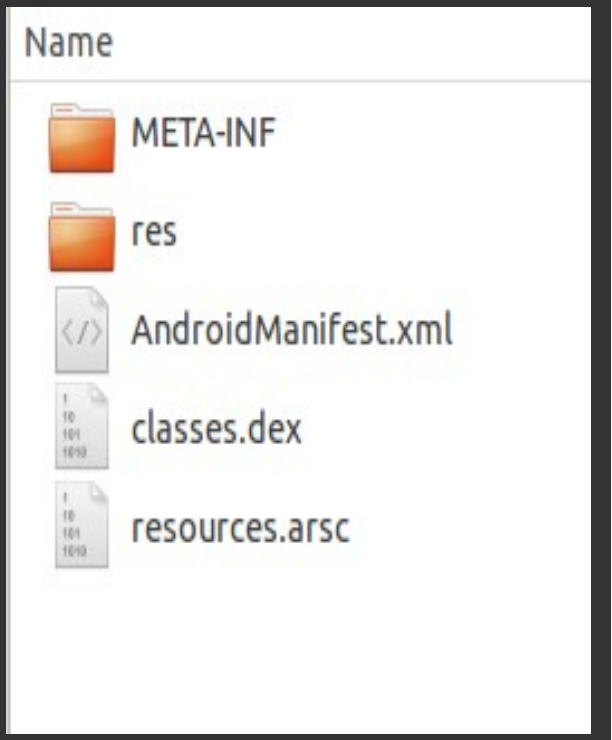
- **Activity:** represents a single screen with a user interface.
- **Service:** No user interface, runs in background.
- **Content provider:** manages a shared set of application data. Eg access google contacts.
- **Broadcast receiver:** responds to system-wide broadcast announcements. Many broadcasts originate from the system.
- I**ntent filter:** messaging object used to request an action from another app component, describes the activity to start and carries any necessary data.
- **AndroidManifest.xml**
  - names the Java package for the application (unique identifier)
  - describes the components of the application
  - declares which permissions the application must have in order to access protected parts of the API and interact with other applications.
  - declares the permissions that others are required to have in order to interact with the application's components.
  - the minimum level of the Android API.

# Anatomy of android application

- An extended .jar file ..which is converted to a simple zip file and then renamed as .apk (extension)
- App resource
- Signatures
- Manifest (The binary XML)
  - Then comes binary code, dalvik compilation,

    more binaries...

**Lets target !!**




**Can be found at  /data/app**

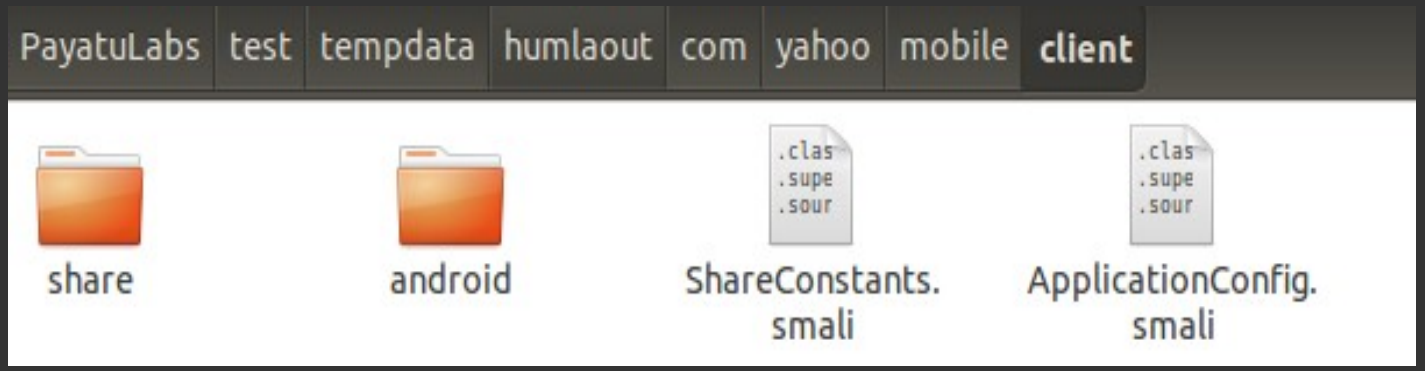
# Stick to the basics...

First thing first :

- Rename the <apk-name>.apk to <apk-name>.zip  & then decompress it to any folder.
- “classes.dex”
  - This contains the compiled vm codes
- Lets disassemble it!!
  - Lets get “Baksmali” to work!

- Unzip yahoo.apk classes.dex
- java -jar baksmali-2.0.6.jar ./classes.dex -o humlaout
- We get a path

  humlaout/com/yahoo/mobile/client....




  - Open and check  all the .smali files

# AndroidManifest.xml

- Provides information about the app to the system.
- Defines the app permissions
- Defines the app components

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.basiccontactables"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <!-- Min/target SDK versions (<uses-sdk>) managed by build.gradle -->
    <permission android:name="android"></permission>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/Theme.Sample" >
        <activity
            android:name="com.example.android.basiccontactables.MainActivity"
            android:label="@string/app_name"
            android:launchMode="singleTop">
            <meta-data
                android:name="android.app.searchable"
                android:resource="@xml/searchable" />
            <intent-filter>
                <action android:name="android.intent.action.SEARCH" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# App Analysis
# - shared prefs

- XML format file with key-value pairs.
- App settings.

# App Analysis
# -sqLite database

- Single file relational database used to store application data and settings.

# For the Developers #Respect

- Thank you so much for making an attackers life so easy!!!

# For the developers

- Insecure Data Storage -

  Shared Preferences without **MODE_WORLD_READABLE.**
- Sensitive information should not be stored.
  - If needed, should be encrypted from derivation of user Password/PIN and not with hardcoded encryption keys.
  - Still vulnerable to offline brute-force. Enforce strong password policy.

- InsufficientTransport Layer Protection
  - Apply SSL/TLS transport in channels that the app transmits sensitive information to the backend.
  - Implement Certificate Pinning if very sensitive information is transmitted.

- Client Side Injection
  - Only export components(Activities,Services, Broadcast Receivers,Content Providers) that make sense and that cannot bypass access controls and leak Internal information.

- Lack of Binary Protection
  - Obfuscate your code, at minimum with ProGuard.