

4

Session

Pentesting Android Applications

Thực hành Bảo mật web và ứng dụng



Lưu hành nội bộ 2023

(Nghiêm cấm đăng tải trên internet dưới mọi hình thức)
A. TỔNG QUAN

A.1 Mục tiêu

Giúp sinh viên có kiến thức và kỹ năng cơ bản trong kiểm thử một ứng dụng Android đơn giản.

B. CHUẨN BỊ MÔI TRƯỜNG

B.1.1 Các tập tin được cung cấp sẵn

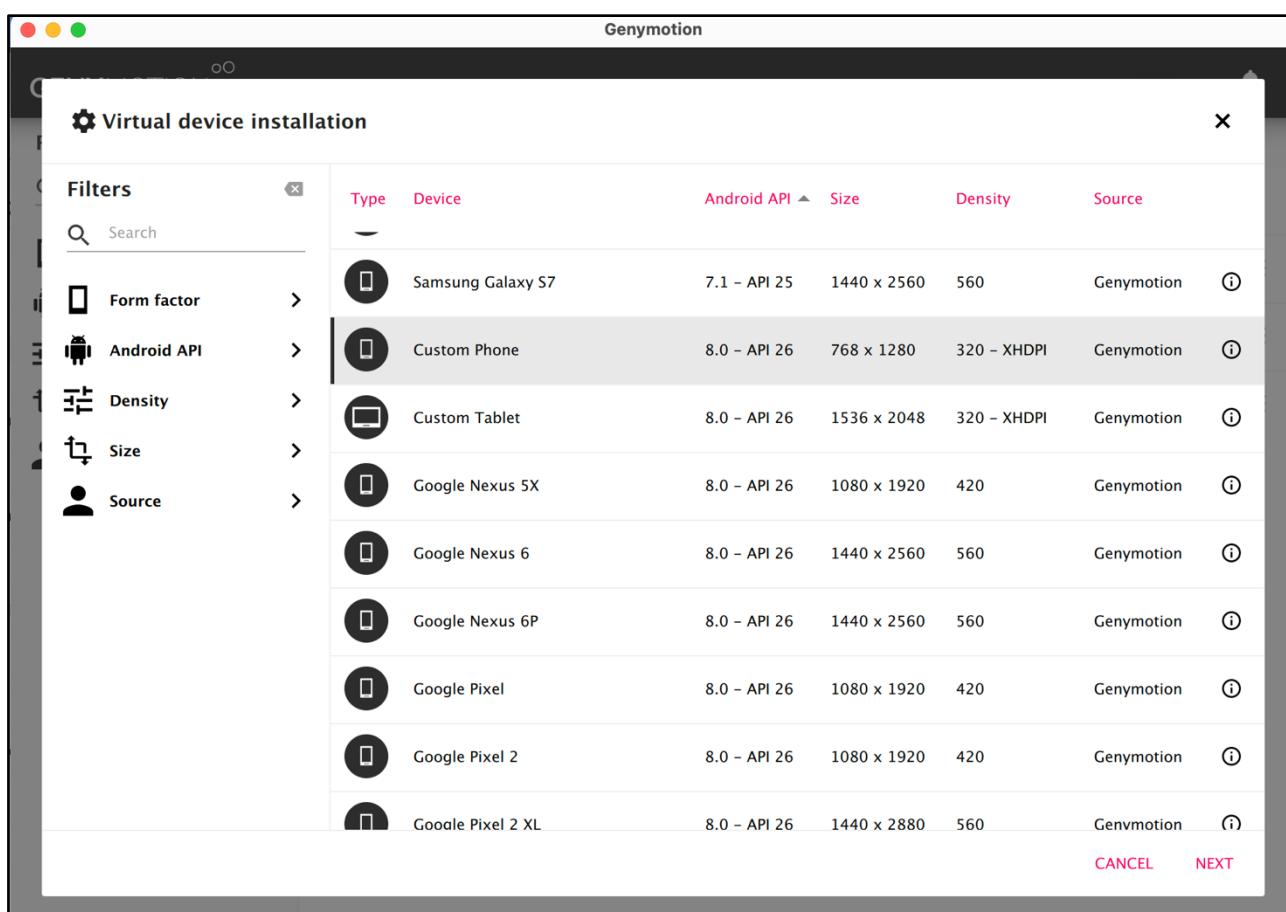
Với mỗi yêu cầu bên dưới, sinh viên tiến hành phân tích một số file APK được giảng viên cung cấp sẵn, bao gồm:

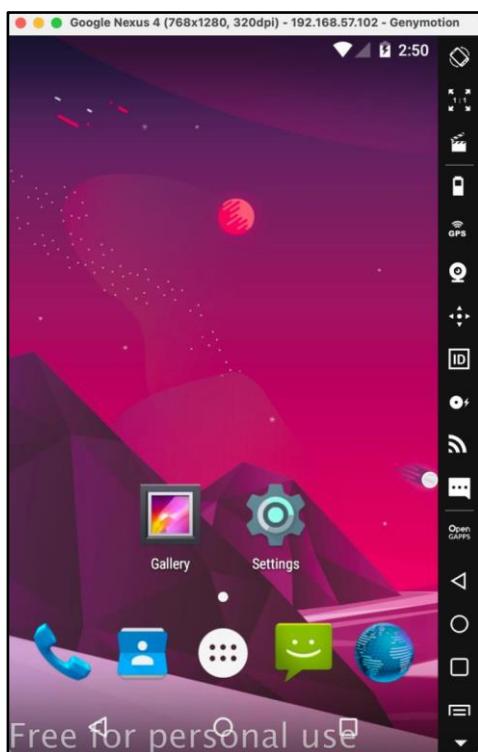
- **InsecureBankv2.apk:** file ứng dụng InsecureBank đã có tùy chỉnh, dùng cho yêu cầu phân tích file APK.
- **AndroLabServer:** thư mục chứa các file Python dùng để chạy server web cho ứng dụng, dùng python3.

B.1.2 Điện thoại ảo

Ta cần hai phần mềm:

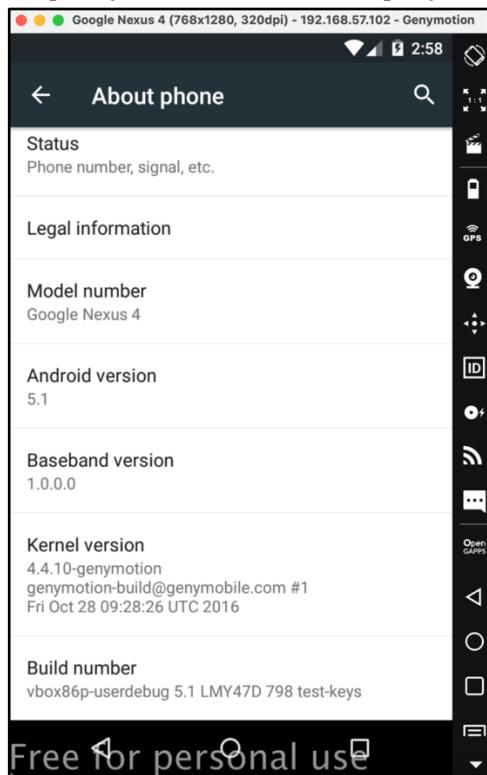
- Virtualbox (<https://www.virtualbox.org/wiki/Downloads>): để chạy Genymotion.
- Genymotion (<https://www.genymotion.com/download/>): Virtualbox dùng làm nền cho Genymotion chạy máy ảo.
- Dùng Genymotion tạo điện thoại ảo “Custom Phone – 8.0 – API 26” (tùy thích).





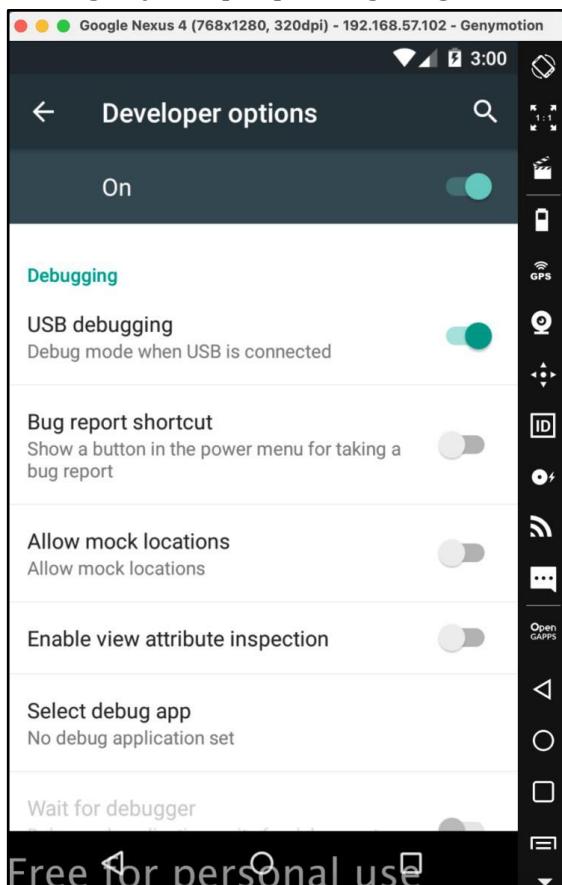
B.1.3 Công cụ Android

- Cài đặt Android Debug Bridge (ADB):
 - Ubuntu/Kali (Linux): `sudo apt install android-tools-adb`
 - MacOS: `brew install android-platform-tools`
 - Windows: theo hướng dẫn [tại đây](#)
- Trên điện thoại ảo vừa tạo, chọn Settings > About Phone > Bấm 6 đến 12 lần vào Build number để bật chế độ developer (You are now developer)



Session 4: Android Pentesting

- Về lại phần Settings, sẽ thấy mục Developer options xuất hiện. Chọn Developer options > USB Debugging lên, chức năng này cho phép tương tác giữa điện thoại ảo và máy thật.



- Bây giờ có thể cài apk bằng adb

```
ADB [~] ~$ ~/Desktop/Android
ADB [~] ~$ adb devices
ADB [~] ~$ List of devices attached
ADB [~] ~$ 192.168.57.102:5555      device
```

192.168.57.102 là địa chỉ IP của điện thoại ảo và sẽ tương tác từ máy thật qua port 5555.

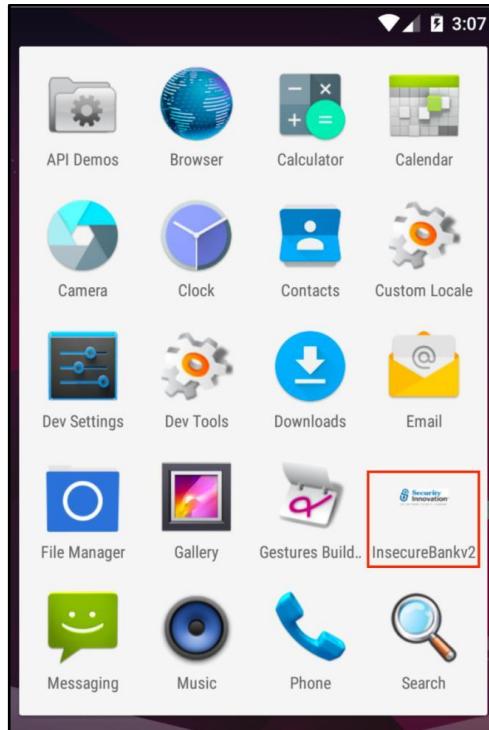
- Cài đặt apk vào điện thoại bằng câu lệnh.

```
adb install InsecureBankv2.apk
```

```
ADB [~] ~$ ~/Desktop/Android
ADB [~] ~$ adb install InsecureBankv2.apk
ADB [~] ~$ Performing Push Install
ADB [~] ~$ InsecureBankv2.apk: 1 file pushed, 0 skipped. 26.2 MB/s (3462429 bytes in 0.126s)
ADB [~] ~$ pkg: /data/local/tmp/InsecureBankv2.apk
ADB [~] ~$ Success
```

Session 4: Android Pentesting

- Xem trên điện thoại, thấy tập tin đã cài đặt thành công



- Để vào terminal shell của điện thoại, gõ lệnh:

```
adb shell
```

```
ADB shell
[adb shell]
root@vbox86p:/ # whoami
root
root@vbox86p:/ # uname -a
Linux localhost 4.4.10-genymotion #1 SMP PREEMPT Fri Oct 28 09:28:26 UTC 2016 x86_64 GNU/Linux
root@vbox86p:/ #
```

- Để xem log, gõ lệnh:

```
adb logcat
V/BackupManagerService( 728): restoreAtInstall pkg=com.android.insecurebankv2 token=3 restoreSet=0
V/BackupManagerService( 728): Finishing install immediately
W/Settings( 728): Setting install_non_market_apps has moved from android.provider.Settings.Global to
                    android.provider.Settings.Secure, returning read-only value.
I/InputReader( 728): Reconfiguring input devices. changes=0x00000010
D/BackupManagerService( 728): Received broadcast Intent { act=android.intent.action.PACKAGE_ADDED dat
=package:com.android.insecurebankv2 flg=0x4000010 (has extras) }
I/ActivityManager( 728): Start proc 2406:com.svox.pico/u0a49 for broadcast com.svox.pico/.VoiceDataIn
stallerReceiver
W/VoiceInteractionManagerService( 728): no available voice recognition services found for user 0
E/libprocessgroup( 2406): failed to make and chown /acct/uid_10049: Read-only file system
W/Zygote ( 2406): createProcessGroup failed, kernel missing CONFIG_CGROUP_CPUACCT?
I/art ( 728): Explicit concurrent mark sweep GC freed 24189(1531KB) AllocSpace objects, 10(241KB)
LOS objects, 33% free, 7MB/11MB, paused 1.948ms total 68.344ms
D/Launcher.Workspace( 1622): 11683562 - removeExtraEmptyScreen()
D/Launcher.Workspace( 1622): 11683562 - convertFinalScreenToEmptyScreenIfNecessary()
I/art ( 2383): System.exit called, status: 0
I/AndroidRuntime( 2383): VM exiting with result code 0.
I/ActivityManager( 728): Killing 1497:android.process.media/u0a5 (adj 15): empty #17
W/libprocessgroup( 728): failed to open /acct/uid_10005/pid_1497/cgroup.procs: No such file or direct
ory
I/InputReader( 728): Reconfiguring input devices. changes=0x00000010
D/BackupManagerService( 728): Received broadcast Intent { act=android.intent.action.PACKAGE_CHANGED d
at=package:com.android.musicfx flg=0x4000010 (has extras) }
W/Launcher( 1622): setApplicationContext called twice! old=com.android.launcher3.LauncherApplication@2
2c5d2af new=com.android.launcher3.LauncherApplication@22c5d2af
I/art ( 1374): Background sticky concurrent mark sweep GC freed 5942(397KB) AllocSpace objects, 0(
0B) LOS objects, 19% free, 2MB/2MB, paused 16.120ms total 24.745ms
I/UsageStatsService( 728): User[0] Flushing usage stats to disk
```

- Để tải lên một tập tin từ máy thật đến điện thoại ảo, gõ lệnh:

```
adb push <path máy thật> <path máy ảo>
```

```

└─[?] ~/Desktop/Android
    echo "test" > test

└─[?] ~/Desktop/Android
    adb push test /tmp/
    test: 1 file pushed, 0 skipped. 0.0 MB/s (5 bytes in 0.003s)

└─[?] ~/Desktop/Android
    adb shell
    root@vbox86p:/ # cat /tmp/test
    test
    root@vbox86p:/ # 
```

- Để tải một tập tin từ điện thoại ảo về máy thật, gõ lệnh:

```
root@vbox86p:/ # echo "device" > /tmp/device
root@vbox86p:/ # exit
```

```

└─[?] ~/Desktop/Android
    adb pull /tmp/device ./
    /tmp/device: 1 file pulled, 0 skipped. 0.0 MB/s (7 bytes in 0.002s)

└─[?] ~/Desktop/Android
    cat device
    device
```

C. THỰC HÀNH

C.1 Phân tích tĩnh (Static Analysis)

Phân tích tĩnh là quá trình rà soát mã nguồn, cấu hình để xem xét lỗ hổng. Song song với việc làm thủ công có thể kết hợp với công cụ tự động để giúp kết quả đạt độ chính xác cao hơn.

Mobile Security Framework (MobSF) là công cụ kiểm thử di động tự động (Android/iOS/Windows), phân tích mã độc, hỗ trợ phân tích tĩnh lẫn phân tích động. MobSF hỗ trợ tập tin binaries (APK, IPA & APPX).



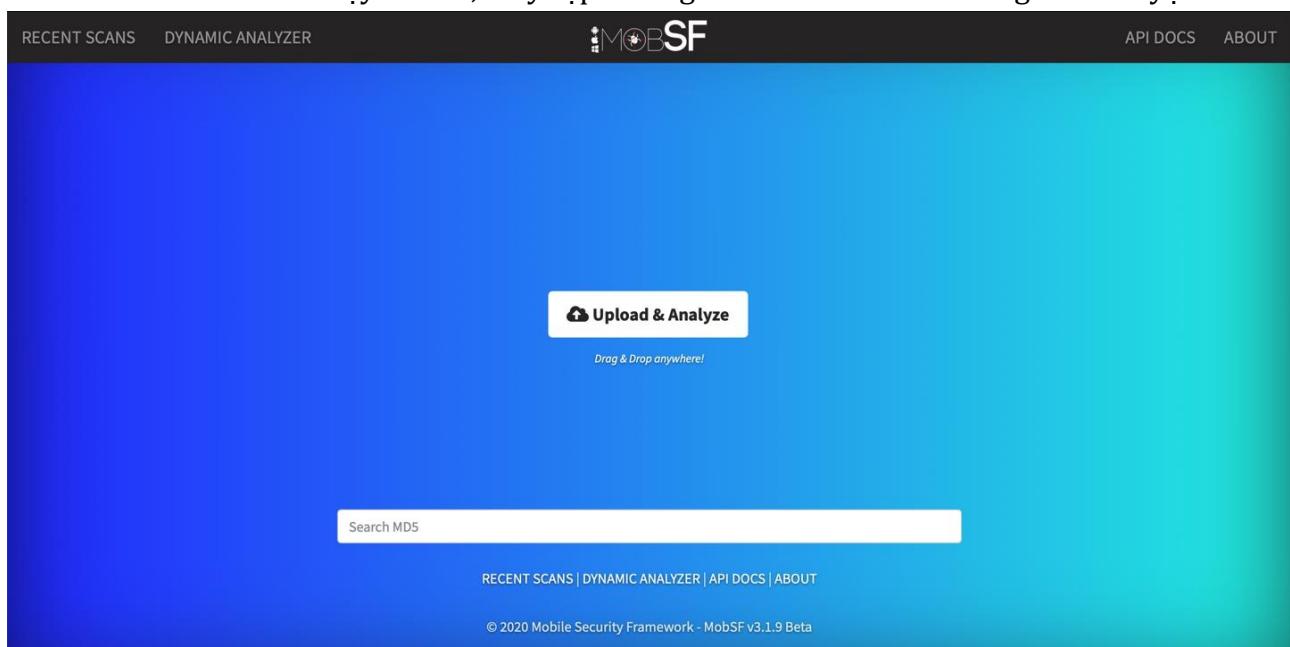
C.1.1 Cài đặt

Bước 1. Xem chuẩn bị môi trường và cài đặt tại: <https://mobsf.github.io/docs/#/>.

Bước 2. Bật MobSF bằng lệnh:

```
./run.sh
```

Bước 3. Sau khi chạy server, truy cập đường dẫn *localhost:8000* bằng trình duyệt.





C.1.2 Phân tích tự động

Tải lên hoặc kéo thả tập tin *InsecureBankv2.apk* MobSF để tiến hành phân tích tĩnh.

The screenshot shows the MobSF static analysis interface. On the left, there's a sidebar with various analysis options like Information, Scan Options, Signer Certificate, Permissions, and Components. The main area displays the following information:

- APP SCORES:** Average CVSS 0, Security Score 100/100, Trackers Detection 3/319.
- FILE INFORMATION:** File Name: InsecureBankv2.apk, Size: 3.3MB, MD5: 5ee4829065640f9c936ac861d1650ff, SHA1: 80b53f80a3c9e6bdf98311f5b26ccddcd1bf0a98, SHA256: b18af2a0e44d7634bbcd93664d9c78a2695e050393fcfb5e8b91f902d194a4.
- APP INFORMATION:** App Name: InsecureBankv2, Package Name: com.android.insecurebankv2, Main Activity: com.android.insecurebankv2.LoginActivity, Target SDK: 22, Min SDK: 15, Max SDK: 1, Android Version Name: 1.0, Android Version Code: 1.
- Key Metrics:**
 - ACTIVITIES: 10 (View)
 - SERVICES: 0 (View)
 - RECEIVERS: 2 (View)
 - PROVIDERS: 1 (View)
 - Exported Activities: 4
 - Exported Services: 0
 - Exported Receivers: 1
 - Exported Providers: 1
- SCAN OPTIONS:** Scan now.
- DECOMPILED CODE:** View AndroidManifest.xml, View Sources, View Smali.

Kiểm tra các cấu hình phân quyền

The screenshot shows the MobSF static analysis interface with the "Permissions" option selected in the sidebar. The main area displays a table of application permissions:

| PERMISSION | STATUS | INFO | DESCRIPTION |
|---|-----------|---------------------------------------|---|
| android.permission.ACCESS_COARSE_LOCATION | dangerous | coarse (network-based) location | Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are. |
| android.permission.ACCESS_NETWORK_STATE | normal | view network status | Allows an application to view the status of all networks. |
| android.permission.GET_ACCOUNTS | dangerous | list accounts | Allows access to the list of accounts in the Accounts Service. |
| android.permission.INTERNET | normal | full Internet access | Allows an application to create network sockets. |
| android.permission.READ_CONTACTS | dangerous | read contact data | Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people. |
| android.permission.READ_PROFILE | dangerous | read the user's personal profile data | Allows an application to read the user's personal profile data. |
| android.permission.SEND_SMS | dangerous | send SMS | Allows application to send SMS messages. Malicious applications may |

Session 4: Android Pentesting

Kiểm tra cấu hình trong tập tin Manifest

MANIFEST ANALYSIS

Search:

| NO ↑↓ | ISSUE | SEVERITY ↑↓ | DESCRIPTION |
|-------|---|-------------|--|
| 1 | Debug Enabled For App [android:debuggable=true] | high | Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes. |
| 2 | Application Data can be Backed up [android:allowBackup=true] | medium | This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device. |
| 3 | Activity (com.android.insecurebankv2.PostLogin) is not Protected. [android:exported=true] | high | An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. |
| 4 | Activity (com.android.insecurebankv2.DoTransfer) is not Protected. [android:exported=true] | high | An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. |
| 5 | Activity (com.android.insecurebankv2.ViewStatement) is not Protected. [android:exported=true] | high | An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. |
| 6 | Content Provider (com.android.insecurebankv2.TrackUserContentProvider) | high | A Content Provider is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. |

MobSF cung cấp thông tin cần thiết để đánh giá độ bảo mật của chương trình

SIGNER CERTIFICATE

APK is signed
v1 signature: True
v2 signature: False
v3 signature: False
Found 1 unique certificates
Subject: ST=MA, L=Boston, O=SI, OU=Services, CN=Dinesh Shetty
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2015-07-24 20:37:08+00:00
Valid To: 2040-07-17 20:37:08+00:00
Issuer: ST=MA, L=Boston, O=SI, OU=Services, CN=Dinesh Shetty
Serial Number: 0x6bb4f616
Hash Algorithm: sha256
md5: 6a736d89abb13d7165e7cff905ac928d
sha1: a1bae91a2b1620f6c9dab425e69fc32ba1e97741
sha256: 8092db81ae717486631a1534977def465ee112903e1553d38d41df8abd57a375
sha512: 53770f3f69916f74ddd6e750ae16fd9b23fa5b2c8e9e53bd5a84202d7d7c44a26ede13e6db450ab0c1d9f64534802b88ebb0b4de1da076b62112d9b12
2cbbd92

Search:

| STATUS ↑↓ | DESCRIPTION |
|-----------|---|
| bad | Application is signed with v1 signature scheme, making it vulnerable to Janus vulnerability on Android <7.0 |
| secure | Application is signed with a code signing certificate |

A STRINGS

```

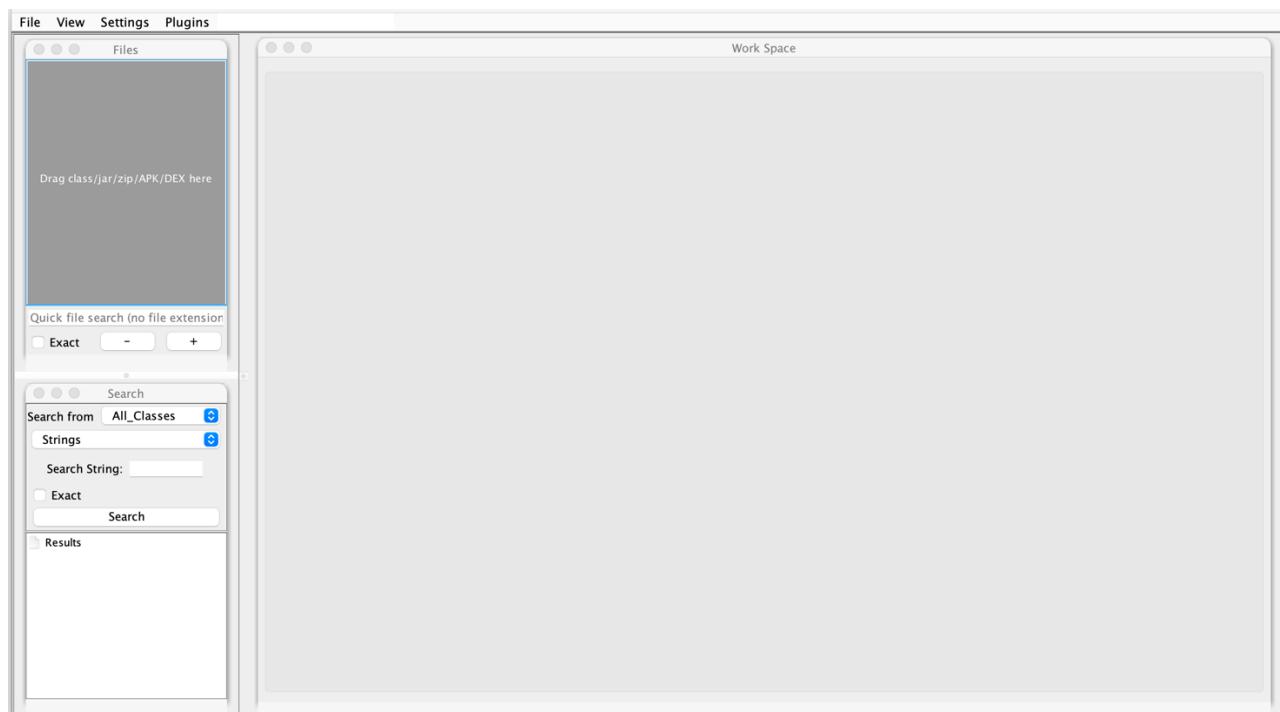
"abc_action_bar_up_description": "اوپر نویگیت کرین"
"common_google_play_services_update_title": "Opdater Google Play-tjenester"
"common_android_wear_update_title": "Eguneratu Android Wear"
"mr_media_route_controller_stop": "Заустави пребацивање"
"common_google_play_services_api_unavailable_text": "%1$s memerlukan satu atau lebih perkhidmatan Google Play yang tidak tersedia pada masa ini. Sila hubungi pembangun untuk mendapatkan bantuan."
"common_google_play_services_unknown_issue": "ມີບັນຫາທີ່ບໍ່ຄວດສິດໃນບໍລິການ Google Play."
"common_google_play_services_update_text": "Lai lietotne %1$s darbotos, jums ir jāatjaunina Google Play pakalpojumi."
"abc_action_bar_up_description": "Navigate up"
"mr_user_route_category_name": "Уреди"
"common_google_play_services_install_text_phone": "您的手机没有安装 Google Play 服务，因此无法运行%1$s。"
"common_google_play_services_update_title": "Ažuriraj usluge za Google Play"
"abc_searchview_description_submit": "Indsend forespørgslen"
"common_google_play_services_needs_enabling_title": "ഒക്ക് അപുവർത്തനാനികി Google Play സ്വല്ല പ്രാരംഭിച്ചുഡാലെ."
"abc_action_mode_done": "Kész"
"abc_activity_chooser_view_see_all": "ഏല്ലാം നോക്കു"
"common_android_wear_update_text": "%1$s no se ejecutará hasta que no actualices la aplicación Android Wear."
"abc_action_bar_home_description": "ת'בבה ר'ל טוינ"
"mr_media_route_chooser_searching": "ଉପକରଣଙ୍କେ ଖର୍ଚ୍ଚକାରୀ ତିରଯୁଗ୍ମୁ..."
"mr_media_route_controller_stop": "Lõpeta ülekanne"
"common_google_play_services_sign_in_failed_title": "സാിന ഇൻ അസഫൽ ഭയോ"
"common_google_play_services_unknown_issue": "Google പ്രିସବାର ସଙ୍ଗେ ଅଜାନୀ ସମସ୍ୟା।"
"common_google_play_services_enable_title": "Activati Servicii Google Play"
"mr_media_route_controller_pause": "Դադար"
"mr_media_route_controller_play": "Phát"

```

C.1.3 Phân tích thủ công

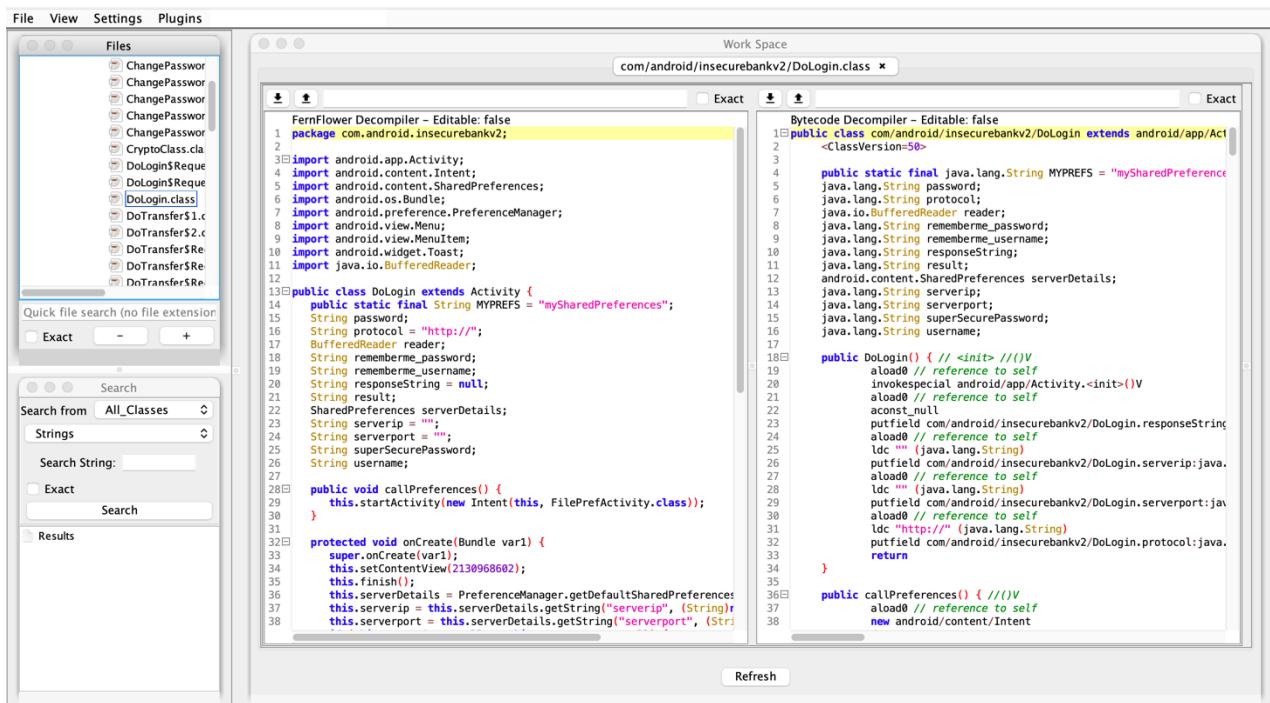
Một trong các bước quan trọng trong phân tích tĩnh là kiểm tra code. Để làm được việc này dùng ByteCode Viewer để xem code nhanh nha trong tập tin apk. Tải tập tin **Bytecode-Viewer-2.9.x.jar** từ <https://github.com/Konloch/byticode-viewer/releases> và mở chương trình bằng dòng lệnh sau:

```
java -jar Bytecode-Viewer-2.9.x.jar
```



Kéo thả tập tin apk, ByteCode Viewer sẽ phân tích và hiển thị code.

Session 4: Android Pentesting



Sử dụng *Malicious Code Scanner* ở mục *Plugins* để quét các đoạn code có vẻ nguy hiểm ☺



Ví dụ: Nhìn thấy đoạn code lạ ở đường dẫn:
`com/android/insecurebanky2/DoLogin$RequestTask.class`

Session 4: Android Pentesting

```

105     protected void onProgressUpdate(Integer... var1) {
106
107
108
109     public void postData(String var1) throws ClientProtocolException, IOException, Invalid
110     DefaultHttpClient var2 = new DefaultHttpClient();
111     HttpPost var3 = new HttpPost(this.this$.protocol + this.this$.serverip + ":" + this.this$.serve
112     HttpPost var5 = new HttpPost(this.this$.protocol + this.this$.serverip + ":" + this.this$.serve
113     ArrayList var4 = new ArrayList(2);
114     var4.add(new BasicNameValuePair("username", this.this$.username));
115     var4.add(new BasicNameValuePair("password", this.this$.password));
116     HttpResponse var6;
117     if (this.this$.username.equals("devadmin")) {
118         var5.setEntity(new UrlEncodedFormEntity(var4));
119         var6 = var2.execute(var5);
120     } else {
121         var3.setEntity(new UrlEncodedFormEntity(var4));
122         var6 = var2.execute(var3);
123     }
124
125     InputStream var7 = var6.getEntity().getContent();
126     this.this$.result = this.convertStreamToString(var7);
127     this.this$.result = this.this$.result.replace("\n", "");
128     if (this.this$.result != null) {
129         Intent var8;
130         if (this.this$.result.indexOf("Correct Credentials") != -1) {
131             Log.d("Successful Login:", " account=" + this.this$.username + ":" + this.this$.password)
132             this.saveCreds(this.this$.username, this.this$.password);
133             this.trackUserLogins();
134             var8 = new Intent(this.this$.getApplicationContext(), PostLogin.class);
135             var8.putExtra("uname", this.this$.username);
136             this.this$.startActivity(var8);
137     } else {
138         var8 = new Intent(this.this$.getApplicationContext(), WrongLogin.class);
139         this.this$.startActivity(var8);
140     }
141
142

```

Yêu cầu 1 Phân tích và chỉ ra điểm bất thường của đoạn code trên?

Thông tin đăng nhập không được mã hoá trong cơ sở dữ liệu

Cài đặt chương trình *InsecureBankv2* lên máy ảo Android, thử đăng nhập để chương trình lưu thông tin (**dinesh/Dinesh@123\$** hoặc **jack/Jack@123\$**).

Từ máy chính, gõ **adb shell** để vào command line của máy ảo Android.

```

generic_x86_arm:/data/data/com.android.insecurebankv2/databases # ls
mydb  mydb-journal
generic_x86_arm:/data/data/com.android.insecurebankv2/databases # sqlite3 mydb
SQLite version 3.28.0 2020-05-06 18:46:38
Enter ".help" for usage hints.
sqlite> 
```

Yêu cầu 2 Chỉ ra rằng dữ liệu lưu trữ có an toàn hay không?

Thông tin nhạy cảm lưu dưới dạng plain-text

1. Sử dụng chương trình
2. Vào *adb shell* đi đến */data/data/com.android.insecurebankv2*
3. Chạy câu trúc lệnh sau:

```
grep -r <string-to-find> $(find)
```

Yêu cầu 3 Kiểm tra xem thông tin nhạy cảm có lưu lại trên thiết bị hay không? Một số từ khóa: deviceId, userId, imei, deviceSerialNumber, devicePrint, phone, XDSN, mdn, IMSI, uuid...

Tập tin sao lưu không được mã hoá

Nếu app cho phép backup, ta sẽ thử tính năng này xem chương trình ứng dụng lưu những gì.

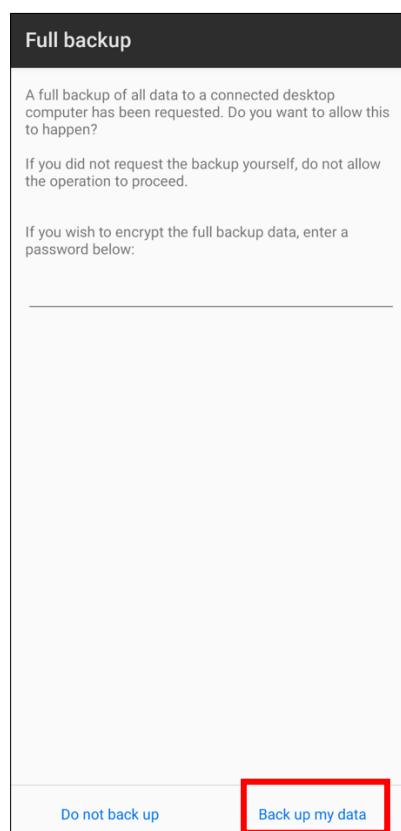
```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-feature android:glEsVersion="20000" android:required="true"/>
<application android:theme="@style/Theme.Holo.Light.DarkActionBar" android:label="@string/app_name"
  android:allowBackup="true">
    <activity android:label="@string/app_name" android:name="com.android.insecurebankv2.LoginActivit
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <activity android:label="@string/title_activity_file_pref" android:name="com.android.insecurebar
      de="adjustUnspecified|stateVisible|adjustResize|adjustPan|adjustNothing"/>
```

1. Đăng nhập với tài khoản người dùng bình thường (**dinesh/Dinesh@123\$** hoặc **jack/Jack@123\$**)
2. Thực thi lệnh sau để sao lưu:

```
adb backup -apk -shared com.android.insecurebankv2
```

```
> adb backup -apk -shared com.android.insecurebankv2
WARNING: adb backup is deprecated and may be removed in a future release
Now unlock your device and confirm the backup operation...
```

3. Tại giao diện điện thoại tiến hành cho phép backup



4. Sao lưu được tạo

```
> ls | grep backup.ab
backup.ab
```

5. Chuyển đổi tập tin sao lưu qua định dạng có thể đọc được (cài gói qpdf)

```
cat backup.ab | (dd bs=24 count=0 skip=1; cat) | zlib-flate -uncompress >
backup_compressed.tar
```

```
> cat backup.ab | (dd bs=24 count=0 skip=1; cat) | zlib-flate -uncompress > backup_compressed.tar
0+0 records in
0+0 records out
0 bytes transferred in 0.000014 secs (0 bytes/sec)
> tar -zxf backup_compressed.tar
x apps/com.android.insecurebankv2/_manifest
x apps/com.android.insecurebankv2/a/base.apk
x apps/com.android.insecurebankv2/db/mydb-journal
x apps/com.android.insecurebankv2/db/mydb
x apps/com.android.insecurebankv2/sp/mySharedPreferences.xml
x apps/com.android.insecurebankv2/sp/com.android.insecurebankv2_preferences.xml
x shared/0/Documents
x shared/0/Audiobooks
x shared/0/DCIM
x shared/0/Download
x shared/0/Movies
x shared/0/Movies/.thumbnails
x shared/0/Movies/.thumbnails/.nomedia
x shared/0/Movies/.thumbnails/.database_uuid
x shared/0/Pictures
x shared/0/Pictures/.thumbnails
x shared/0/Pictures/.thumbnails/.nomedia
x shared/0/Pictures/.thumbnails/.database_uuid
x shared/0/Notifications
x shared/0/Alarms
```

6. Kiểm tra các tập tin sao lưu sau khi giải nén

```
> cd apps/com.android.insecurebankv2/sp
> ls -la
total 16
drwxr-xr-x  4 phaphajian  staff  128 Nov 24 13:39 .
drwxr-xr-x  6 phaphajian  staff  192 Nov 24 13:39 ..
-rw-r-----  1 phaphajian  staff  160 Nov 23 16:09 com.android.insecurebankv2_preferences.xml
-rw-r-----  1 phaphajian  staff  221 Nov 23 16:10 mySharedPreferences.xml
> cat mySharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="superSecurePassword">v/sJpihDCo2ckDmLW5Uwiw==
;      </string>
    <string name="EncryptedUsername">amFjaw==
;      </string>
</map>
```

Yêu cầu 4 Theo bạn thư mục sao lưu chứa thông tin nào cần mã hoá, chỉ ra.

Phương thức mã hoá yếu

Tại đường dẫn `com/android/insecurebankv2/CryptoClass.class`, chứa các hàm mã hoá và giải mã.

```

14
15 public class CryptoClass {
16     String base64Text;
17     byte[] cipherData;
18     String cipherText;
19     byte[] ivBytes = new byte[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
20     String key = "This is the super secret key 123";
21     String plaintext;
22
23     public static byte[] aes256decrypt(byte[] var0, byte[] var1, byte[] var2) throws UnsupportedEncodingException, NoSuchAlgorithmException {
24         IvParameterSpec var4 = new IvParameterSpec(var0);
25         SecretKeySpec var5 = new SecretKeySpec(var1, "AES");
26         Cipher var3 = Cipher.getInstance("AES/CBC/PKCS5Padding");
27         var3.init(2, var5, var4);
28         return var3.doFinal(var2);
29     }
30
31     public static byte[] aes256encrypt(byte[] var0, byte[] var1, byte[] var2) throws UnsupportedEncodingException, NoSuchAlgorithmException {
32         IvParameterSpec var4 = new IvParameterSpec(var0);
33         SecretKeySpec var3 = new SecretKeySpec(var1, "AES");
34         Cipher var5 = Cipher.getInstance("AES/CBC/PKCS5Padding");
35         var5.init(1, var3, var4);
36         return var5.doFinal(var2);
37     }
38
39     public String aesDecryptedString(String var1) throws UnsupportedEncodingException, InvalidKeyException, NoSuchAlgorithmException {
40         byte[] var2 = this.key.getBytes("UTF-8");
41         this.cipherData = aes256decrypt(this.ivBytes, var2, Base64.decode(var1.getBytes("UTF-8"), 0));
}

```

- Ta có key: "This is the super secret key 123"
- Ta có đoạn dữ liệu bị mã hoá

```

> cat mySharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="superSecurePassword">v/sJpihDCo2ckDmLW5Uiw===&#10;      </string>
    <string name="EncryptedUsername">amFjaw===&#13;&#10;      </string>
</map>

```

Yêu cầu 5 Viết chương trình giải mã đoạn dữ liệu mã hoá (python3 chẵng hạn...)

```

[~/Desktop/Android] python3 -m pip install pycrypto
Defaulting to user installation because normal site-packages is not writeable
Collecting pycrypto
  Using cached pycrypto-2.6.1.tar.gz (446 kB)
Building wheels for collected packages: pycrypto
  Building wheel for pycrypto (setup.py) ... done
  Created wheel for pycrypto: filename=pycrypto-2.6.1-cp38-cp38-macosx_10_14_6_x86_64.whl size=606112 sha256=976b39567eac9886338a359c556bfeae422f83ebf4bb7dd9937dff786e5c8f9a
  Stored in directory: /Users/phaphajian/Library/Caches/pip/wheels/d0/99/d0/0298ea019d63f1d63a0965b9944b719e875f9bd6fffc6dcf293
Successfully built pycrypto
Installing collected packages: pycrypto
Successfully installed pycrypto-2.6.1

[~/Desktop/Android] python3 decrypt.py
Jack@123$ 

```

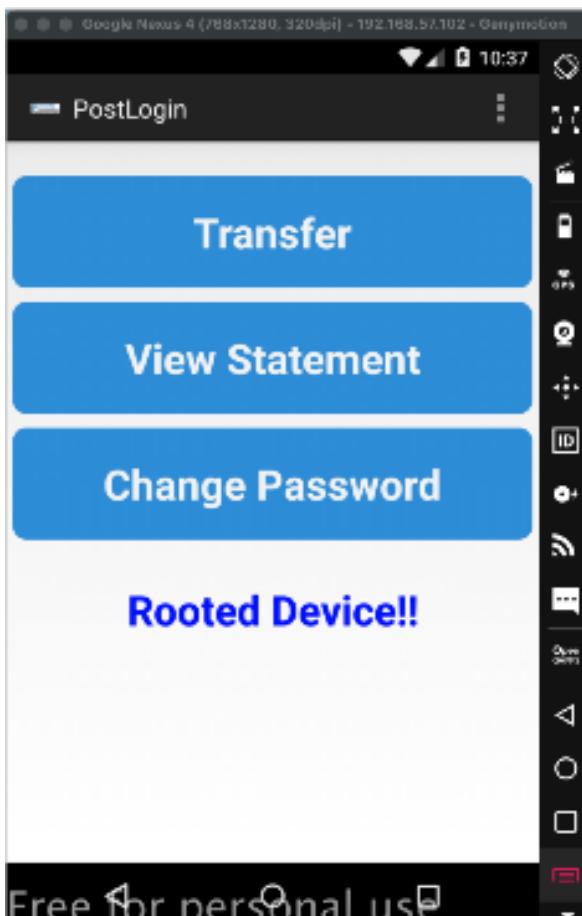
Activity Hijacking

- Gọi Activity Manager (am): là lệnh gọi các lệnh của hệ thống; ví dụ chạy Activity, dừng một tiến trình, sửa đổi các thuộc tính...
- Thủ gọi activity PostLogin, dùng lệnh sau:

```
am start -n com.android.insecurebankv2/.PostLogin
```

```
➜ ~/Desktop/Android
└─ adb shell
root@vbox86p:/ # am start -n com.android.insecurebankv2/.PostLogin
Starting: Intent { cmp=com.android.insecurebankv2/.PostLogin }
root@vbox86p:/ #
```

App tự động chuyển đến PostLogin mà không cần đăng nhập.



C.2 Dịch ngược (Reverse Engineering)

C.2.1 Bên trong APK

- APK (Android application package) là 1 tập tin chứa các thành phần cần thiết cho việc cài đặt ứng dụng trên thiết bị Android.
- Giống như Windows có tập tin .exe Linux có tập tin .elf, thì Android cũng có tập apk, quá trình cài đặt tương tự.
- Việc cài đặt thủ công (cài bằng tập tin apk) được gọi là side-loading.
- Dưới đây là các thành phần thường có trong tập tin apk:
 - META-INF/: Thư mục này có ở trong tập apk đã ký, nó bao gồm các danh sách tập tin trong APK và chữ kí của chúng.

- lib/: Các file của thư viện native (file có đuôi *.so) được lưu vào thư mục con của lib/ như x86, x86_64.
- res/: thư mục này gồm tất cả các resource XML, drawables (PNG, JPEG) với các loại khác nhau từ mdpi, hdpi, sw600dp... cho tới resource ngôn ngữ.
- AndroidManifest.xml: Mô tả tên, phiên bản, và nội dung của tập tin apk.
- **classes.dex**: Bao gồm mã nguồn đã biên dịch, được chuyển dưới dạng Dex bytecode
- resources.arsc: Một số tài nguyên và định nghĩa được biên dịch và lưu trữ ở đây. Các resource này lưu vào apk mà không cần nén để truy cập nhanh hơn lúc runtime.
- Apk là một tập tin nén (zip format-type) được dựa trên JAR file format với đuôi .apk là phần mở rộng. Uzip tập tin apk:

```
phaphajian@PhaPha-Jians-MacBook-Pro:~/Desktop/Android ~%1
└─[unzip InsecureBankv2.apk
Archive: InsecureBankv2.apk
inflating: AndroidManifest.xml
inflating: res/anim/abc_fade_in.xml
inflating: res/anim/abc_fade_out.xml
inflating: res/anim/abc_grow_fade_in_from_bottom.xml
inflating: res/anim/abc_popup_enter.xml
inflating: res/anim/abc_popup_exit.xml
inflating: res/anim/abc_shrink_fade_out_from_bottom.xml
inflating: res/anim/abc_slide_in_bottom.xml
inflating: res/anim/abc_slide_in_top.xml
inflating: res/anim/abc_slide_out_bottom.xml
inflating: res/anim/abc_slide_out_top.xml
inflating: res/color-v11/abc_background_cache_hint_selector_material_dark.xml
inflating: res/color-v11/abc_background_cache_hint_selector_material_light.xml
```

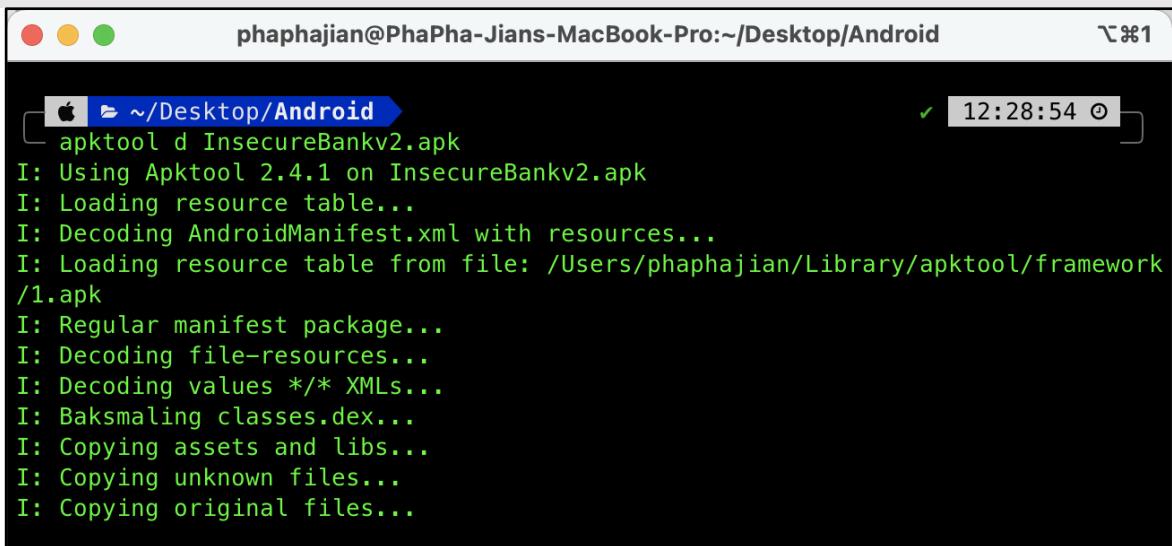
- Các thành phần nói trên đã xuất hiện

```
ls -al
total 12872
drwxr-xr-x@ 7 phaphajian staff      224 Apr 24 12:21 .
drwxr-xr-x@ 25 phaphajian staff      800 Apr 24 12:21 ..
-rw-r--r--@ 1 phaphajian staff     7588 Sep 18 2015 AndroidManifest.xml
drwxr-xr-x@ 5 phaphajian staff      160 Apr 24 12:21 META-INF
-rw-r--r--@ 1 phaphajian staff  6125692 Sep 18 2015 classes.dex
drwxr-xr-x@ 27 phaphajian staff     864 Apr 24 12:21 res
-rw-r--r--@ 1 phaphajian staff   454300 Jul 26 2015 resources.arsc
```

- ⇒ Phần này chưa gọi là dịch ngược vì nó còn ở dạng binary.
- ⇒ Deomplie: phiên dịch ngược mã nguồn.
- apktool được sử dụng để decomplie (Chi tiết cài đặt [tại đây](#))
- apktool giúp decomplie tập AndroidManifest.xml về tập tin .xml nguyên bản, cũng như resources.arsc và classses.dex thành một dạng ngôn ngữ gọi là SMALI.

- Thực hiện lệnh sau:

```
apktool d InsecureBankv2.apk
```

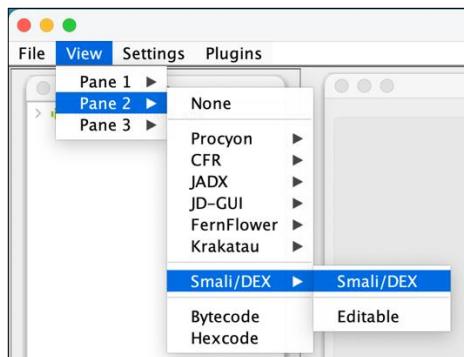


```
phaphajian@PhaPha-Jians-MacBook-Pro:~/Desktop/Android
apktool d InsecureBankv2.apk
I: Using Apktool 2.4.1 on InsecureBankv2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/phaphajian/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

- Bước kế tiếp ta có thể lấy được mã nguồn bằng tool dex2jar. Sau đó đọc nội dung các tập tin jar bằng công cụ jd-gui.
- ⇒ Nhằm tự động hoá các quá trình trên ta dùng Bytecode Viewer.

C.2.2 Smali

Thêm tập tin apk vào Bytecode Viewer.



Mở tập tin *InsecureBankv2/com/android/insecurebankv2/DoLogin\$RequestTask.class*, ta sẽ thấy smali code ở tab/panel thứ 2

Session 4: Android Pentesting

```

Work Space
com/android/insecurebankv2/ChangePassword$RequestCha... x com/android/insecurebankv2/DoLogin$RequestTask.class x
Exact
FernFlower Decompiler - Editable: false
1 package com.android.insecurebankv2;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences.Editor;
5 import android.os.AsyncTask;
6 import android.util.Base64;
7 import android.util.Log;
8 import com.android.insecurebankv2.DoLogin.RequestTask.1;
9 import java.io.BufferedReader;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import java.io.InputStreamReader;
13 import java.io.UnsupportedEncodingException;
14 import java.security.InvalidAlgorithmParameterException;
15 import java.security.InvalidKeyException;
16 import java.security.NoSuchAlgorithmException;
17 import java.util.ArrayList;
18 import javax.crypto.BadPaddingException;
19 import javax.crypto.IllegalBlockSizeException;
20 import javax.crypto.NoSuchPaddingException;
21 import org.apache.http.HttpResponse;
22 import org.apache.http.client.ClientProtocolException;
23 import org.apache.http.client.entity.UrlEncodedFormEntity;
24 import org.apache.http.client.methods.HttpPost;
25 import org.apache.http.impl.client.DefaultHttpClient;
26 import org.apache.http.message.BasicNameValuePair;
27 import org.json.JSONException;
28
29 class DoLogin$RequestTask extends AsyncTask<String, String, String> {
30     // $FF: synthetic field
31     final DoLogin this$0;
32
33     DoLogin$RequestTask(DoLogin var1) {
34         this.this$0 = var1;
35     }
36
37     private String convertStreamToString(InputStream var1) throws
38     try {
        ...
    }

```

```

Smali Decompiler - Editable: false
1 .class Lcom/android/insecurebankv2/DoLogin$RequestTask;
2 .super Landroid/os/AsyncTask;
3
4 .annotation system Ldalvik/annotation/EnclosingClass;
5     value = Lcom/android/insecurebankv2/DoLogin;
6 .end annotation
7 .annotation system Ldalvik/annotation/InnerClass;
8     accessFlags = 0
9     name = "RequestTask"
10    .end annotation
11   .annotation system Ldalvik/annotation/Signature;
12     value = {
13         "Landroid/os/AsyncTask",
14         "<",
15         "Ljava/lang/String;",
16         "Ljava/lang/String;",
17         "Ljava/lang/String;",
18         ">,"
19     }
20 .end annotation
21
22 .field final synthetic this$0:Lcom/android/insecurebankv2/DoLogin;
23
24 .method constructor <init>(<Lcom/android/insecurebankv2/DoLogin;)
25     .registers 2
26     iput-object p1, p0, Lcom/android/insecurebankv2/DoLogin$RequestTask;
27     invoke-direct { p0 }, Landroid/os/AsyncTask;.><init>()
28     return void
29 .end method
30
31 .method private convertStreamToString(Ljava/io/InputStream;)Ljava/lang/String;
32     .annotation system Ldalvik/annotation/Throws;
33     value = {
34         Ljava/io/IOException;
35     }
36 .end annotation
37     .catch Ljava/io/UnsupportedEncodingException; { :L0 .. :L1
38     .registers 6

```

Phần trên có đoạn “devadmin”, xem thử smali thì nó hình thù như thế nào.

```

117 if (this.this$0.username.equals("devadmin")) {
118     var5.setEntity(new UrlEncodedFormEntity(var4));
119     var6 = var2.execute(var5);
120 } else {
121     var3.setEntity(new UrlEncodedFormEntity(var4));
122     var6 = var2.execute(var3);
123 }

```

```

283 const_string v5, "devadmin"
284 invoke-virtual { v4, v5 }, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
285 move-result v4
286 if-eqz v4, :L2

```

- Nó sẽ lấy chuỗi ta nhập vào (v4) và so sánh chuỗi “devadmin” (v5) bằng cách gọi method java.lang.String.equals(), nếu đúng trả về 1 sai trả về 0.
- Sau đó lưu kết quả vào v4.
- Tiếp tục chương trình gọi câu lệnh if-eqz (nếu bằng 0 – if equals zero), nghĩa là nếu chuỗi nhập vào không thỏa mãn sẽ nhảy đến phần code ở :L2.

Session 4: Android Pentesting

- Xem qua đoạn code tiếp theo nếu if-eqz thoả mãn và không thoả mãn.
 - o Là “devadmin”

if-eqz v4, :L2

```

new-instance v1, Lorg/apache/http/client/entity/UrlEncodedFormEntity;

invoke-direct { v1, v3 }, Lorg/apache/http/client/entity/UrlEncodedFormEntity;-><init>(Ljava/util/List;)V

invoke-virtual { v2, v1 }, Lorg/apache/http/client/methods/HttpPost;->setEntity(Lorg/apache/http/HttpEntity;)V

invoke-interface { v0, v2 }, Lorg/apache/http/client/HttpClient;-
>execute(Lorg/apache/http/client/methods/HttpRequest;)Lorg/apache/http/HttpResponse;

move-result-object v0

```

:L0

Đoạn java tương đương

```

var5.setEntity(new UrlEncodedFormEntity(var4));
var6 = var2.execute(var5)

```

⇒ v2 là gì?

```

const-string v4, "/devlogin"
invoke-virtual { v3, v4 }, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;

move-result-object v3

invoke-virtual { v3 }, Ljava/lang/StringBuilder;->toString()Ljava/lang/String;

move-result-object v3

invoke-direct { v2, v3 }, Lorg/apache/http/client/methods/HttpPost;-><init>(Ljava/lang/String;)V

```

Đoạn code java tương đương

```

HttpPost var5 = new HttpPost(this.this$0.protocol + this.this$0.serverip + ":" + this.this$0.serverport
+ "/devlogin");

```

⇒ Kết luận đoạn này nếu chúng ta nhập “devadmin”, thì nó truy vấn HTTP POST đến http://ip_server/devlogin.

- o Không là là “devadmin”

:L2

```

new-instance v2, Lorg/apache/http/client/entity/UrlEncodedFormEntity;

invoke-direct { v2, v3 }, Lorg/apache/http/client/entity/UrlEncodedFormEntity;-><init>(Ljava/util/List;)V

invoke-virtual { v1, v2 }, Lorg/apache/http/client/methods/HttpPost;->setEntity(Lorg/apache/http/HttpEntity;)V

invoke-interface { v0, v1 }, Lorg/apache/http/client/HttpClient;-
>execute(Lorg/apache/http/client/methods/HttpRequest;)Lorg/apache/http/HttpResponse;

move-result-object v0

goto/16 :L0

```

C.2.3 Patching ứng dụng Android

- Patch là tác động và thay đổi code của chương trình, làm cho nó hoạt động theo ý muốn.
- apktool có thể patch ứng dụng.
- Vào thư mục res đã decompile trước đó, Insecurebankv2/res/values/strings.xml

```

73   <string name="create_calendar_message">Allow Ad to create a calendar event?</string>
74   <string name="create_calendar_title">Create calendar event</string>
75   <string name="decline">Decline</string>
76   <string name="hello_world">Hello world!</string>
77   <string name="is_admin">no</string>
78   <string name="loginscreen_password">Password:</string>
79   <string name="loginscreen_username">Username:</string>
80   <string name="pref_submit">Submit:</string>
81   <string name="server_ip">Server IP:</string>
82   <string name="server_port">Server Port:</string>
83   <string name="status_bar_notification_info_overflow">999+</string>

```

- Sửa value "is_admin" bằng yes và cái kết???

```

73   <string name="create_calendar_message">Allow Ad to create a calendar event?</string>
74   <string name="create_calendar_title">Create calendar event</string>
75   <string name="decline">Decline</string>
76   <string name="hello_world">Hello world!</string>
77   <string name="is_admin">yes</string>
78   <string name="loginscreen_password">Password:</string>
79   <string name="loginscreen_username">Username:</string>
80   <string name="pref_submit">Submit:</string>
81   <string name="server_ip">Server IP:</string>
82   <string name="server_port">Server Port:</string>
83   <string name="status_bar_notification_info_overflow">999+</string>

```

- Bây giờ recompile lại tập tin apk

```

apktool b InsecureBankv2 -o InsecureBankv3.apk
I: Using Apktool 2.4.1
I: Checking whether sources has changed...
I: Smaling smalli folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...

```

- Sau đó cài đặt vào điện thoại

```

adb install InsecureBankv3.apk
Performing Streamed Install
adb: failed to install InsecureBankv3.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect certificates from /data/app/vmdl67017203.tmp/base.apk: Attempt to get length of null array]

```

⇒ Failed rồi! Android yêu cầu các tập tin APK đều phải được ký bằng một chứng chỉ trước khi được phép cài đặt trên thiết bị. Sau khi chỉnh sửa, tập tin APK sẽ không còn toàn vẹn như ban đầu nên cần phải được ký lại.

- Tạo Keystore:

Session 4: Android Pentesting

```
keytool -genkey -v -keystore InsecureBankv3.keystore -alias InsecureBankv3 -keyalg RSA -keysize 2048 -validity 10000
keytool -genkey -v -keystore InsecureBankv3.keystore -alias InsecureBankv3 -keyalg RSA -keysize 2048 -validity 10000

Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing InsecureBankv3.keystore]
```

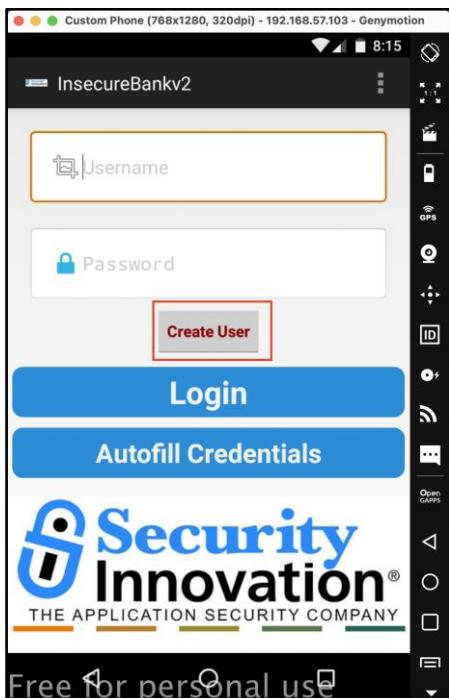
- Sử dụng công cụ **apksigner** hoặc **jarsigner**:

```
apksigner sign --ks InsecureBankv3.keystore InsecureBankv3.apk
```

```
~/Desktop/Android
~/Users/phaphajian/Library/Android/sdk/build-tools/30.0.2/apksigner sign --ks InsecureBankv3.keys
tore InsecureBankv3.apk
Keystore password for signer #1:
```

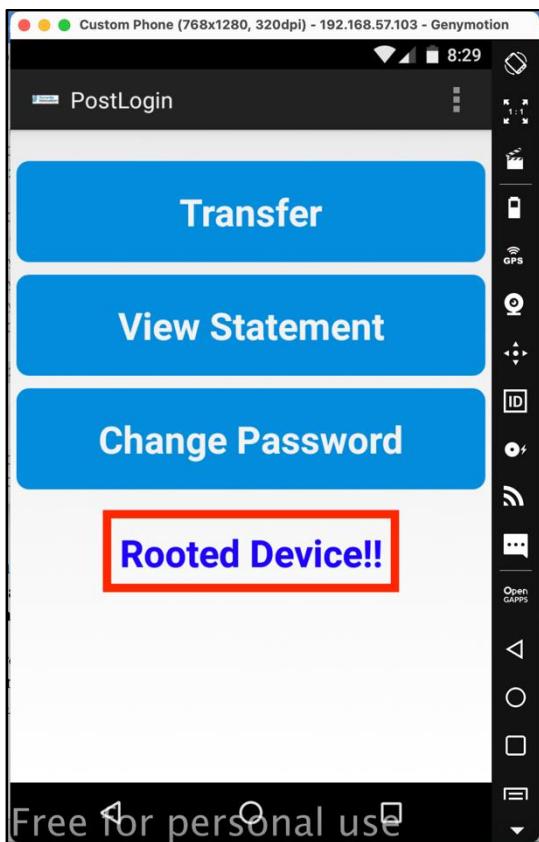
- Tiến hành cài lại và mở ứng dụng

```
~/Desktop/Android
adb install InsecureBankv3.apk
Performing Streamed Install
Success
```



⇒ Nút Create User xuất hiện chứng tỏ chỉ có admin mới có.

- Các chương trình đều có chức năng chặn phân tích như:
 - o Root Detection (anti-root)
 - o Emulator Detection (anti-vm)



- Đoạn check root nằm trong class PostLogin.class

```

80 void showRootStatus() {
81     boolean var1;
82     if (!this.doesSuperuserApkExist("/system/app/Superuser.apk") && !this.doesSUexist()) {
83         var1 = false;
84     } else {
85         var1 = true;
86     }
87
88     if (var1) {
89         this.root_status.setText("Rooted Device!!");
90     } else {
91         this.root_status.setText("Device not Rooted!!");
92     }
93
94 }
```

- Đoạn mã kiểm tra sự tồn tại của "/system/app/Superuser.apk"???
- Để kiểm chứng ta vào shell của điện thoại và tìm kiếm

```
vbox86p:/ # ls -a /system/app/Superuser/Superuser.apk
/system/app/Superuser/Superuser.apk
```

⇒ Kiểm tra sai đường dẫn ⓘ)))))))

Yêu cầu 6 Sinh viên điều chỉnh mã nguồn ứng dụng sao cho luôn hiển thị trạng thái “Rooted Device!!” với bất kỳ trạng thái nào của thiết bị.

Gợi ý:

```

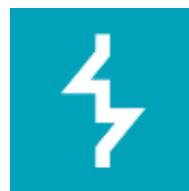
422 .method showRootStatus()V
423     .locals 3
424
425     .prologue
426     const/4 v1, 0x1
427
428     .line 86
429     const-string v2, "/system/app/Superuser.apk"
430
431     invoke-direct {p0, v2}, Lcom/android/insecurebankv2/PostLogin;->doesSuperuserApkExist(Ljava/lang/String;)Z
432
433     move-result v2
434
435     if-nez v2, :cond_0
436
437     .line 87
438     invoke-direct {p0}, Lcom/android/insecurebankv2/PostLogin;->doesSUexist()Z
439
440     move-result v2
441
442     if-eqz v2, :cond_1
443
444     :cond_0
445     move v0, v1
446
447     .line 88
448     .local v0, "isrooted":Z
449     :goto_0
450     if-ne v0, v1, :cond_2
451
452     .line 90
453     ige-object v1, p0, Lcom/android/insecurebankv2/Postlogin;->root_status:Landroid/widget/TextView;
454
455     const-string v2, "Rooted Device!!"
456
457     invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V
458
459     .line 96
460     :goto_1
461     return-void
462
463     .line 87
464     .end local v0    # "isrooted":Z
465     :cond_1
466     const/4 v0, 0x0
467
468     goto :goto_0
469
470     .line 94
471     .restart local v0    # "isrooted":Z
472     :cond_2
473     ige-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
474
475     const-string v2, "Device not Rooted!!"
476
477     invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V
478
479     goto :goto_1
480 .end method

```

if condition

C.3 Phân tích động (Dynamic Analysis)

- Phân tích động là quá trình kiểm thử và đánh giá trong thời gian thực (real time).
- Mục đích của phân tích tĩnh theo là theo dõi biểu hiện trao đổi dữ liệu giữa chương trình và internet mà thay vì ngồi đọc mã nguồn.
- Công cụ hỗ trợ tiến hành phân tích là **Burp Suite**. Burp Suite là một ứng dụng java chạy đa nền tảng (Windows/Linux/Mac) dùng để kiểm thử xâm nhập ứng dụng Web.
- Cài đặt tại: <https://portswigger.net/burp/communitydownload>



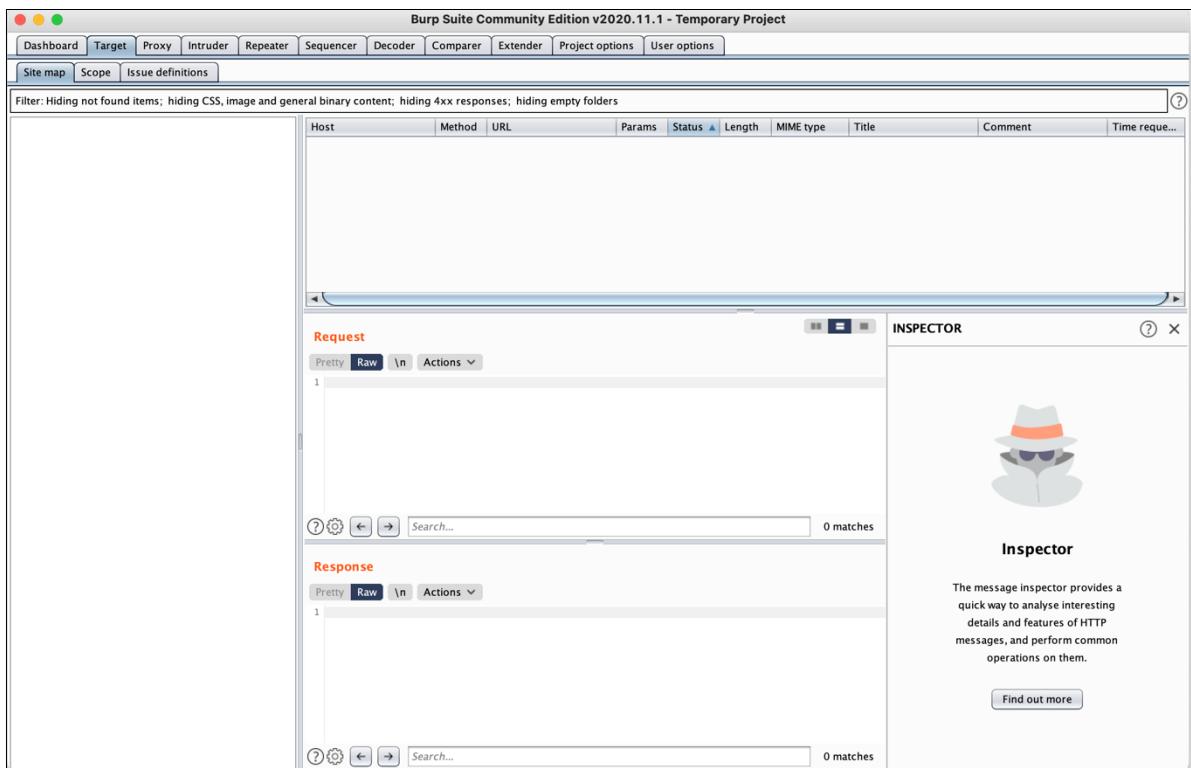
C.3.1 Cài đặt

Cấu hình cho điện thoại ảo “proxy” thông qua Burp Suite, điều này có nghĩa là mọi yêu cầu (request) được gửi từ điện thoại đều được Burp Suite bắt lại, rồi mới đi đến nơi cần đến (server).

- Kiểm tra địa chỉ ip của điện thoại ảo

```
> adb devices
List of devices attached
192.168.56.103:5555    device
```

- Mở Burp Suite



- Chọn Tab *Proxy > Options*

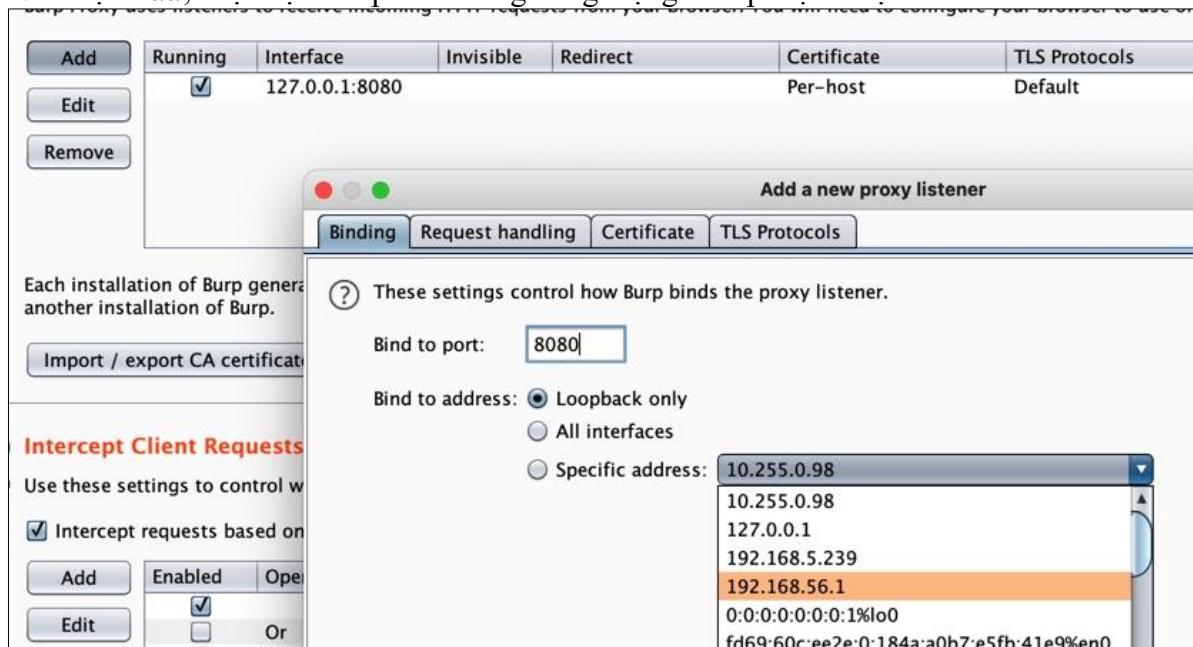
| Add | Running | Interface | Invisible | Redirect | Certificate | TLS Protocols |
|-----------------------------------|----------------|-----------|-----------|----------|-------------|---------------|
| <input checked="" type="button"/> | 127.0.0.1:8080 | | | | Per-host | Default |

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate.

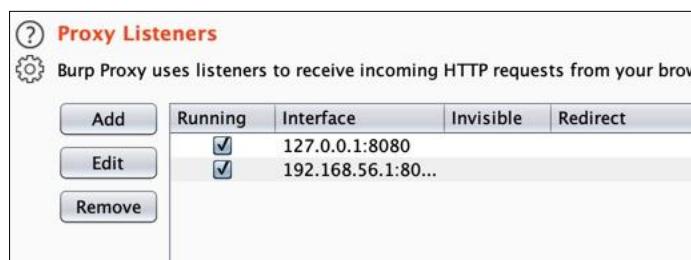
Import / export CA certificate Regenerate CA certificate

Session 4: Android Pentesting

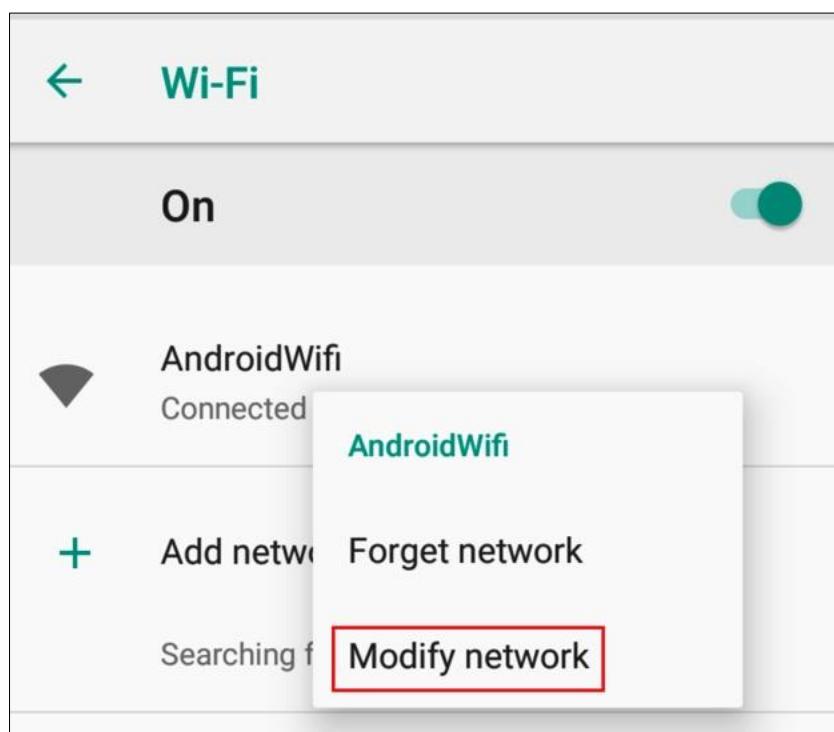
4. Chọn *Add*, chọn địa chỉ ip nằm trong vùng mạng của ip điện thoại ảo



5. *Proxy Listeners* đã xuất hiện interface vừa tạo, chọn tick phần *Running*.

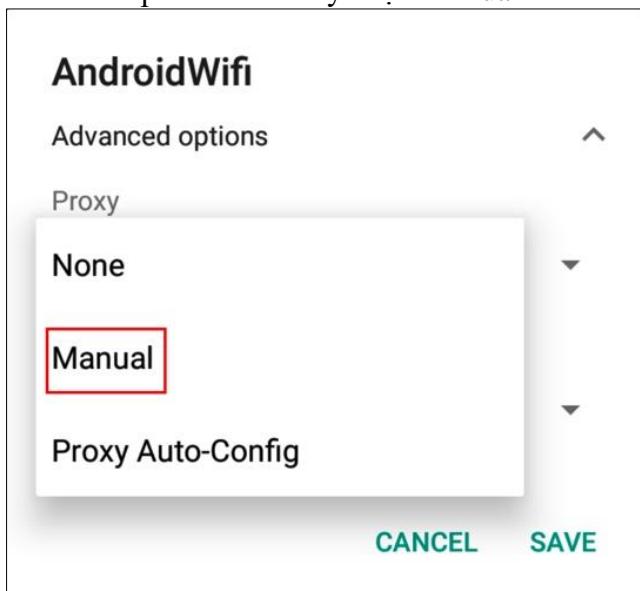


6. Ở điện thoại ảo vào *Settings > Network & Internet > WiFi*, click giữ vào trường *tên SSID*. Chọn *Modify network*

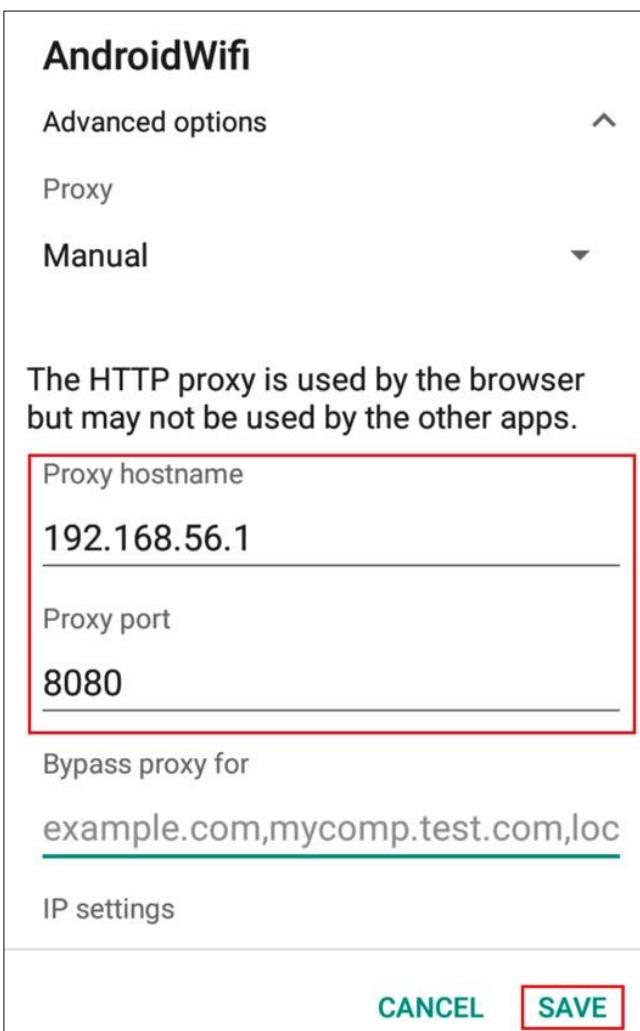


Session 4: Android Pentesting

7. Chọn xổ xuống Advanced options. Ở Proxy chọn Manual



8. Điền thông tin proxy đã thiết lập bên trên



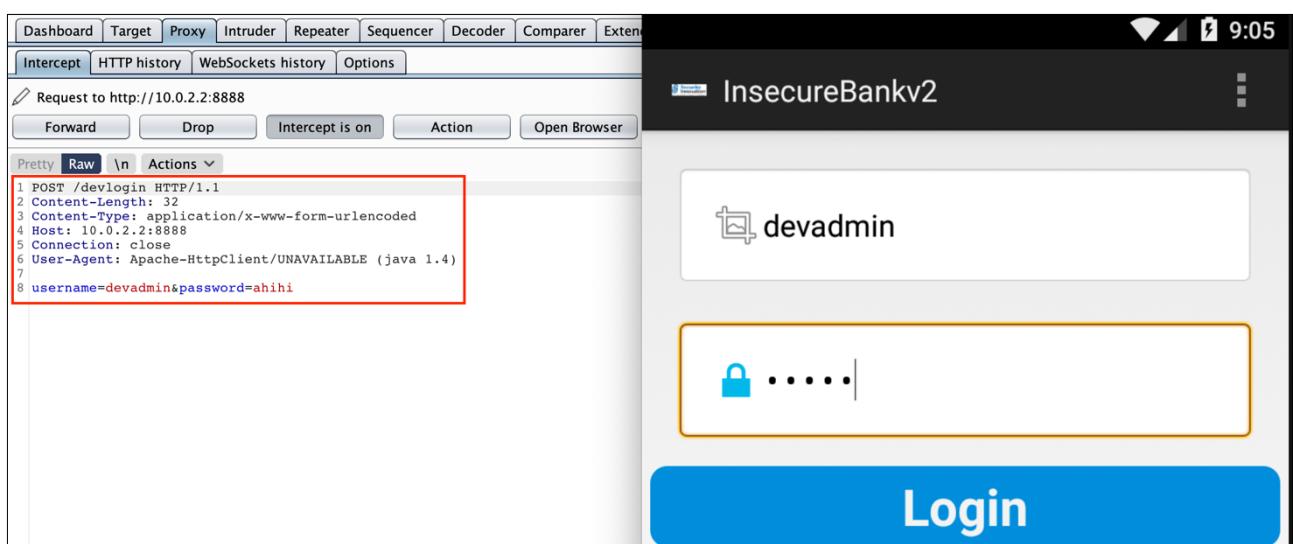
9. Hoàn tất cấu hình, truy cập thử một trang web

Session 4: Android Pentesting

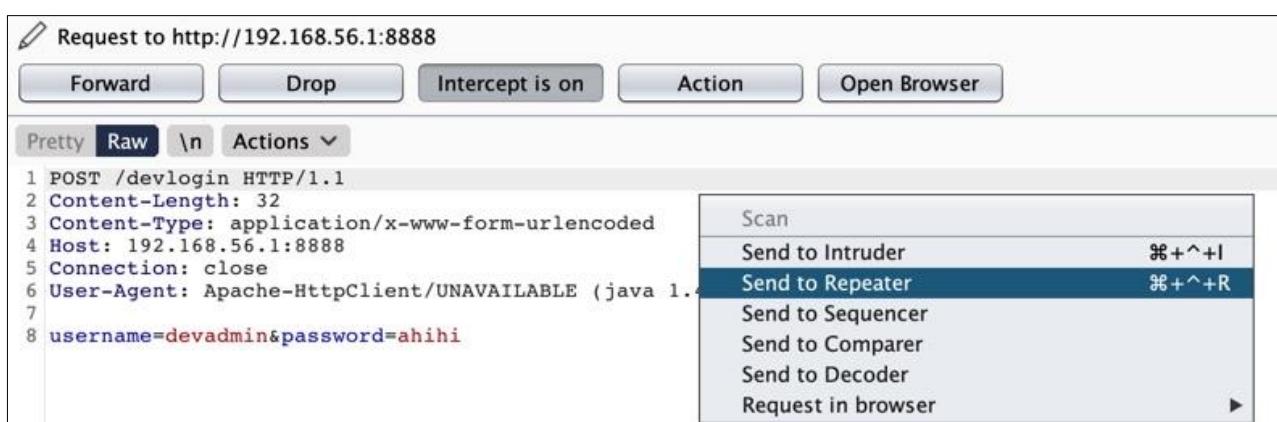


C.3.2 Ứng dụng

Sử dụng ứng dụng Insecurebankv2, khi đó tiến trình bị bắt lại, ta có thể thay đổi các trường của payload



Sử dụng chức năng Repeater, tính năng giúp tiết kiệm thời gian cho việc phải bắt lại request nhiều lần



Chọn *Send* vào Repeater, và sử dụng nhiều lần như thế ở tab này

Session 4: Android Pentesting

The screenshot shows the ZAP interface with the 'Proxy' tab selected. In the 'Request' pane, a POST /devlogin HTTP/1.1 request is shown with the parameters 'username=devadmin&password=ahihihahaha'. The 'Send' button is highlighted with a red box. In the 'Response' pane, a 200 OK response is displayed with the message '{"message": "Correct Credentials", "user": "devadmin"}'.

Tấn công vét cạn mật khẩu bằng cách gửi request vào tab *Intruder*

The screenshot shows the ZAP interface with the 'Intruder' tab selected. A captured POST /devlogin HTTP/1.1 request is shown in the 'Raw' tab. A context menu is open over the password field, with 'Send to Intruder' highlighted. Other options include Scan, Send to Repeater, Send to Sequencer, Send to Comparer, Send to Decoder, and Request in browser.

Vét cạn mật khẩu cho tài khoản “jack”. Ký hiệu § để định dạng cho phần vét cạn

The screenshot shows the ZAP interface with the 'Intruder' tab selected. Under 'Payload Positions', the attack type is set to 'Sniper'. A configuration window is open, showing the base request and the payload '8 username=jack&password=\$ahihih§' highlighted with a red box. To the right are buttons for 'Add §', 'Clear §', 'Auto §', and 'Refresh'.

Tại tab *Intruder > Payloads*, ở *Payload Option* là ta thêm phần payload để vét cạn vào vị trí password.

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 8

Payload type: Simple list Request count: 8

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

| | |
|--------------------------------------|--------------|
| Paste | 123 |
| Load ... | abc |
| Remove | Dinesh@123\$ |
| Clear | ahih |
| Add | jack@123 |
| Add from list ... [Pro version only] | |

Có một request có độ khác với request còn lại

| Request | Payload | Status | Error | Timeout | Length | Comment |
|---------|--------------|--------|-------|---------|--------|---------|
| 0 | | 200 | | | 199 | |
| 1 | 123 | 200 | | | 199 | |
| 2 | abc | 200 | | | 199 | |
| 3 | Dinesh@123\$ | 200 | | | 199 | |
| 4 | jack | 200 | | | 199 | |
| 5 | Jack@123\$ | 200 | | | 204 | |
| 6 | aaaaaaaaa | 200 | | | 199 | |
| 7 | dinesh | 200 | | | 199 | |

Text

Request Response

Pretty Raw Render \n Actions ▾

```

1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 Content-Length: 50
4 Connection: close
5 Date: Wed, 25 Nov 2020 09:40:36 GMT
6 Server: localhost
7
8 {"message": "Correct Credentials", "user": "jack"}

```

Phần History hiển thị lịch sử của các request được gửi từ điện thoại ảo

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type |
|----|----------------------------|--------|--------------|--------|--------|--------|--------|-----------|
| 1 | http://tsu0d.com | GET | / | | | | | |
| 3 | http://inseclab.uit.edu.vn | GET | / | | | | | |
| 4 | http://inseclab.uit.edu.vn | GET | / | | | | | |
| 5 | http://inseclab.uit.edu.vn | GET | /favicon.ico | | | 301 | 242 | HTML |
| 6 | http://10.0.2.2:8888 | POST | /devlogin | ✓ | ✓ | | | |
| 7 | http://192.168.56.1:8888 | POST | /devlogin | ✓ | ✓ | 200 | 208 | JSON |
| 8 | http://192.168.56.1:8888 | POST | /devlogin | ✓ | ✓ | 200 | 208 | JSON |
| 9 | http://192.168.56.1:8888 | POST | /devlogin | ✓ | ✓ | 200 | 208 | JSON |
| 10 | http://192.168.56.1:8888 | POST | /devlogin | ✓ | ✓ | 200 | 208 | JSON |
| 11 | http://192.168.56.1:8888 | POST | /devlogin | ✓ | ✓ | 200 | 208 | JSON |
| 12 | http://192.168.5.81:8888 | POST | /login | ✓ | ✓ | | | |
| 13 | http://192.168.5.81:8888 | POST | /login | ✓ | ✓ | | | |
| 14 | http://192.168.56.1:8888 | POST | /login | ✓ | | 200 | 199 | JSON |

C.4 Hooking với Frida

Frida là bộ công cụ tự động dành cho lập trình viên, kỹ sư dịch ngược và nghiên cứu viên bảo mật với chức năng: tracing, profiling và debugging.

C.4.1 Cài đặt

Máy thật dùng python3

```
pip3 install frida-tools  
pip3 install frida
```

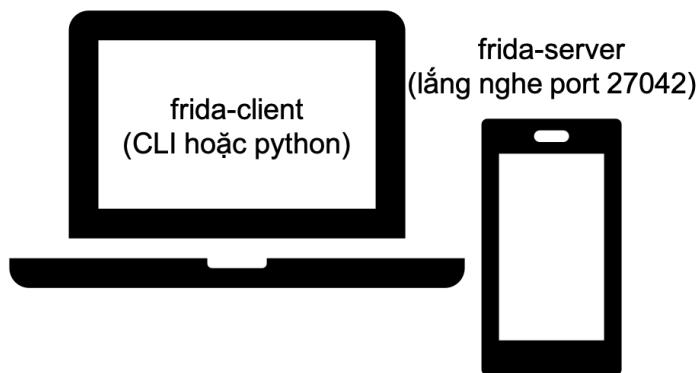
```
└── [✓] Apple ➜ /usr/local/bin
    └── frida
        Usage: frida [options] target

        frida: error: target file, process name or pid must be specified

└── [✓] Apple ➜ /usr/local/bin
    └── python3
        Python 3.8.2 (default, Oct 2 2020, 10:45:42)
        [Clang 12.0.0 (clang-1200.0.32.27)] on darwin
        Type "help", "copyright", "credits" or "license" for more information.
        >>> import frida
        >>> print(frida.__version__)
        14.0.8
        >>> █
```

Điện thoại ảo tải frida-server (<https://github.com/frida/frida/releases>), cụ thể frida-server-14.0.8-android-x86.xz

```
[-] [2]: ~ ~/Downloads ➜ adb push frida-server-14.0.8-android-x86 /data/local/tmp/frida-server  
frida-server-14.0.8-android-x86: 1 file pushed, 0 skipped. 92.1 MB/s (14581988 bytes in 0.151s)  
  
[-] [3]: ~ ~/Downloads ➜ adb shell  
vbox86p:/ # cd /data/local/tmp/  
vbox86p:/data/local/tmp # chmod +x frida-server  
vbox86p:/data/local/tmp # ./frida-server &  
[1] 1992  
vbox86p:/data/local/tmp #
```



Kiểm tra cài đặt thành công hay chưa, bằng cách gõ lệnh sau:

```
└─ frida-ps -U
    PID  Name
    --- -----
    1656  abdb
    1612  android.ext.services
    462   android.hardware.camera.provider@2.4-service
    463   android.hardware.configstore@1.0-service
    220   android.hardware.gnss@1.0-service
    464   android.hardware.graphicsallocator@2.0-service
    173   android.hardware.keymaster@3.0-service
    465   android.hardware.sensors@1.0-service
    466   android.hardware.wifi@1.0-service
    461   android.hidl.allocator@1.0-service
    1320  android.process.acore
    208   audioserver
```

C.4.2 Thực hành

Import thư viện Frida:

```
import frida
```

Dùng hàm get_usb_device() để lấy thông tin thiết bị:

```
device = frida.get_usb_device()
```

Spawn chương trình lên

```
pid=device.spawn("com.android.insecurebankv2")
device.resume(pid)
```

Bây giờ có được tiến trình của chương trình vừa tạo, thêm pid vào session

```
session=device.attach(pid)
```

Inject đoạn hook_script vào chương trình

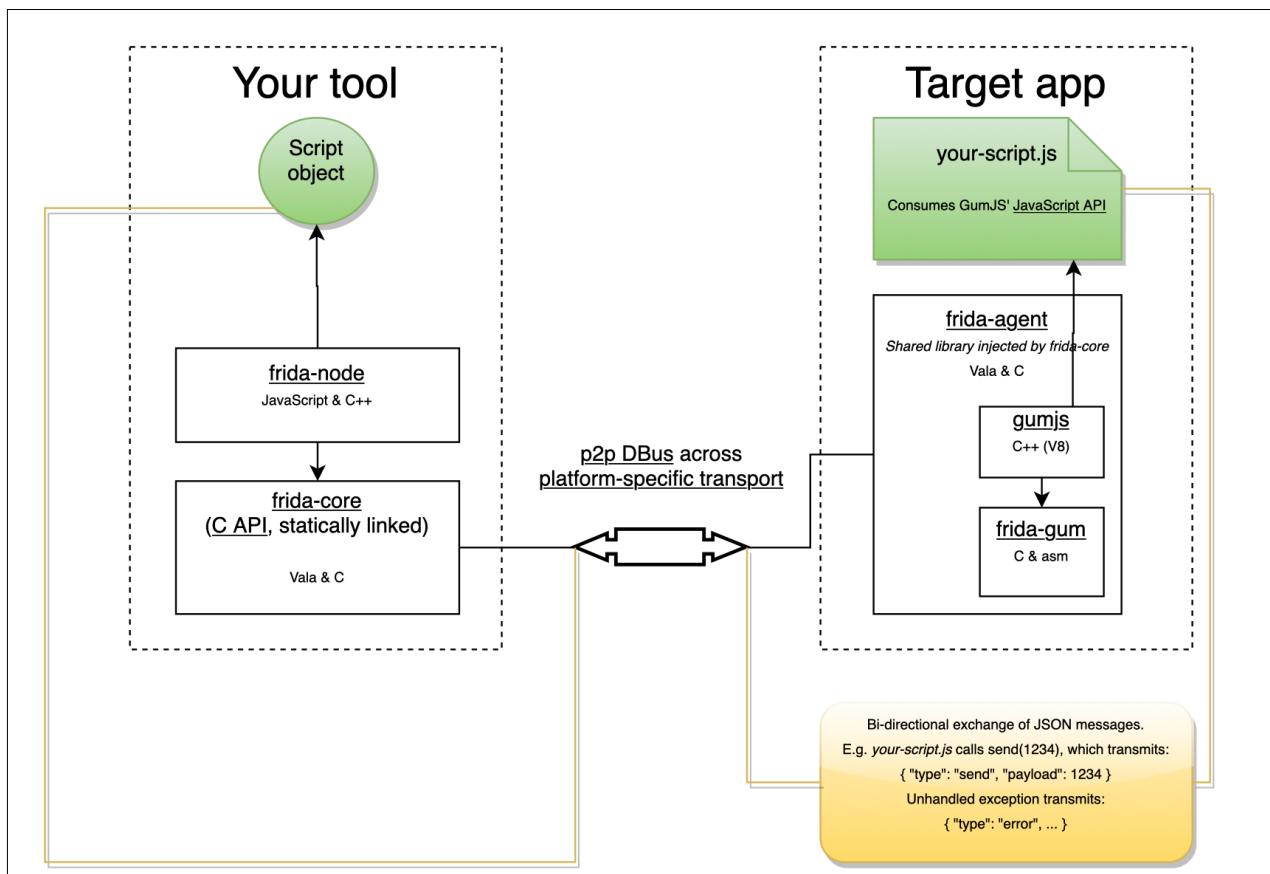
```
script=session.create_script(hook_script)  
script.load()
```

Đoạn hoàn thiện sẽ như thế này:

```
import frida  
import time  
  
device = frida.get_usb_device()  
pid = device.spawn("com.android.insecurebankv2")  
device.resume(pid)  
  
time.sleep(1) # sleep 1 to avoid crash (sometime)  
  
session=device.attach(pid)  
  
hook_script=""  
"""  
  
script=session.create_script(hook_script)  
script.load()  
  
input('...?') # prevent terminate
```

hook_script là gì?

Đây sẽ là một tập lệnh cung cấp cho Frida sử dụng Javascript API, điều này giúp tương tác với Java functions và objects.



Đoạn code cung cấp cho Frida sẽ nằm trong **Java.perform(function () {...})**, khi đó hook_script sẽ như sau:

```
hook_script=""""
Java.perform(function () {
    # do something
});"""
"""
```

Ở phần *Phân tích tĩnh Yêu cầu 5*, chúng ta phải viết code để giải mã thông điệp, nếu trường hợp không tìm thấy key, hay không rõ thuật toán mã hoá thì sao?

Đoạn code Decrypt-Giải mã tại com/android/inscurebankv2/CryptoClas.class

```
public String aesDecryptedString(String var1) throws UnsupportedEncodingException, InvalidKeyException,
NoSuchAlgorithmException, NoSuchPaddingException, InvalidAlgorithmParameterException, IllegalBlockSizeException,
BadPaddingException {
    byte[] var2 = this.key.getBytes("UTF-8");
    this.cipherData = aes256decrypt(this.ivBytes, var2, Base64.decode(var1.getBytes("UTF-8"), 0));
    this.plainText = new String(this.cipherData, "UTF-8");
    return this.plainText;
}
```

- ⇒ Đoạn code nhận vào một đoạn mã hoá crypt-text và trả về đoạn plain-text, suy ra chúng ta không cần quan tâm nó làm gì mà chỉ cần gọi hàm và sử dụng.
- ⇒ Làm thế nào để gọi hàm-function trong Java

Sử dụng **Java.choose()** như sau:

- **Java.choose(className, callbacks)** : enumerate live instances of the **className** class by scanning the Java heap, where **callbacks** is an object specifying:
 - **onMatch(instance)** : called with each live instance found with a ready-to-use **instance** just as if you would have called **Java.cast()** with a raw handle to this particular instance.
This function may return the string **stop** to cancel the enumeration early.
 - **onComplete()** : called when all instances have been enumerated

⇒ Class sẽ được scan trên Java heap, nếu tìm thấy onMatch() sẽ được thực thi dựa vào đó gọi instance tìm được.

Code code code

```

1 import frida
2 import time
3
4 device = frida.get_usb_device()
5 pid = device.spawn("com.android.insecurebankv2")
6 device.resume(pid)
7
8 time.sleep(1) # sleep 1 to avoid crash (sometime)
9
10 session=device.attach(pid)
11
12 hook_script=""""
13 Java.perform
14 (
15     function()
16     {
17         console.log("Inside the hook_script");
18         cryptoClass = Java.choose('com.android.insecurebankv2.CryptoClass',
19         {
20             onMatch : function(instance)
21             {
22                 console.log("Found instance " + instance);
23                 console.log("Result decrypt: " + instance.aesDecryptedString("v/sJpihDCo2ckDmLW5Uwiw=="));
24             },
25             onComplete: function()
26             {
27                 console.log("end");
28             }
29         });
30     }
31 );
32 """
33
34 script=session.create_script(hook_script)
35 script.load()
36
37 input('...?')

```

Kết quả:

```

└─ python3 hook.py
Inside the hook_script
Found instance com.android.insecurebankv2.CryptoClass@214be23
Result decrypt: Jack@123$
end
...?█

```

Ta để ý ứng dụng có tính năng phát hiện root, bây giờ sẽ hook và sửa nó. Đoạn code tại com/andoird/insecurebankv2/PostLogin.class

```
void showRootStatus() {
    boolean var1;
    if (!this.doesSuperuserApkExist("/system/app/Superuser.apk") && !this.doesSUexist()) {
        var1 = false;
    } else {
        var1 = true;
    }

    if (var1) {
        this.root_status.setText("Rooted Device!!!");
    } else {
        this.root_status.setText("Device not Rooted!!!");
    }
}
```

doesSuperuserApkExist() trả về true thì root detected. Hàm này trả về giá trị boolean, ở đây ta hijack giá trị trả về ???

```
private boolean doesSUexist() {
// $FF: Couldn't be decompiled
}

private boolean doesSuperuserApkExist(String var1) {
    boolean var2 = true;
    if (!Boolean.valueOf((new File("/system/app/Superuser.apk")).exists())) {
        var2 = false;
    }

    return var2;
}
```

Sử dụng **Java.use()**, cho phép chúng ta ghi đè một phương-thúc-method trong class.

- **Java.use(className)** : dynamically get a JavaScript wrapper for **className** that you can instantiate objects from by calling **\$new()** on it to invoke a constructor. Call **\$dispose()** on an instance to clean it up explicitly (or wait for the JavaScript object to get garbage-collected, or script to get unloaded). Static and non-static methods are available, and you can even replace a method implementation and throw an exception from it:

```
Java.perform(() => {
    const Activity = Java.use('android.app.Activity');
    const Exception = Java.use('java.lang.Exception');
    Activity.onResume.implementation = function () {
        throw Exception.$new('Oh noes!');
    };
});
```

Đầu tiên, chọn class có method đó

```
classPostLogin = Java.use('com.android.insecurebankv2.PostLogin')
```

Sử dụng implementation để ghi đè

```
classPostLogin.doesSuperuserApkExist.implementation = function()
{
    // làm cái gì đó trong trường hợp này, return true
}
```

Code code code

```

1   import frida
2   import time
3
4   device = frida.get_usb_device()
5   pid = device.spawn("com.android.insecurebankv2")
6   device.resume(pid)
7
8   time.sleep(1) # sleep 1 to avoid crash (sometime)
9
10 session=device.attach(pid)
11
12 hook_script="""
13 Java.perform
14 (
15     function()
16     {
17         console.log("Inside the hook_script");
18         classPostLogin = Java.use('com.android.insecurebankv2.PostLogin');
19         classPostLogin.doesSuperuserApkExist.implementation = function()
20         {
21             // làm cái gì đó trong trường hợp này, return true
22         };
23     }
24 );
25 """
26
27 script=session.create_script(hook_script)
28 script.load()
29
30 input('...?')
```

Yêu cầu 7 Hoàn thiện đoạn code trên và demo.

D. CHALLENGES CTF (BÀI TẬP BẢO MẬT ỨNG DỤNG)

D.1 EVABS

Tập tin APK: **EVABSV5.apk**

Định dạng flag: EVABS{s0m3_rand0m_fl2g_h3r3}

Challenges 1 Hoàn thành 12 levels.

D.2 Droid

Định dạng flag: picoCTF{th1s_1s_fl2g}

Challenges 2 Hoàn thành 5 challenges

1. Nhật ký droid đã đi đâu. Bạn có thể tìm thấy tại: **one.apk**.
2. Tìm kiếm và lấy flag. Bạn có thể tìm thấy tại: **two.apk**.
3. Tìm kiếm và lấy flag. Bạn có thể tìm thấy tại: **three.apk**.
4. Dịch ngược, vá lại tập tin và lấy cờ. Bạn có thể tìm thấy tại: **four.apk**.
5. Dịch ngược, vá lại tập tin và lấy cờ. Bạn có thể tìm thấy tại: **five.apk**.

* *Viết writeup chi tiết*

E. YÊU CẦU

- Sinh viên tìm hiểu và thực hành theo hướng dẫn theo nhóm sinh viên đã sắp xếp.
- Nộp mã nguồn (**Code**) và báo cáo kết quả chi tiết những việc (**Report**) đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Trong file báo cáo yêu cầu **ghi rõ** nhóm sinh viên thực hiện.
- Đặt tên theo định dạng: [Mã lớp]-Lab2_MSSV1-MSSV2-MSSV3.pdf
Ví dụ: [NT213.K11.ANTN.1]-Lab2_1552xxxx-1552yyyy-1552zzzz.pdf
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trẽ,... sẽ được xử lý tùy mức độ vi phạm.

HẾT