

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

----- ❦ ★ ❦ -----



BÁO CÁO KẾT QUẢ THỰC HÀNH BẢO MẬT WEB VÀ ỨNG DỤNG

Lab 04: PENTESTING ANDROID APPLICATIONS

Giảng viên giảng dạy: Nghi Hoàng Khoa

Nhóm sinh viên thực hiện:

- | | |
|-----------------------|----------|
| 1. Phạm Khôi Nguyên | 18520114 |
| 2. Phan Thanh Hải | 18520705 |
| 3. Nguyễn Lý Đình Nhì | 18521205 |

TP. HỒ CHÍ MINH, 05/2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

----- ❧ ★ ❧ -----



BÁO CÁO KẾT QUẢ THỰC HÀNH BẢO MẬT WEB VÀ ỨNG DỤNG

Lab 04: PENTESTING ANDROID APPLICATIONS

Giảng viên giảng dạy: Nghi Hoàng Khoa

Nhóm sinh viên thực hiện:

- | | |
|-----------------------|----------|
| 1. Phạm Khôi Nguyên | 18520114 |
| 2. Phan Thanh Hải | 18520705 |
| 3. Nguyễn Lý Đình Nhì | 18521205 |

TP. HỒ CHÍ MINH, 05/2021

MỤC LỤC

LỜI NÓI ĐẦU	1
BỔ SUNG PHẦN CÀI ĐẶT MÔI TRƯỜNG ĐỂ CHẠY ĐƯỢC PHẦN ĐĂNG NHẬP TRONG APP THỰC HÀNH.....	2
YÊU CẦU 1.....	5
YÊU CẦU 2.....	6
YÊU CẦU 3.....	7
YÊU CẦU 4.....	8
YÊU CẦU 5.....	9
YÊU CẦU 6.....	10
YÊU CẦU 7.....	13

LỜI NÓI ĐẦU

Đây là phần bài làm của nhóm cho bài tập thực hành buổi 04 về pentest (kiểm thử xâm nhập) ứng dụng Android. Toàn bộ nội dung thực hành được triển khai trên môi trường Windows của máy thật.

Cảm ơn anh Nghi Hoàng Khoa trong thời gian tuần vừa qua chịu khó trả lời những câu hỏi, những thắc mắc của nhóm về bài tập thực hành buổi 04 trên lớp và ngoài buổi học ạ.

BỔ SUNG PHẦN CÀI ĐẶT MÔI TRƯỜNG ĐỂ CHẠY ĐƯỢC PHẦN ĐĂNG NHẬP TRONG APP THỰC HÀNH

Bước 1: Ta nhấn giữ phím **Shift** và đồng thời nhấn chuột phải vào thư mục `platform-tools` đã được giải nén trước đó. Sau đó, ta nhấn chọn **Open command window here** (Một số máy Windows 10 có thể thấy **PowerShell** thay vì **command window**) để mở **Command Prompt**.

Bước 2: Ta truy cập vào thư mục `AndroLabServer` sử dụng lệnh sau: `cd .\AndroLabServer\`.

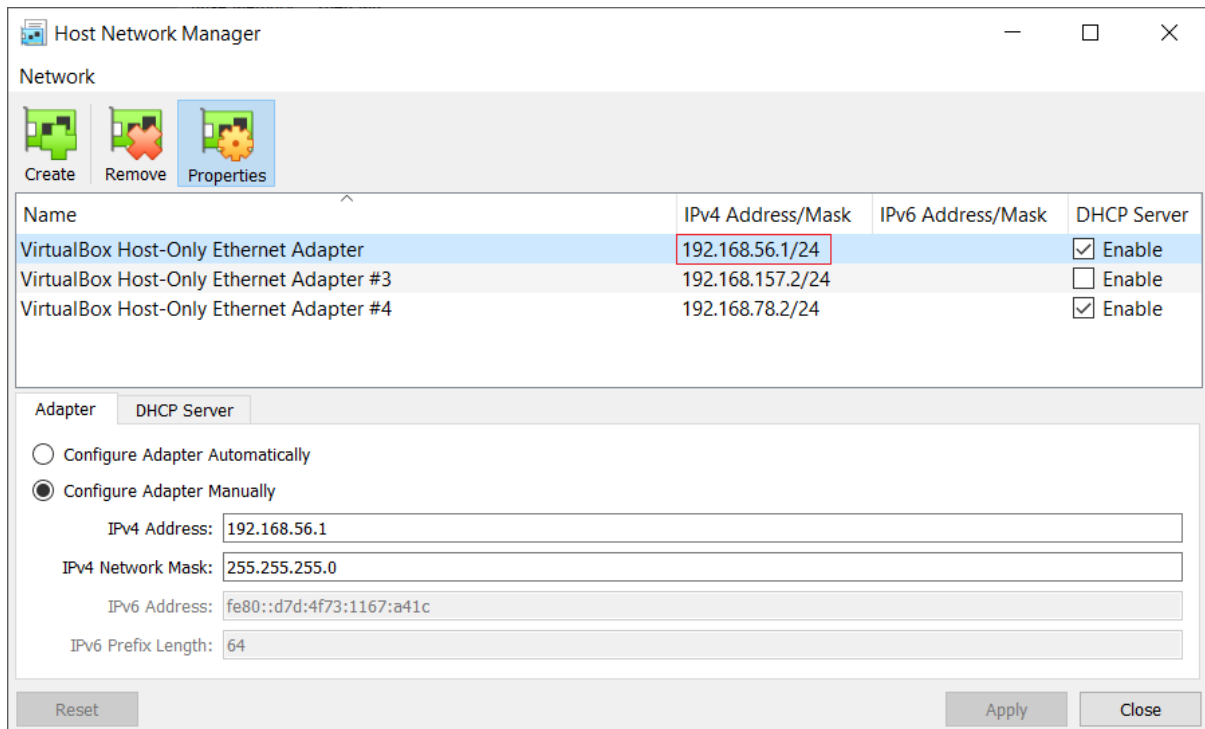
Bước 3: Nếu chưa tải những package có trong tập tin `requirements.txt`, thì ta tiến hành tải sử dụng lệnh sau: `pip install -r requirements.txt`.

```
PS D:\Virtual Machine\platform-tools> cd .\AndroLabServer\
PS D:\Virtual Machine\platform-tools\AndroLabServer> pip install -r requirements.txt
Collecting flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    | 94 kB 183 kB/s
Collecting sqlalchemy
  Downloading SQLAlchemy-1.4.13-cp39-cp39-win_amd64.whl (1.5 MB)
    | 1.5 MB 126 kB/s
Collecting simplejson
  Downloading simplejson-3.17.2.tar.gz (83 kB)
    | 83 kB 98 kB/s
Collecting web.py
  Downloading web.py-0.62.tar.gz (623 kB)
    | 623 kB 435 kB/s
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing wheel metadata ... done
Collecting cherrypy
```

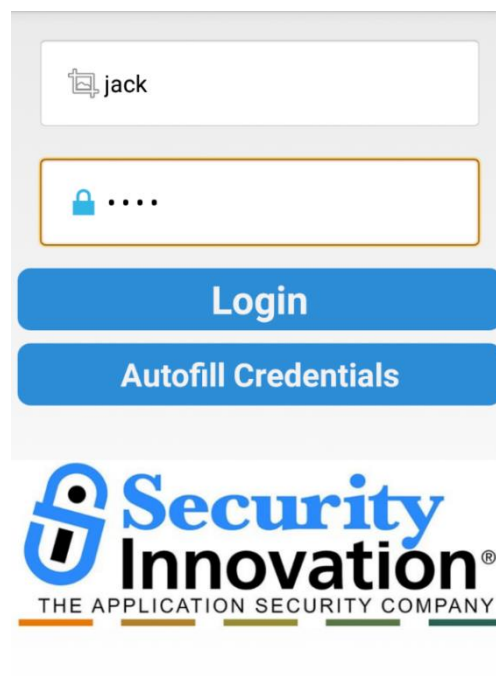
Bước 4: Sau khi tải xong, ta chạy tập tin `app.py` sử dụng lệnh sau: `python app.py`.

```
PS D:\Virtual Machine\platform-tools\AndroLabServer> python app.py
The server is hosted on port: 8888
```

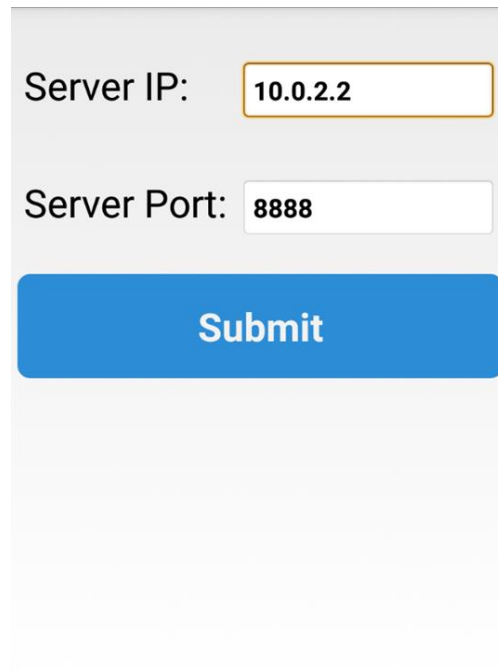
Bước 5: Ta mở app VirtualBox lên. Đối với Windows 10, ta vào **File → Host Network Manager...** (có thể sử dụng tổ hợp phím **Ctrl + H** thay cho thao tác này). Ta ghi chú lại địa chỉ IP đầu tiên như trong hình dưới:



Bước 6: Ta mở điện thoại ảo đã tạo lên và mở app **InsecureBankv2** lên. Nhập thông tin đăng nhập là **jack:1234** vào 2 trường thông tin như hình dưới và bấm nút **Login**.

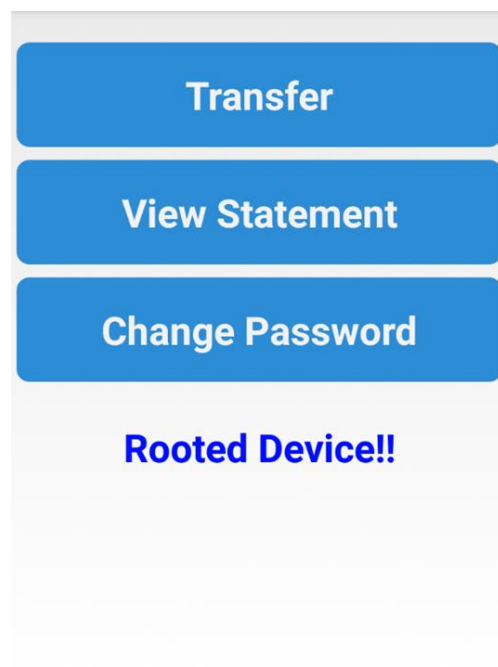


Bước 7: Sau khi bấm nút **Login** sẽ xuất hiện giao diện như hình bên dưới. Ta tiến hành thay đổi giá trị tại trường **Server IP** thành giá trị IP mà ta đã ghi chú lại ở **bước 5**. Sau đó, ta bấm nút **Submit**.



A web form with a light gray background. It contains two input fields: "Server IP:" with the value "10.0.2.2" and "Server Port:" with the value "8888". Below the inputs is a large blue button with the text "Submit" in white.

Bước 8: Sau khi bấm nút **Submit** sẽ quay lại giao diện như ở **bước 6**. Ta bấm nút **Submit** lại và xuất hiện thông báo root thiết bị thành công như hình bên dưới:

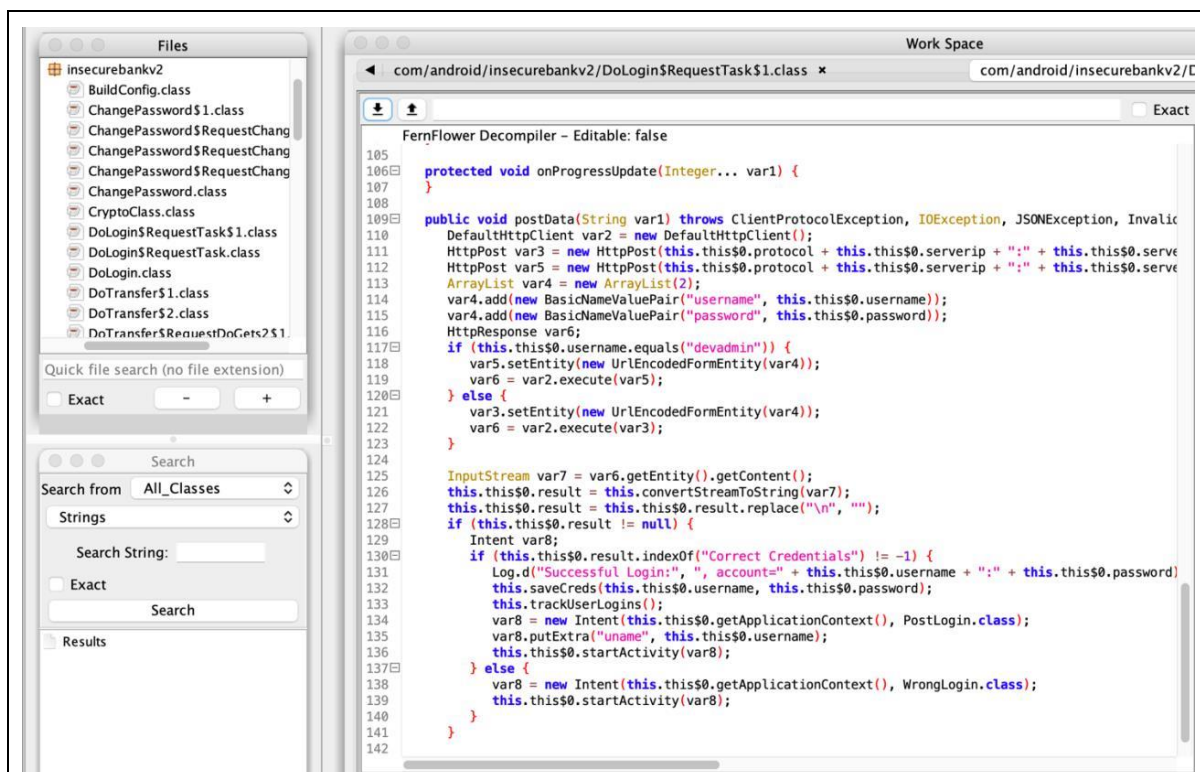


A web interface with a light gray background. It features three stacked blue buttons with white text: "Transfer", "View Statement", and "Change Password". Below these buttons, the text "Rooted Device!!" is displayed in a bold, blue font.

Bên trong màn hình Command Prompt kia, sẽ xuất hiện dòng thông báo đăng nhập thành công:

```
PS D:\Virtual Machine\platform-tools\AndroLabServer> python app.py
The server is hosted on port: 8888
u= <User 'jack'>
{"message": "Correct Credentials", "user": "jack"}
```

YÊU CẦU 1



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

Ta quan sát dòng code từ dòng 117 đến dòng 123:

```
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 }
```

Như vậy, ta thấy chỉ cần nhập thông tin username là devadmin là đã tự động đăng nhập được, không quan tâm đến mật khẩu là gì. Chương trình như vậy thật sự rất bất thường một cách quá *ngộ nghĩnh*.

YÊU CẦU 2

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

Ta sẽ xem thử trong cơ sở dữ liệu có chứa thông tin đăng nhập nhạy cảm hay không.

Bước 1: Đưa vào Commandline Shell của điện thoại ảo đã tạo, ta sử dụng lệnh `.\adb shell`.

Bước 2: Ta truy cập đến thư mục chứa cơ sở dữ liệu của app theo đường dẫn `/data/data/<tên_app>/databases/`. Ta sử dụng lệnh `cd /data/data/com.android.insecurebankv2/databases`.

Bước 3: Ta mở SQLite Shell để tiến hành thực hiện các thao tác kiểm tra trên cơ sở dữ liệu sử dụng lệnh `sqlite3 mydb` (mydb là cơ sở dữ liệu chứa thông tin cần kiểm tra).

```
PS D:\Virtual Machine\platform-tools> .\adb shell
vbox86p:/ # cd data/data/com.android.insecurebankv2/databases
vbox86p:/data/data/com.android.insecurebankv2/databases # sqlite3 mydb
SQLite version 3.18.2 2017-07-21 07:56:09
Enter ".help" for usage hints.
```

Bước 4: Ta kiểm tra xem trong cơ sở dữ liệu mydb có những bảng nào sử dụng lệnh `.tables` trong SQLite.

```
sqlite> .table
android_metadata  names
```

Như ta thấy, chỉ có 2 bảng là `android_metadata` và `names` trong cơ sở dữ liệu mydb.

Bước 5: Ta hiển thị toàn bộ thông tin có trong các bảng ở **bước 4** sử dụng lệnh `select * from <tên_bảng>` trong SQLite.

```
sqlite> select * from names;
1|jack
sqlite> select * from android_metadata;
en_US
```

Như ta thấy, không hiện thông tin gì nhạy cảm. Do đó, ta cho rằng dữ liệu lưu trữ an toàn.

YÊU CẦU 3

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

Bước 1: Để vào Commandline Shell của điện thoại ảo đã tạo, ta sử dụng lệnh `.\adb shell`.

Bước 2: Ta truy cập đến thư mục chứa cơ sở dữ liệu của app theo đường dẫn `/data/data/<tên_app>/`. Ta sử dụng lệnh `cd /data/data/com.android.insecurebankv2`.

Bước 3: Ta chạy cấu trúc lệnh `grep -r '<string_to_find>' $(find)`, trong đó `<string_to_find>` là những từ khóa gợi ý trong nội dung yêu cầu 3.

```
PS D:\Virtual Machine\platform-tools> .\adb shell
vbox86p:/ # cd data/data/com.android.insecurebankv2
vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'device' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'imei' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'phone' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'mdn' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'uuid' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 # grep -r 'user' $(find)
1|vbox86p:/data/data/com.android.insecurebankv2 #
```

Như ta thấy, không hiện thông tin gì nhạy cảm. Do đó, ta cho rằng thông tin nhạy cảm tạm thời không có lưu lại trên thiết bị.

YÊU CẦU 4

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

Thư mục sao lưu chứa thông tin user, lịch sử đăng nhập và các log tương tác.

Ta có đoạn dữ liệu được mã hoá:

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="superSecurePassword">u734b1SGyPt7eobqiWxF0g==&#10;  </string>
  <string name="EncryptedUsername">amFjaw==&#13;&#10;  </string>
</map>
```

Ta thấy thông tin user đã được mã hóa.

YÊU CẦU 5

Viết chương trình giải mã đoạn dữ liệu mã hoá (python3 chẳng hạn...)

Ta có chương trình giải mã đoạn dữ liệu mã hóa được viết bằng Python như sau:

```
1  from Crypto.Cipher import AES
2  import base64
3
4  def unpad(st):
5      return st[:-(st[-1])]
6
7  def decrypt(key, cipher, iv):
8      return AES.new(key, AES.MODE_CBC, iv).decrypt(cipher)
9
10 key = "This is the super secret key 123"
11 iv = '\x00'*16
12 cipher = base64.b64decode("u734blSGyPt7eobqiWxF0g==")
13
14 result=unpad(decrypt(key, cipher, iv)).decode("utf-8")
15 print(result)
```

Ta chạy thử và xem kết quả của chương trình trên (chạy trên online compiler <https://onecompiler.com/>):



Trên hình là mật khẩu của user jack.

YÊU CẦU 6

```
114 void showRootStatus() {  
115     int n;  
116     if (this.doesSuperuserApkExist("/system/app/Superuser.apk") || this.doesSUexist()) {  
117         n = 1;  
118     }  
119     else {  
120         n = 0;  
121     }  
122     if (n == 1) {  
123         this.root_status.setText((CharSequence) "Rooted Device!!");  
124     }  
125     else {  
126         this.root_status.setText((CharSequence) "Device not Rooted!!");  
127     }  
128 }
```

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.

Trước tiên, ta dùng apktool để dịch ngược tập tin apk.

```
D:\Virtual Machine\platform-tools>apktool d InsecureBankv2.apk  
I: Using Apktool 2.5.0 on InsecureBankv2.apk  
I: Loading resource table...  
I: Decoding AndroidManifest.xml with resources...  
I: Loading resource table from file: C:\Users\Win 10\AppData\Local\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...
```

Sau đó, ta truy cập vào đường dẫn `smali/com/android/insecurebankv2/PostLogin.smali`. Ta thay đổi code trong tập tin `PostLogin.smali` sao cho nhánh `else` sẽ set text thành `Rooted Device!!` thay vì `Device not Rooted!!` như code cũ. Khi đó dù nhánh `if` hay nhánh `else` được thực hiện thì đều in ra màn hình dòng chữ `Rooted Device!!`.

```

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     iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;-.root_status:Landroid/widget/TextView;
474
475     const-string v2, "Rooted Device!!"
476
477     invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V
478
479     goto :goto_1
480 .end method

```

Tiếp theo, ta vá lại tập tin apk sử dụng apktool.

```

D:\Virtual Machine\platform-tools>apktool b InsecureBankv2 -o InsecureBankv2_1.apk
I: Using Apktool 2.5.0
I: Checking whether sources has changed...
I: Smaling smali 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...

```

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. Ta tiến hành tạo keystore và dùng công cụ apksigner.

```

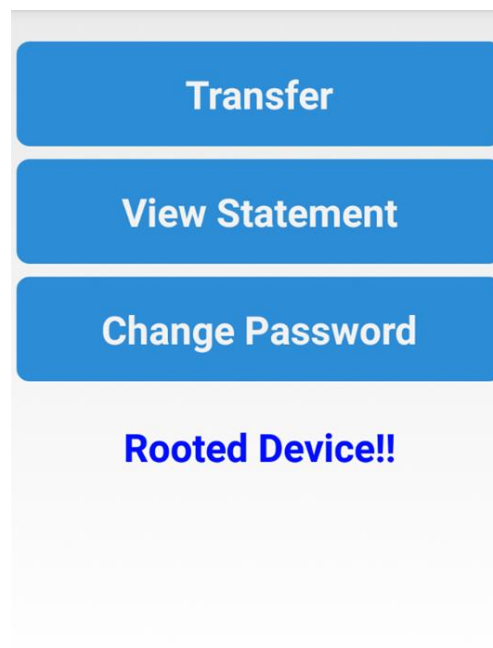
C:\Program Files\Java\jdk-16.0.1\bin>keytool -genkeypair -v -keystore key.keystore -alias publishingdoc -keyalg RSA -key
size 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: 123456
What is the name of your organizational unit?
[Unknown]: 123456
What is the name of your organization?
[Unknown]: 123456
What is the name of your City or Locality?
[Unknown]: 123456
What is the name of your State or Province?
[Unknown]: 123456
What is the two-letter country code for this unit?
[Unknown]: 123456
Is CN=123456, OU=123456, O=123456, L=123456, ST=123456, C=123456 correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=123456, OU=123456, O=123456, L=123456, ST=123456, C=123456
[Storing key.keystore]

C:\Program Files\Java\jdk-16.0.1\bin>apksigner sign --ks key.keystore InsecureBankv2_1.apk
Keystore password for signer #1:

```

Sau đó, ta vô lại app:



YÊU CẦU 7

```
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('...?')
```

Hoàn thiện đoạn code trên và demo.

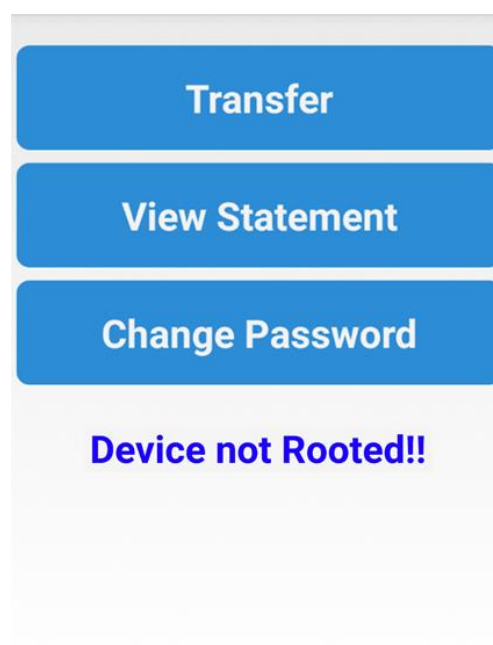
Đoạn code sau khi hoàn thiện như sau:


```

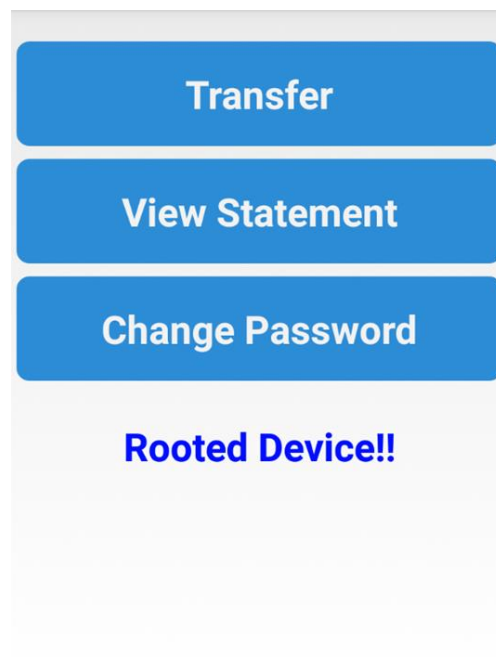
1  import frida
2  import time
3
4  device = frida.get_usb_device()
5  pid = device.spawn("com.android.insecurebankv2")
6  device.resume(pid)
7
8  time.sleep(1) # sleep 1 to avoid crash (sometime)
9
10 session=device.attach(pid)
11
12 hook_script="""
13 Java.perform
14 (
15     function()
16     {
17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

```

Trước khi chạy đoạn code trên:



Sau khi chạy đoạn code trên ta vào lại app:



----- HẾT -----