

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 4: Pentesting Android Applications

GVHD: Nghi Hoàng Khoa

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.P11.ATCL.1

STT	Họ và tên	MSSV	Email
1	Tôn Thất Bình	21520639	2152xxxx@gm.uit.edu.vn
2	Nguyễn Văn Hào	20521293	2052xxxx@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%
3	Bài tập 3	100%
4	Bài tập 4	100%
5	Bài tập 5	100%
6	Bài tập 6	100%
7	Bài tập 7	0%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

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

```
public void postData(String valueIWantToSend) throws ClientProtocolException, IOException, JSONException, InvalidKeyException {  
    HttpResponse responseBody;  
    DefaultHttpClient defaultHttpClient = new DefaultHttpClient();  
    HttpPost httpPost = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport);  
    HttpPost httpPost2 = new HttpPost(DoLogin.this.protocol + DoLogin.this.serverip + ":" + DoLogin.this.serverport);  
    List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);  
    nameValuePairs.add(new BasicNameValuePair("username", DoLogin.this.username));  
    nameValuePairs.add(new BasicNameValuePair("password", DoLogin.this.password));  
  
    if (DoLogin.this.username.equals("devadmin")) {  
        httpPost2.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
        responseBody = defaultHttpClient.execute(httpPost2);  
    } else {  
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
        responseBody = defaultHttpClient.execute(httpPost);  
    }  
  
    InputStream in = responseBody.getEntity().getContent();  
    DoLogin.this.result = convertStreamToString(in);  
    DoLogin.this.result = DoLogin.this.result.replace("\n", "");  
  
    if (DoLogin.this.result != null) {  
        if (DoLogin.this.result.indexOf("Correct Credentials") != -1) {  
            Log.d("Successful Login", "account " + DoLogin.this.username + ":" + DoLogin.this.password);  
            saveCreds(DoLogin.this.username, DoLogin.this.password);  
            trackUserLogins();  
            Intent pL = new Intent(DoLogin.this.getApplicationContext(), PostLogin.class);  
            pL.putExtra("uname", DoLogin.this.username);  
            DoLogin.this.startActivity(pL);  
            return;  
        }  
    }  
    Intent xi = new Intent(DoLogin.this.getApplicationContext(), WrongLogin.class);  
    DoLogin.this.startActivity(xi);  
}
```

- **Phân tích:** Code đang thực hiện việc đăng nhập qua http protocol. Đầu tiên là sẽ khởi tạo các đối tượng sau đó lấy thông tin đăng nhập được thực hiện bởi HttpPost. Tiếp tục thực hiện đến kết nối máy chủ bằng kết nối http và gửi thông tin login bao gồm username và password. Sau đó server sẽ trả kết quả về và kiểm tra xem cho phép đăng nhập hay không.

- Điểm bất thường:

+ Để lộ thông tin đăng nhập mặc định:

```
if (DoLogin.this.username.equals("devadmin"))
```

+ Không mã hóa thông tin đăng nhập:

```
nameValuePairs.add(new BasicNameValuePair("username", DoLogin.this.username));  
nameValuePairs.add(new BasicNameValuePair("password", DoLogin.this.password));
```

+ Sử dụng thư viện cũ

```
DefaultHttpClient defaultHttpClient = new DefaultHttpClient();
```

+ Sử dụng HTTP thay vì HTTPS.

```
HttpPost httpPost = new HttpPost
```

+ Kiểm tra xác thực đơn giản và không an toàn.

```
if (DoLogin.this.result.indexOf(str:"Correct Credentials") != -1)
```

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

- Truy cập vào thư mục **/data/data/com.android.insecurebankv2/databases** ta tìm thấy cơ sở dữ liệu của ứng dụng. Thực hiện các câu truy vấn cơ bản trong cơ sở dữ liệu ta thấy được tên của người dùng được lưu trực tiếp trong cơ sở dữ liệu mà không qua mã hóa, kết quả trả về là dữ liệu lưu trữ không an toàn:

```
sqlite> .tables  
android_metadata  names  
sqlite>  
sqlite> select * from names  
...> ;  
1|dinesh  
2|dinesh  
3|dinesh  
4|dinesh  
5|dinesh
```

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...

- Sử dụng lệnh **grep -ri -E**

"deviceId|userId|imei|deviceSerialNumber|devicePrint|phone|XDSN|mdn|IMSI|uuid|username|password|key" \$(find) >> found.txt

- Trong đó:

+ ri : tìm kiếm đệ quy và không phân biệt chữ hoa hay thường

+ E : tìm kiếm nhiều chuỗi cùng lúc

- Kết quả trả về cho thấy mật khẩu và tên người dùng có được lưu trên thiết bị nhưng dưới dạng đã mã hóa

```
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer/backup_contents$ adb shell
x86_64:/ $ su
:/ # cd /data/data/com.android.insecurebankv2
r|devicePrint|phone|XDSN|mdn|IMSI|uuid|username|password|key" $(find) <
./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./shared_prefs/mySharedPreferences.xml: <string name="EncryptedUsername">ZGluZXNo&#13;&#10; </string>
Binary file ./databases/mydb-wal matches
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./shared_prefs/mySharedPreferences.xml: <string name="EncryptedUsername">ZGluZXNo&#13;&#10; </string>
./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
Binary file ./databases/mydb-wal matches
Binary file ./databases/mydb-wal matches
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
./found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10; </string>
/data/data/com.android.insecurebankv2 #
```

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

- Thực hiện giải nén file backup

```
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer$ mkdir backup_contents
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer$ tar -xf backup_compressed.tar -C backup_contents
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer$ cd backup_contents/
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer/backup_contents$ ls
apps shared
```

- Kết quả tìm kiếm các chuỗi trong file sử dụng lệnh **grep -ri -E**

"deviceId|userId|imei|deviceSerialNumber|devicePrint|phone|XDSN|mdn|IMSI|uuid|username|password|key" \$(find)

```

ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/AndroLabServer/backup_content$ grep -ri -E "deviceId|userId|imei|deviceSerialNumber|
devicePrint|phone|XDSN|mdn|IMSI|uid|username|password|key" $(find)
grep: ./apps/com.android.insecurebankv2/db/mydb-wal: binary file matches
./apps/com.android.insecurebankv2/sp/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;
</string>
./apps/com.android.insecurebankv2/sp/mySharedPreferences.xml: <string name="EncryptedUsername">ZGLuZXNo&#13;&#10; </string>
grep: ./apps/com.android.insecurebankv2/a/base.apk: binary file matches
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
grep: ./apps/com.android.insecurebankv2/db/mydb-wal: binary file matches
./apps/com.android.insecurebankv2/sp/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;
</string>
./apps/com.android.insecurebankv2/sp/mySharedPreferences.xml: <string name="EncryptedUsername">ZGLuZXNo&#13;&#10; </string>
grep: ./apps/com.android.insecurebankv2/a/base.apk: binary file matches
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
./apps/com.android.insecurebankv2/r/found.txt:./shared_prefs/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXj
SoFdg0e61fHxJg==&#10; </string>
grep: ./apps/com.android.insecurebankv2/db/mydb-wal: binary file matches
grep: ./apps/com.android.insecurebankv2/db/mydb-wal: binary file matches
./apps/com.android.insecurebankv2/sp/mySharedPreferences.xml: <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;

```

- Các thông tin cần được mã hóa là username và password của người dùng.

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

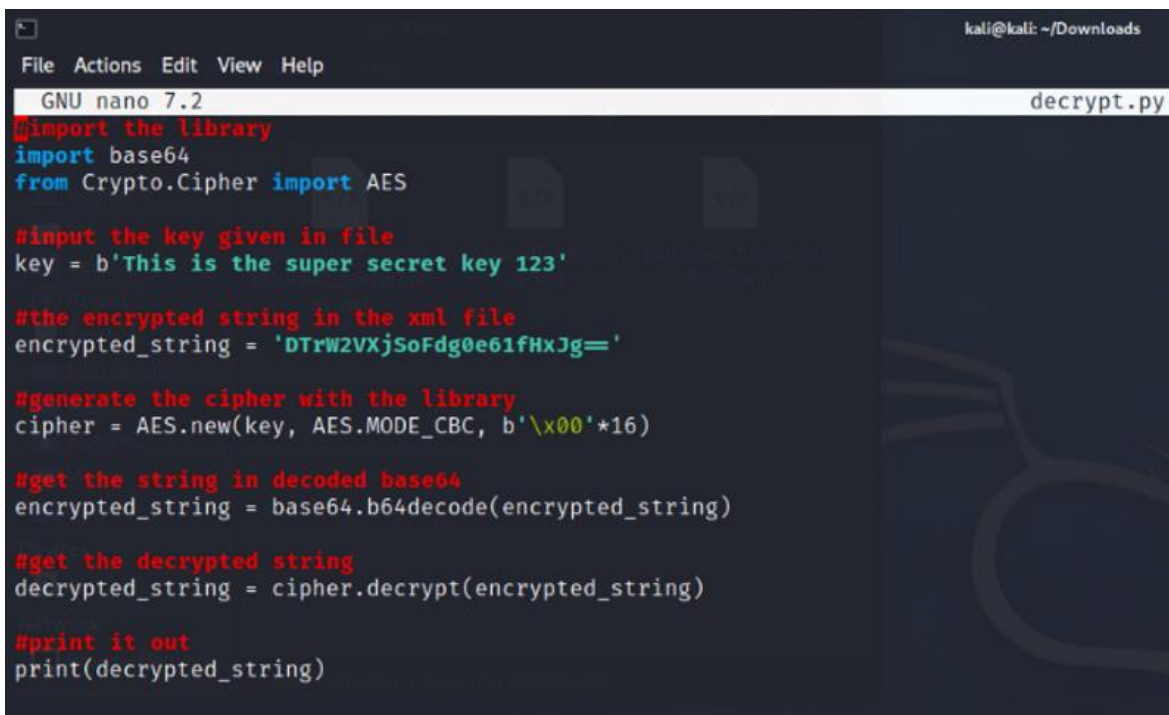
- Tiếp tục tìm kiếm thông tin liên quan đến cơ chế mã hoá thì ta thấy được mã hoá đang sử dụng là aes cbc, với key là This is the super secret key 123 và iv như hình

```

1 public class CryptoClass {
2     private String key;
3     private byte[] ivBytes;
4
5     public CryptoClass() {
6         super();
7         this.key = "This is the super secret key 123";
8         this.ivBytes = new byte[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
9     }
10
11     public static byte[] aes256Decrypt(final byte[] iv, final byte[] key, final byte[] input)
12         throws UnsupportedOperationException, NoSuchAlgorithmException,
13             NoSuchPaddingException, InvalidKeyException, InvalidAlgorithmParameterException,
14             IllegalBlockSizeException, BadPaddingException {
15         IvParameterSpec params = new IvParameterSpec(iv);
16         SecretKeySpec key2 = new SecretKeySpec(key, "AES");
17         Cipher instance = Cipher.getInstance("AES/CBC/PKCS5Padding");
18         instance.init(Cipher.DECRYPT_MODE, key2, params);
19         return instance.doFinal(input);
20     }
21
22     public static byte[] aes256Encrypt(final byte[] iv, final byte[] key, final byte[] input)
23         throws UnsupportedOperationException, NoSuchAlgorithmException,
24             NoSuchPaddingException, InvalidKeyException, InvalidAlgorithmParameterException,
25             IllegalBlockSizeException, BadPaddingException {
26         IvParameterSpec params = new IvParameterSpec(iv);
27         SecretKeySpec key2 = new SecretKeySpec(key, "AES");
28         Cipher instance = Cipher.getInstance("AES/CBC/PKCS5Padding");
29         instance.init(Cipher.ENCRYPT_MODE, key2, params);
30         return instance.doFinal(input);
31     }
32 }
33

```

- Đồng thời như bên trên ta cũng thấy được các thông tin mã hoá
- Thực hiện code python để lấy thông tin và giải mã
- Với chương trình này ta sẽ thực hiện import các thư viện mật mã, truyền các tham số key và string mã hoá, tạo ra cipher bằng thư viện aes và cuối cùng thực hiện quá trình giải mã và thu kết quả ra màn hình:



```

kali@kali: ~/Downloads
File Actions Edit View Help
GNU nano 7.2 decrypt.py
import the library
import base64
from Crypto.Cipher import AES

#input the key given in file
key = b'This is the super secret key 123'

#the encrypted string in the xml file
encrypted_string = 'DTrW2VXjSoFdgoe61fHxJg=='

#generate the cipher with the library
cipher = AES.new(key, AES.MODE_CBC, b'\x00'*16)

#get the string in decoded base64
encrypted_string = base64.b64decode(encrypted_string)

#get the decrypted string
decrypted_string = cipher.decrypt(encrypted_string)

#print it out
print(decrypted_string)

```

- Ta thấy được thông tin bị mã hoá là mật khẩu của tài khoản Dinesh:

```

File Actions Edit View Help

(kali@kali)-[~/Downloads]
$ nano decrypt.py

(kali@kali)-[~/Downloads]
$ python decrypt.py
b'Dinesh@123$\x05\x05\x05\x05\x05'

(kali@kali)-[~/Downloads]
$

```

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.

- Dùng lệnh **grep -ri "root" \$(find)** để tìm các file có chứa thông điệp về root. Ta tìm thấy file cần sửa là **PostLogin.smali**

```

ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank/InsecureBankv2/smali/com/android/insecurebankv2$ grep -ri "root" $(find)
./R$id.smali:.field public static final action_bar_root:I = 0x7fd0052
./R$id.smali:.field public static final rootStatus:I = 0x7fd0080
./PostLogin.smali:.field root_status:Landroid/widget/TextView;
./PostLogin.smali:    .local v1, "rootFile":Ljava/io/File;
./PostLogin.smali:    iput-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
./PostLogin.smali:    invoke-virtual {p0}, Lcom/android/insecurebankv2/PostLogin;->showRootStatus()V
./PostLogin.smali:.method showRootStatus()V
./PostLogin.smali:    .local v0, "isrooted":Z
./PostLogin.smali:    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
./PostLogin.smali:    const-string v2, "Rooted Device!!"
./PostLogin.smali:    .end local v0    # "isrooted":Z
./PostLogin.smali:    .restart local v0    # "isrooted":Z
./PostLogin.smali:    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
./PostLogin.smali:    const-string v2, "Device not Rooted!!"
./R$id.smali:.field public static final action_bar_root:I = 0x7fd0052
./R$id.smali:.field public static final rootStatus:I = 0x7fd0080
./PostLogin.smali:.field root_status:Landroid/widget/TextView;
./PostLogin.smali:    .local v1, "rootFile":Ljava/io/File;
./PostLogin.smali:    iput-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
./PostLogin.smali:    invoke-virtual {p0}, Lcom/android/insecurebankv2/PostLogin;->showRootStatus()V
./PostLogin.smali:.method showRootStatus()V
./PostLogin.smali:    .local v0, "isrooted":Z
./PostLogin.smali:    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
./PostLogin.smali:    const-string v2, "Rooted Device!!"
./PostLogin.smali:    .end local v0    # "isrooted":Z
./PostLogin.smali:    .restart local v0    # "isrooted":Z
./PostLogin.smali:    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank/InsecureBankv2/smali/com/android/insecurebankv2$

```

- Method sau nhằm để xác định trạng thái root của thiết bị


```

.method showRootStatus()V
    .locals 3

    .prologue
    const/4 v1, 0x1

    .line 86
    const-string v2, "/system/app/Superuser.apk"

    invoke-direct {p0, v2}, Lcom/android/insecurebankv2/PostLogin;->doesSuperuserApkExist(Ljava/lang/String;)Z

    move-result v2

    if-nez v2, :cond_0

    .line 87
    invoke-direct {p0}, Lcom/android/insecurebankv2/PostLogin;->doesSUexist()Z

    move-result v2

    if-eqz v2, :cond_1

    :cond_0
    move v0, v1

    .line 88
    .local v0, "isrooted":Z
    :goto_0
    if-ne v0, v1, :cond_2

    .line 90
    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
    const-string v2, "Rooted Device!!"

    invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

    .line 96
    :goto_1
    return-void

    .line 87
    .end local v0    # "isrooted":Z
    :cond_1
    const/4 v0, 0x0

    goto :goto_0

    .line 94
    .restart local v0    # "isrooted":Z
    :cond_2
    iget-object v1, p0, Lcom/android/insecurebankv2/PostLogin;->root_status:Landroid/widget/TextView;
    const-string v2, "Device not Rooted!!"

    invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

    goto :goto_1
.end method

```

- Giải thích code:

- **const/4 v1, 0x1** Gán giá trị 1 vào biến v1, dùng để đánh dấu trạng thái "đã root"
- kiểm tra sự tồn tại của tệp **Superuser.apk** tại đường dẫn được cung cấp. Nếu tệp này tồn tại, hàm sẽ trả về true, ngược lại sẽ trả về false, lưu kết quả kiểm tra vào biến v2.
- **if-nez v2, :cond_0** Nếu v2 khác 0 (có nghĩa là tệp Superuser.apk tồn tại), nhảy đến nhãn cond_0.
- Nếu tệp **Superuser.apk** không tồn tại thì sẽ tiếp tục gọi hàm **doesSUexist** để kiểm tra lệnh su có tồn tại trên hệ thống hay không, lưu kết quả kiểm tra vào v2.

- **if-eqz v2, :cond_1** Nếu v2 bằng 0 (không tìm thấy su), nhảy đến cond_1. Nếu v2 khác 0 (tìm thấy su), tiếp tục đến cond_0.
- **:cond_0** Nếu phát hiện dấu hiệu root (tồn tại tệp Superuser.apk hoặc lệnh su), gán v0 = v1 (đặt isrooted là true).
- **:cond_1** Nếu không phát hiện dấu hiệu root, const/4 v0, 0x0 (đặt isrooted là false).

- **if-ne v0, v1, :cond_2** Nếu isrooted là true, thực hiện hiển thị trạng thái thiết bị đã root và ngược lại nhảy đến **:cond_2** Nếu isrooted là false, hiển thị thông báo "Device not Rooted!!"

- Thực hiện patch để luôn hiện "rooted device" bằng cách chỉnh sửa lệnh kiểm tra điều kiện để luôn nhảy tới "Rooted Device"

- Thay **if-nez v2, :cond_0** bằng **goto :cond_0** để mã sẽ luôn nhảy đến nhãn **:cond_0** nơi isrooted được đặt thành true.

- Thay **if-ne v0, v1, :cond_2** bằng **goto :goto_0** để luôn chuyển tới phần "Rooted Device!!".

- Thực hiện tạo tệp apk đã patch

```
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ apktool b InsecureBankv2 InsecureBankv3.apk
I: Using Apktool 2.10.0 with 2 thread(s).
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
W: Unknown file type, ignoring: InsecureBankv2/smali/com/android/insecurebankv2/PostLogin.smali.bak
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: InsecureBankv2/dist/InsecureBankv2.apk
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ ls
```

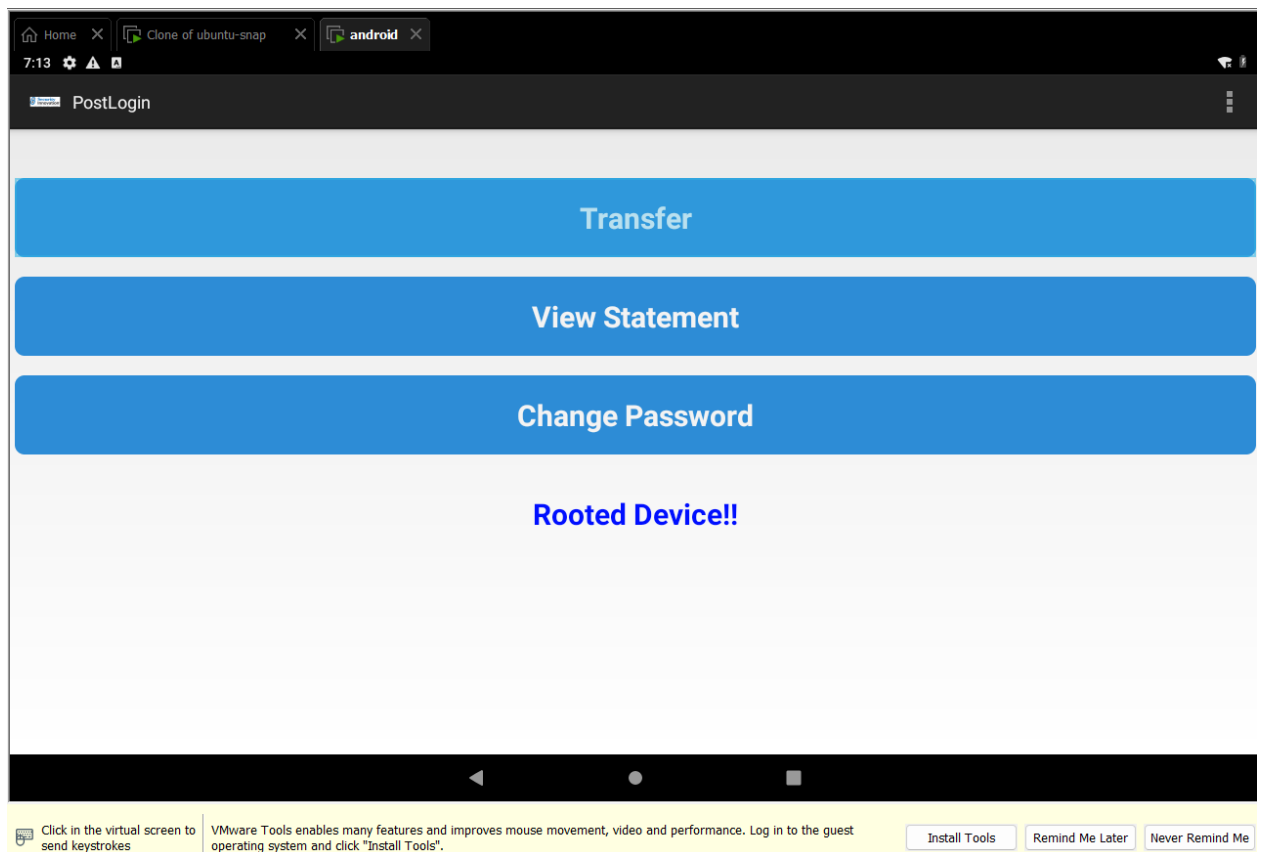
- Kí tệp

```
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ apksigner sign --ks my-release-key.jks --out InsecureBankv3-signed.apk InsecureBankv3.apk
Keystore password for signer #1:
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ ls
```

- Gỡ cài đặt bản cũ và cài bản mới

```
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ adb uninstall com.android.insecurebankv2
Success
ubuntu1@ubuntu1-virtual-machine:~/Desktop/lab4/insecurebank$ adb install InsecureBankv3-signed.apk
Performing Streamed Install
Success
```

- Kết quả đăng nhập:



Yêu cầu 7: Hoàn thiện đoạn code trên và demo.