Phòng thí nghiệm An toàn Thông tin

Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Kỳ báo cáo: Buổi 04 (Session 04)

Tên chủ đề: Pentesting Android Application

GVHD: Ngô Đức Hoàng Sơn

Ngày báo cáo: 3/5/2024

Nhóm: 09

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.O22.ATCL.2

STT	Họ và tên	MSSV	Email
1	Hồ Ngọc Thiện	21522620	21522620@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 01	100%
2	Kịch bản 02	100%
3	Kịch bản 03	100%
4	Kịch bản 04	100%
5	Kịch bản 05	100%
6	Kịch bản 06	100%
7	Kịch bản 07	100%

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

Sau khi tiến hành phân tích file InsecureBankv2.apk bằng ByteCode Viewer và scan với plugin "Malicious Code Scanner", ta nhận thấy đoạn code bất thường ở file "**com/android/insecurebankv2/DoLogin$RequestTask.class**"

```
public void postData(String var1) throws ClientProtocolException, IOException, JSONException, InvalidKeyException, NoSuchAlgorithmException
   DefaultHttpClient var2 = new DefaultHttpClient();
   HttpPost var4 = new HttpPost(this.this$0.protocol + this.this$0.serverip + ":" + this.this$0.serverport + "/login");
   HttpPost var3 = new HttpPost(this.this$0.protocol + this.this$0.serverip + ":" + this.this$0.serverport + "/devlogin");
   ArrayList var5 = new ArrayList(2);
   var5.add(new BasicNameValuePair("username", this.this$0.username));
   var5.add(new BasicNameValuePair("password", this.this$0.password));
   HttpResponse var6;
   if (this.this$0.username.equals("devadmin")) {
      var3.setEntity(new UrlEncodedFormEntity(var5));
      var6 = var2.execute(var3);
   } else {
      var4.setEntity(new UrlEncodedFormEntity(var5));
      var6 = var2.execute(var4);
   }

   InputStream var7 = var6.getEntity().getContent();
   this.this$0.result = this.convertStreamToString(var7);
   this.this$0.result = this.this$0.result.replace("\n", "");
   if (this.this$0.result != null) {
      Intent var8;
      if (this.this$0.result.indexOf("Correct Credentials") != -1) {
         Log.d("Successful Login:", ", account=" + this.this$0.username + ":" + this.this$0.password);
         this.saveCreds(this.this$0.username, this.this$0.password);
         this.trackUserLogins();
         var8 = new Intent(this.this$0.getApplicationContext(), PostLogin.class);
         var8.putExtra("uname", this.this$0.username);
         this.this$0.startActivity(var8);
      } else {
         var8 = new Intent(this.this$0.getApplicationContext(), WrongLogin.class);
         this.this$0.startActivity(var8);
      }
   }
```

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

Điểm bất thường đầu tiên ta tìm được là đoạn code trên sử dụng giao thức Http, điều đó khiến cho các dữ liệu trao đổi giữa các bên có thể bị attacker chiếm đoạt bất cứ lúc nào nhằm mục đích sửa chữa và khai thác. Hơn nữa, cả trường password và username đều không được mã hóa mà truyền thẳng vào đối tượng HttpPost (**var3** và **var4**), sau đó gửi HttpPost. Việc không mã hóa này gây ra lỗi bảo mật nghiêm trọng, attacker có thể đứng giữa, bắt gói tin và chỉnh sửa gói tin, khiến cho các dữ liệu được lưu ở server có thể bị leak ra một cách dễ dàng.

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

Sau khi gõ lệnh adb shell để vào được máy ảo, ta tiến hành vào đường dẫn **"/data/data/com.android.insecurebankv2/databases"** và truy cập vào database **"mydb"**

```
vbox86p:/ # cd /data/data/com.android.insecurebankv2/databases
vbox86p:/data/data/com.android.insecurebankv2/databases # ls
mydb  mydb-journal
vbox86p:/data/data/com.android.insecurebankv2/databases # sqlite3 mydb
SQLite version 3.32.2 2021-07-12 15:00:17
Enter ".help" for usage hints.
```

Sau đó ta thực hiện truy xuất tất cả các bảng trong mydb bằng lệnh **SELECT name FROM sqlite_master WHERE type='table';.** Sau đó truy xuất tất cả dữ liệu của tất cả các bảng.

```
sqlite> SELECT name FROM sqlite_master WHERE type='table';
android_metadata
names
sqlite_sequence
sqlite> select* from android_metadata;
en_US
sqlite> select* from names;
1|dinesh
2|dinesh
3|dinesh
4|jack
sqlite> select* from sqlite_sequence;
names|4
sqlite>
```

Ta thấy thông tin về names đã bị lộ, điều đó chứng tỏ dữ liệu lưu trữ ở đây không an toàn khi người ngoài có thể truy xuất 1 cách dễ dàng.

> **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…

Sau khi thực hiện tìm kiếm các từ khóa nhạy cảm thì ta không phát hiện được bát kì thông tin nào hữu ích.

```
1|vbox86p:/ # grep -r userID $(find)
find: ./proc/2/task/2/exe: No such file or directory
find: ./proc/2/exe: No such file or directory
find: ./proc/3/task/3/exe: No such file or directory
find: ./proc/3/exe: No such file or directory
find: ./proc/4/task/4/exe: No such file or directory
find: ./proc/4/exe: No such file or directory
find: ./proc/5/task/5/exe: No such file or directory
find: ./proc/5/exe: No such file or directory
find: ./proc/7/task/7/exe: No such file or directory
find: ./proc/7/exe: No such file or directory
find: ./proc/9/task/9/exe: No such file or directory
find: ./proc/9/exe: No such file or directory
find: ./proc/11/task/11/exe: No such file or directory
find: ./proc/11/exe: No such file or directory
find: ./proc/12/task/12/exe: No such file or directory
find: ./proc/12/exe: No such file or directory
find: ./proc/13/task/13/exe: No such file or directory
find: ./proc/13/exe: No such file or directory
find: ./proc/14/task/14/exe: No such file or directory
find: ./proc/14/exe: No such file or directory
find: ./proc/15/task/15/exe: No such file or directory
find: ./proc/15/exe: No such file or directory
find: ./proc/16/task/16/exe: No such file or directory
```

```
126|vbox86p:/ # grep -r deviceSerialNumber $(find)
find: ./proc/2/task/2/exe: No such file or directory
find: ./proc/2/exe: No such file or directory
find: ./proc/3/task/3/exe: No such file or directory
find: ./proc/3/exe: No such file or directory
find: ./proc/4/task/4/exe: No such file or directory
find: ./proc/4/exe: No such file or directory
find: ./proc/5/task/5/exe: No such file or directory
find: ./proc/5/exe: No such file or directory
find: ./proc/7/task/7/exe: No such file or directory
find: ./proc/7/exe: No such file or directory
find: ./proc/9/task/9/exe: No such file or directory
find: ./proc/9/exe: No such file or directory
find: ./proc/11/task/11/exe: No such file or directory
find: ./proc/11/exe: No such file or directory
find: ./proc/12/task/12/exe: No such file or directory
find: ./proc/12/exe: No such file or directory
find: ./proc/13/task/13/exe: No such file or directory
find: ./proc/13/exe: No such file or directory
find: ./proc/14/task/14/exe: No such file or directory
find: ./proc/14/exe: No such file or directory
find: ./proc/15/task/15/exe: No such file or directory
find: ./proc/15/exe: No such file or directory
find: ./proc/16/task/16/exe: No such file or directory
find: ./proc/16/exe: No such file or directory
find: ./proc/17/task/17/exe: No such file or directory
```

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

Sau khi giải nén ta thu được các tập tin như sau:

```
┌──(root㉿kali)-[/home/kali]
└─# mv backup.ab backup.tar

┌──(root㉿kali)-[/home/kali]
└─# dd if=backup.tar bs=24 skip=1 | zlib-flate -uncompress > extracted.tar
119644+1 records in
119644+1 records out
2871472 bytes (2.9 MB, 2.7 MiB) copied, 0.103298 s, 27.8 MB/s

┌──(root㉿kali)-[/home/kali]
└─# tar -xvf extracted.tar
apps/com.android.insecurebankv2/_manifest
apps/com.android.insecurebankv2/a/base.apk
apps/com.android.insecurebankv2/db/mydb
apps/com.android.insecurebankv2/db/mydb-journal
apps/com.android.insecurebankv2/sp/com.android.insecurebankv2_preferences.xml
apps/com.android.insecurebankv2/sp/mySharedPreferences.xml
shared/0/Documents
shared/0/Recordings
shared/0/Notifications
shared/0/DCIM
shared/0/Download
shared/0/Alarms
shared/0/Audiobooks
shared/0/Music
shared/0/Music/.thumbnails
shared/0/Music/.thumbnails/.nomedia
shared/0/Music/.thumbnails/.database_uuid
shared/0/Movies
```

Các tập tin chúng ta cần mã hóa là:

- apps/com.android.insecurebankv2/sp. Tập tin bao gồm 2 file quan trọng sau:

```
┌──(root㉿kali)-[/home/…/extracted/apps/com.android.insecurebankv2/sp]
└─# cat com.android.insecurebankv2_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="serverport">8888</string>
    <string name="serverip">192.168.1.16</string>
</map>

┌──(root㉿kali)-[/home/…/extracted/apps/com.android.insecurebankv2/sp]
└─# cat mySharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="superSecurePassword">u734blSGyPt7eobqiWxFOg==&#10;    </string>
    <string name="EncryptedUsername">amFjaw==&#13;&#10;    </string>
</map>
```

- apps/com.android.insecurebankv2/db

```
┌──(root㉿kali)-[/home/…/extracted/apps/com.android.insecurebankv2/db]
└─# l
mydb  mydb-journal

┌──(root㉿kali)-[/home/…/extracted/apps/com.android.insecurebankv2/db]
└─# sqlite3 mydb
SQLite version 3.45.1 2024-01-30 16:01:20
Enter ".help" for usage hints.
sqlite> select name from sqlite_master where type='table';
android_metadata
names
sqlite_sequence
```

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

Như ta có thể thấy, 2 chuỗi cần giải mã đang được mã hóa ở định dạng base64, ta thử decode base64 để xem có ra được chuỗi cần tìm hay không.

```
┌──(root㉿kali)-[/home/…/extracted/apps/com.android.insecurebankv2/sp]
└─# cat mySharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="superSecurePassword">u734blSGyPt7eobqiWxFOg==&#10;    </string>
    <string name="EncryptedUsername">amFjaw==&#13;&#10;    </string>
</map>
```

Ta sử dụng đoạn code này để decode

```python
import base64
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad

base64_username = "amFjaw==&#13;&#10;    "
base64_passwd = "u734blSGyPt7eobqiWxFOg==&#10;    "

#Base64 decode
username_bytes = base64.b64decode(base64_username)
passwd_bytes = base64.b64decode(base64_passwd)

print(username_bytes)
print(passwd_bytes)
```

Kết quả thu được như sau:

```
PS C:\adb tool\platform-tools> python3 .\decryptorv1.py
b'jack'
b'\xbb\xbd\xf8nT\x86\xc8\xfb{z\x86\xea\x89lE:'
PS C:\adb tool\platform-tools>
```

Ta đã tìm được username, tuy nhiên password vẫn chưa tìm được. Ta thử giải mã password bằng AES256. Update đoạn code như sau:

```python
import base64
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad

base64_username = "amFjaw==&#13;&#10;    "
base64_passwd = "u734blSGyPt7eobqiWxFOg==&#10;    "

#Base64 decode
username_bytes = base64.b64decode(base64_username)
passwd_bytes = base64.b64decode(base64_passwd)


iv_bytes = bytes([0] * 16)  # 16 bytes of 0s for initialization vector
key = "This is the super secret key 123"


def aes256_decrypt(cipher_data, key, iv_bytes):
    cipher = AES.new(key.encode(), AES.MODE_CBC, iv_bytes)
    plain_text = unpad(cipher.decrypt(cipher_data), AES.block_size, style='pkcs7')
    return plain_text.decode()


print("EncryptedUsername:", username_bytes.decode('utf-8'))

plain_text = aes256_decrypt(passwd_bytes, key, iv_bytes)
print("superSecurePassword:", plain_text)
```

Kết quả thu được:

```
PS C:\adb tool\platform-tools> python3 .\decryptorv1.py
EncryptedUsername: jack
superSecurePassword: 1234
PS C:\adb tool\platform-tools>
```

> **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.

Sử dụng Bytecode Viewer để tiến hành xem đoạn code ở file **/com/android/insecurebankv2/PostLogin.class.** Ở file này, ta thấy được hàm **showRootStatus()** hiển thị "RootDevice" hoặc "Device not Rooted!!"

```
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!!");
   }

}
```

Theo yêu cầu đề bài, ta cần khiến cho điện thoại luôn ở trạng thái "Root Device!!", vậy ta phải thỏa điều kiện của 2 hàm **doesSuperuserApkExist()** và **doesSUexist().** Cùng xem code của 2 hàm này.

```
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;
}
```

Ta thấy hàm **doesSUexist()** không thể compile thành java được, tuy nhiên ta có thể xem code của hàm này dưới dạng ngôn ngữ Smalli. Còn hàm **doesSuperuserApkExist()** thì kiểm tra xem có tồn tại file trong đường dẫn **"/system/app/Superuser.apk"** không. Ta connect vào shell và thử tìm file này.

```
vbox86p:/ # cd system/app/Superuser/
vbox86p:/system/app/Superuser # ls
Superuser.apk  oat
vbox86p:/system/app/Superuser #
```

Ta thấy file Superuser.apk thực chất chứa trong đường dẫn **"system/app/Superuser/Superuser.apk"**, vậy hàm **doesSuperuserApkExist()** đang kiểm tra sai đường dẫn, ta có thể sửa lại đường dẫn này trong tập tin Smalli đã được decompile trước đó.

```
.method private doesSuperuserApkExist(Ljava/lang/String;)Z
    .locals 4
    .param p1, "s"    # Ljava/lang/String;

    .prologue
    const/4 v2, 0x1

    .line 115
    new-instance v1, Ljava/io/File;

    const-string v3, "/system/app/Superuser/Superuser.apk"

    invoke-direct {v1, v3}, Ljava/io/File;-><init>(Ljava/lang/String;)V

    .line 116
    .local v1, "rootFile":Ljava/io/File;
    invoke-virtual {v1}, Ljava/io/File;->exists()Z
```

Ta tiếp tục xem hàm doesSUexist() xem nó hoạt động như thế nào

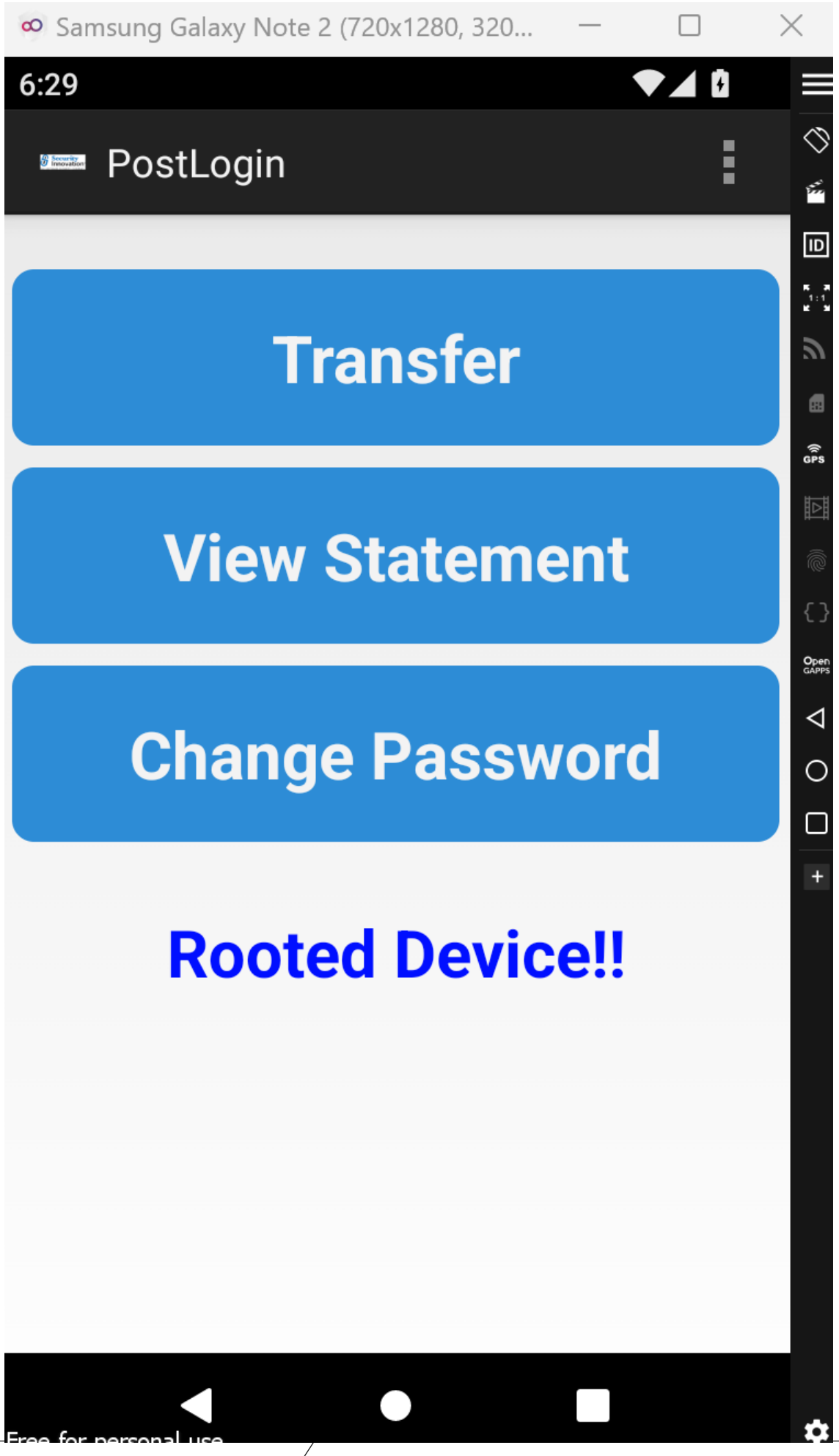
```
    const/4 v6, 0x2

    new-array v6, v6, [Ljava/lang/String;

    const/4 v7, 0x0

    const-string v8, "/system/xbin/which"

    aput-object v8, v6, v7

    const/4 v7, 0x1

    const-string v8, "su"

    aput-object v8, v6, v7

    invoke-virtual {v5, v6}, Ljava/lang/Runtime;->exec([Ljava/lang/String;)Ljava/lang/Process;
```

Ở hàm này hệ thống sẽ thực thi lệnh **/system/xbin/which su** để tìm kiếm file nhị phân su, nếu có thì trả về đường dẫn của nó, nếu tồn tại đường dẫn thì trả về true, còn không thì trả về false. Vậy ở hàm này ta không cần phải chỉnh sửa thêm gì cả.

Sau khi đã chỉnh sửa, ta tiến hành build lại file .apk và kí lên nó.

```
┌──(root㉿kali)-[/home/kali/APK]
└─# apktool b InsecureBankv2 -o RootedBypass.apk
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: RootedBypass.apk

┌──(root㉿kali)-[/home/kali/APK]
└─# keytool -genkey -v -keystore RootedBypass.keystore -alias RootedBypass -keyalg RSA -keysize 2048 -validity 10000

Enter keystore password:
Re-enter new password:
They don't match. Try again
Enter keystore password:
Re-enter new password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default
 value in braces.
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 (SHA384withRSA) with a validity of 10,000 days
        for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing RootedBypass.keystore]

┌──(root㉿kali)-[/home/kali/APK]
└─# apksigner sign --ks RootedBypass.keystore  RootedBypass.apk

Keystore password for signer #1:
```

Cài đặt lên máy ảo và xem kết quả:


Samsung Galaxy Note 2 (720x1280, 320...
6:29
PostLogin
Transfer
View Statement
Change Password
Rooted Device!!
Free for personal use

Yêu cầu 7 Hoàn thiện đoạn code trên và demo.

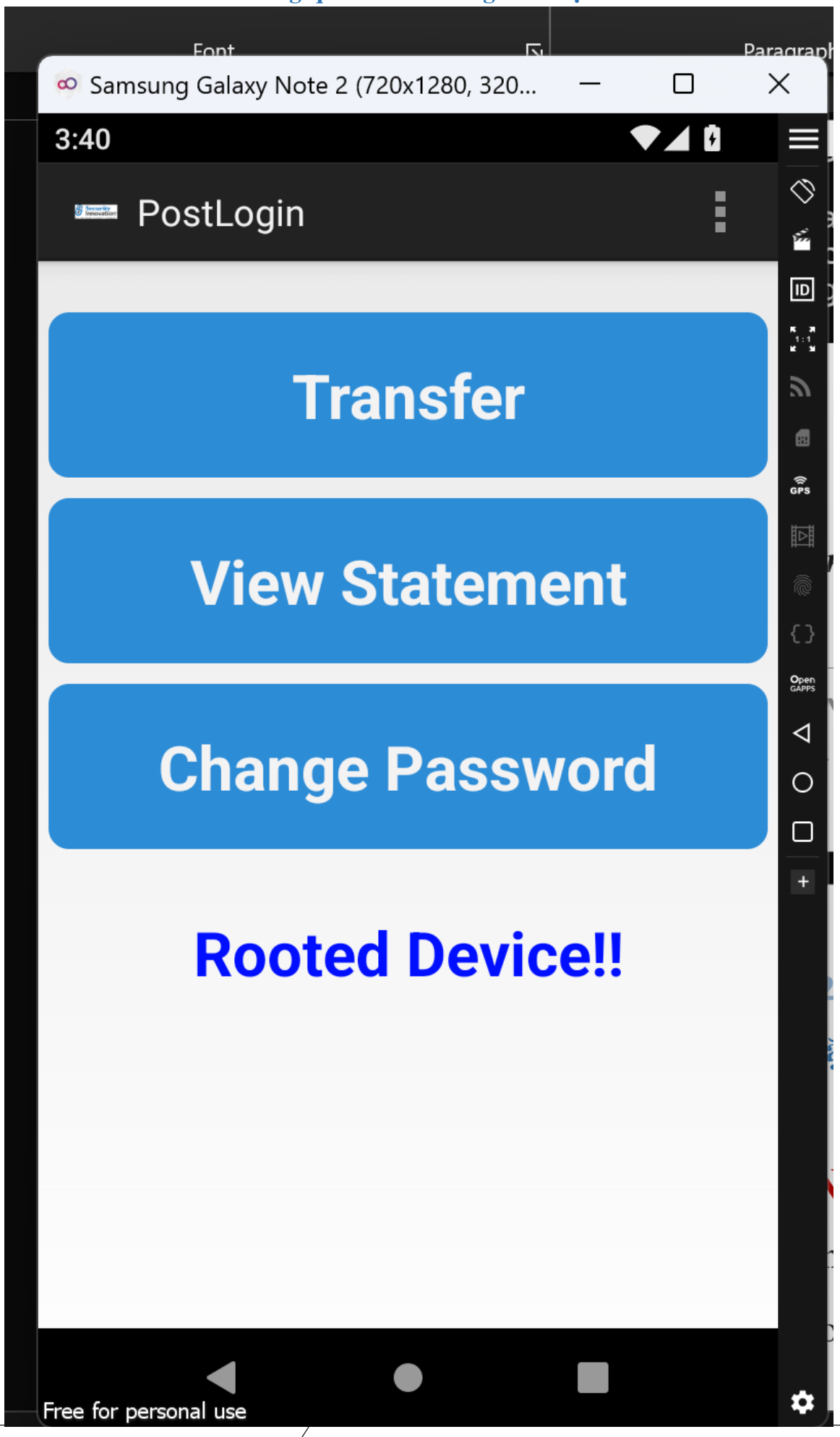
```
hook_script="""
Java.perform
(
  function()
  {
    console.log("Inside the hook_script");
    classPostLogin = Java.use('com.android.insecurebankv2.PostLogin');
    classPostLogin.doesSuperuserApkExist.implementation = function()
    {
        // làm cái gì đó trong trường hợp này, return true
    };
  }
);
"""
```

Ta nhận thấy ta có thể ghi đè hàm **doesSuperuserApkExist()** dựa vào đoạn code trên, vậy phần code ta cần hoàn thành phải luôn luôn đưa hàm này trả về giá trị true. Vậy ta thử ghi đè hàm bằng dòng "return true" xem điều gì xảy ra.

```python
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="""
Java.perform
(
    function()
    {
        console.log("Inside the hook_script");
        var classPostLogin = Java.use('com.android.insecurebankv2.PostLogin');
        if(!classPostLogin)
        {
            console.log("Null!");
        }
        classPostLogin.doesSuperuserApkExist.implementation = function()
        {
            console.log("Root is coming!")
            return true
        };
    }
);
"""
script=session.create_script(hook_script)
script.load()

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

Kết quả ta thu được:

```
PS C:\adb tool\platform-tools> python3 .\frida-decrypt.py
Inside the hook_script
...?Root is coming!
```


Samsung Galaxy Note 2 (720x1280, 320...
3:40
PostLogin
Transfer
View Statement
Change Password
Rooted Device!!
Free for personal use

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

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

Báo cáo:

- File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

  *Ví dụ: [*NT101.K11.ATCL*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài Lab.
- 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.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

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

HẾT