Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng

Kỳ báo cáo: Buổi 04 (Session 04)

Tên chủ đề: Android Pentesting

GV: Nghi Hoàng Khoa

Ngày báo cáo: 11/05/2023

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)
Lớp: NT213.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Bùi Tấn Hải Đăng	20520173	20520173@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Kịch bản 01/Câu hỏi 01	100%	Đăng
2	Kịch bản 02	100%	Đăng
3	Kịch bản 03	100%	Đăng
4	Kịch bản 04	100%	Đăng
5	Kịch bản 05	100%	Đăng
6	Kịch bản 06	100%	Đăng
7	Kịch bản 07	100%	Đăng

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

1. Kịch bản 01/Câu hỏi 01

```
public void postData(String var1) throws ClientProtocolException, IOException, JSONException, InvalidKeyException, NoSuchAlgorit
   DefaultHttpClient var3 = new DefaultHttpClient();
   HttpPost var4 = new HttpPost(this.this$0.protocol + this.this$0.serverip + ":" + this.this$0.serverport + "/login");
   HttpPost var2 = 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")) {
      var2.setEntity(new UrlEncodedFormEntity(var5));
      var6 = var3.execute(var2);
   } else {
      var4.setEntity(new UrlEncodedFormEntity(var5));
      var6 = var3.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 ở đoạn code trên là nó tiết lộ 1 route bí mật là /devlogin, cụ thể là lập trình viên so sánh username với chuỗi plaintext "devadmin". Nếu như đúng là "devadmin" thì sẽ truy vấn HTTP Post đến /devlogin và ngược lại thì dùng route /login thông thường.

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?

- Sau khi vào databases của ứng dụng bank thì thấy có 2 cơ sở dữ liệu là mydb và my-journal.
- Vào

```
sqlite> .tables
android_metadata  names
sqlite> select * from names
   ...> ;
1|dinesh
```

- Dữ liệu trong table names chứa các thông tin username được lưu trữ không an toàn do giữ nguyên plaintext mà không dùng bất kì thuật toán mã hóa hay hash nào.

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…

- Kiểm tra các thông tin nhạy cảm lưu trên thiết bị thì em thấy khi grep string " name" thì sẽ trả về nhiều thông tin nhạy cảm như serverip, serverport, username, password.

```
1|genymotion:/data/data/com.android.insecurebankv2 # grep -r "name" $(find)
Binary file ./databases/mydb matches
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverport">8888</string>
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverip">10.45.144.116</string>
./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 matches
Binary file ./databases/mydb matches
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverport">8888</string>
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverip">10.45.144.116</string>
./shared_prefs/mySharedPreferences.xml:    <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;    </string>
./shared_prefs/mySharedPreferences.xml:    <string name="EncryptedUsername">ZGluZXNo&#13;&#10;    </string>
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverport">8888</string>
./shared_prefs/com.android.insecurebankv2_preferences.xml:    <string name="serverip">10.45.144.116</string>
./shared_prefs/mySharedPreferences.xml:    <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;    </string>
./shared_prefs/mySharedPreferences.xml:    <string name="EncryptedUsername">ZGluZXNo&#13;&#10;    </string>
genymotion:/data/data/com.android.insecurebankv2 #
```

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

- Sau khi backup dữ liệu, bởi vì user đã đăng nhập với tài khoản của mình trước khi backup cho nên tại file mySharedPrefrences.xml lưu trữ thông tin username/password đã được mã hóa.

```
hidang@LAPTOP-0FG5MOP3:/tmp$ ls
InsecureBankv2.apk  apps  backup.ab  backup_compressed.tar  shared
```

```
hidang@LAPTOP-0FG5MOP3:/tmp/apps/com.android.insecurebankv2/sp$ ls
com.android.insecurebankv2_preferences.xml  mySharedPreferences.xml
```

```
hidang@LAPTOP-0FG5MOP3:/tmp/apps/com.android.insecurebankv2/sp$ cat mySharedPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="superSecurePassword">DTrW2VXjSoFdg0e61fHxJg==&#10;    </string>
    <string name="EncryptedUsername">ZGluZXNo&#13;&#10;    </string>
</map>
hidang@LAPTOP-0FG5MOP3:/tmp/apps/com.android.insecurebankv2/sp$
```

- Theo em thì những thông tin mang tính nhạy cảm như username, password đều cần được mã hóa bên trong thư mục sao lưu.

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

- Tại đường dẫn com/android/insecurebankv2/CryptoClass.class, chứa các hàm mã hoá và mã giả thấy chương trình sử dụng một số phương thức mã hóa yếu và có sẵn key.

```
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, NoSuchA
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, NoSuchA
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 aesDeccryptedString(String var1) throws UnsupportedEncodingException, InvalidKeyException, NoSuchAlgorit
40          byte[] var2 = this.key.getBytes("UTF-8");
41          this.cipherData = aes256decrypt(this.ivBytes, var2, Base64.decode(var1.getBytes("UTF-8"), 0));
```

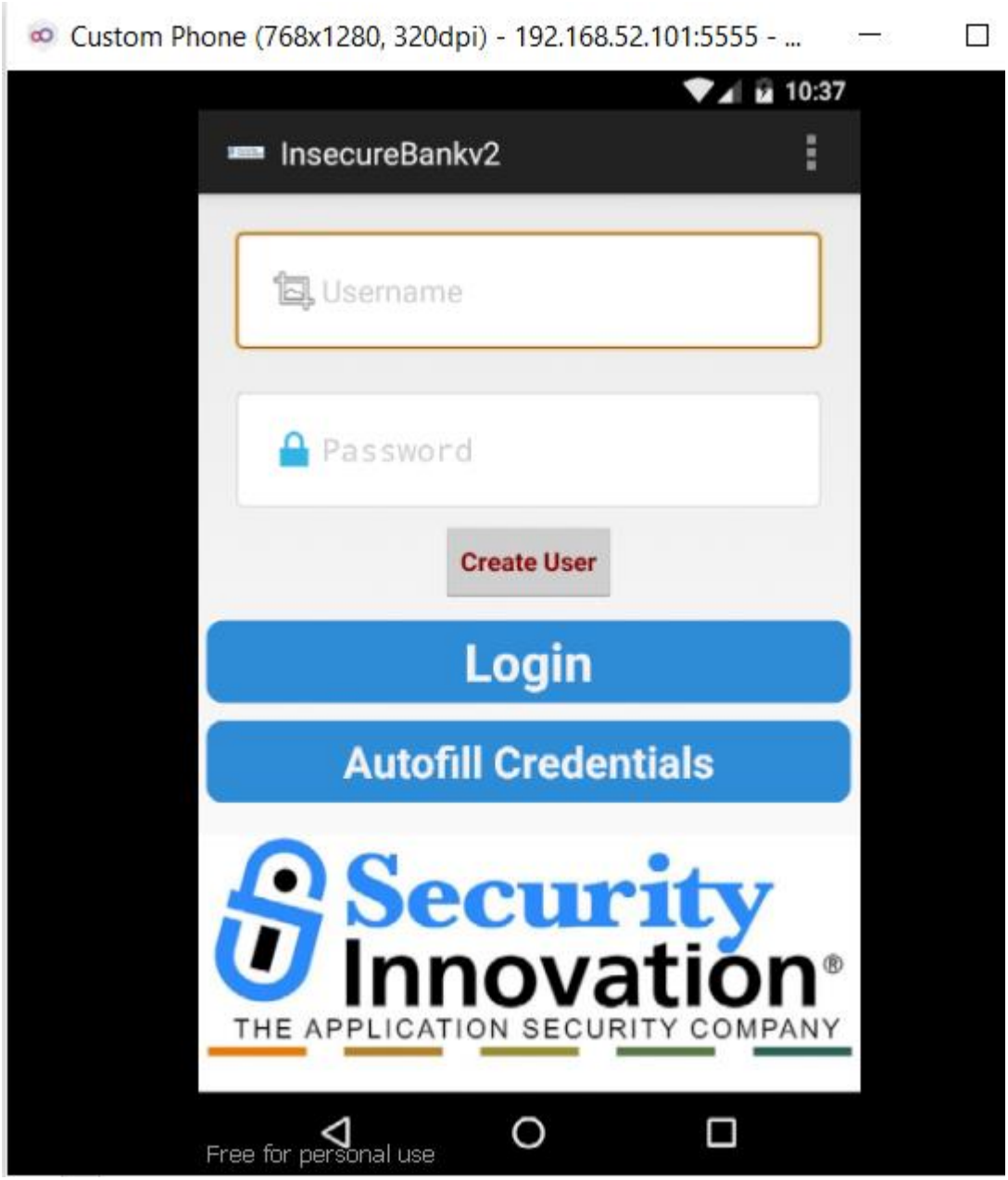
- Có thể thấy lập trình viên sử dụng thuật toán mã hóa AES với mode CBC với key là “This is the super secret key 123” và sau đó decode base64.
  ➔ Dựa vào đó attacker có thể dễ dàng giải mã với đoạn script sau:

```python
import base64
from Crypto.Cipher import AES
key = b'This is the super secret key 123'

encrypted_text = 'v/sJpihDCo2ckDmLW5Uwiw==&#10'
cipher = AES.new(key, AES.MODE_CBC, b'\x00'*16)

decoded = base64.b64decode(encrypted_text)
decrypted = cipher.decrypt(decoded)

print(decrypted)
```

```
PS C:\Users\Dan> python -u "c:\Users\Dan\OneDrive - gm.uit.edu.vn\Desktop\decrypt.py"
b'Jack@123$\x07\x07\x07\x07\x07\x07\x07'
PS C:\Users\Dan>
```

- Sau khi giải mã đoạn password trên ta được pass của user là Jack@123
- **Activity Hijacking**

```
genymotion:/ # am start -n com.android.insecurebankv2/.PostLogin
Starting: Intent { cmp=com.android.insecurebankv2/.PostLogin }
genymotion:/ #
```



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.

- Sau khi decompile file InsecureBank ta không có mã nguồn nên phải đọc đoạn smalli để thay đổi luồng thực thi của chương trình.

```
.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;
```

- Đọc trong showRoot Status(), tại .line86 sẽ kiểm tra xem có tồn tại file /system/app/Superuser.apk hay không và gán kết quả vào v2, nếu = 1 nghĩa sẽ nhảy đến label :cond_0 chạy dòng lệnh bên dưới label này để in ra dòng "isrooted". Nếu v2 = 0 thì nghĩa là không tồn tại file Superuser.apk, thì nó sẽ kiểm tra tiếp tới doesSUexist(), kết quả gán vào v2, nếu như v2=0 thì nó sẽ nhảy đến cond_1. Label này để in ra "not rooted".
- Vậy để in ra "is rooted" trong mọi hoàn cảnh thì ta có thể xóa dòng được khoanh trong ô màu đỏ "if -eqz v2, :cond_1" hoặc đổi lệnh đó thành "if -eqz v2, :cond_0" để nó không nhảy đến cond_1 mà chạy thẳng xuống dòng bên dưới là in ra "is rooted".

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

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

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

Yêu cầu 7 Hoàn thiện đoạn code trên và demo.

- Đoạn code hoàn chỉnh.
- Giải thích code:
  - Ý tưởng vẫn giống như yêu cầu 6, em sẽ hook hàm doesSuperuserApkexist để làm cho giá trị trả về của nó bằng true. Em sẽ sử dụng Java.use() để ghi đè nội dung của phương thức này. Có như vậy thì sẽ trả về rooted trong mọi trường hợp.
  - Trong đoạn hook_script sẽ hook đến doesSuperuserApkexist và ghi đè nó với lệnh return true. Lúc này giá trị trả về của hàm này đều luôn bằng true thì ứng dụng sẽ luôn hiển thị "is Rooted".

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

        var doesSuperuserApkExist = PostLogin.doesSuperuserApkExist;
        doesSuperuserApkExist.implementation = function (s) {
            console.log("Hooking Success");
            return true;
        };
  });
"""
script=session.create_script(hook_script)
script.load()
input('...?') # prevent terminate
```

HẾT