



BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng
Lab 4: Pentesting Android Applications

Ngày báo cáo: 08/05/2024

1. THÔNG TIN CHUNG:

Lớp: NT213.O22.ATCL.2

STT	Họ và tên	MSSV	Email
1	Hoàng Gia Bảo	21521848	21521848@gm.uit.edu.vn
2	Phạm Hoàng Phúc	21521295	21521295@gm.uit.edu.vn
3	Nguyễn Đại Nghĩa	21521182	21521182@gm.uit.edu.vn

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

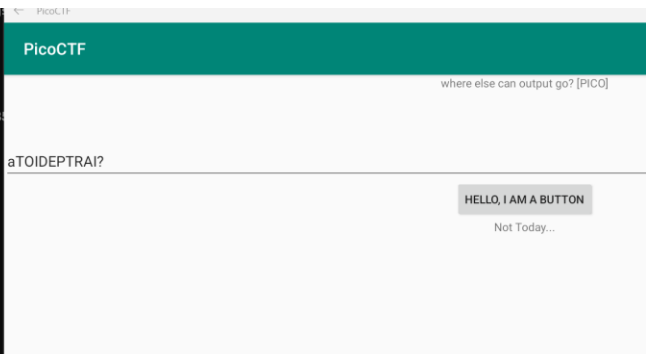
BÁO CÁO CHI TIẾT

Bài tập CTF:

D.2 Droid

FLAG1

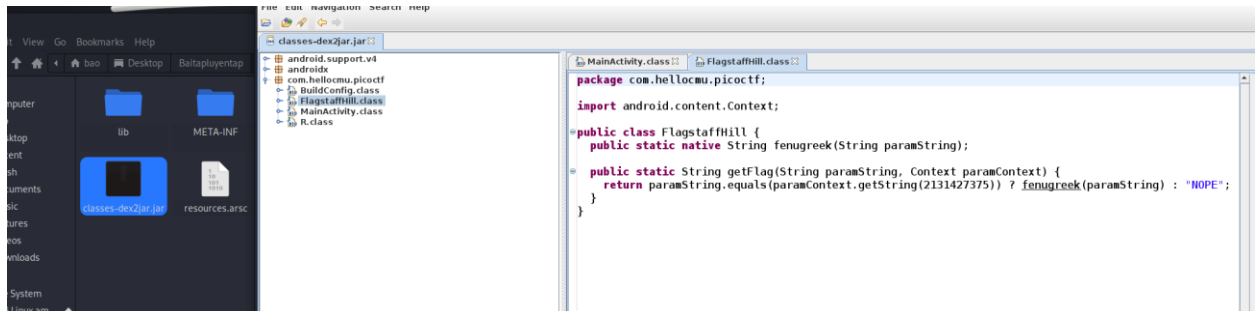
```
one=KEYCODE_ALT_LEFT, scanCode=96, metaState=META_ALT_ON|META_ALT_LEFT_ON, flags=30710, downTime=2647102000000, deviceId=-1, source=0x101, displayId=-1 }
^C
C:\Users\BaoBao>adb -s 127.0.0.1:58526 logcat | grep picoCTF
'grep' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\BaoBao>adb -s 127.0.0.1:58526 logcat | grep picoCTFadb -s 127.0.0.1:58
C:\Users\BaoBao>adb -s 127.0.0.1:58526 logcat | Select-String "picoCTF"
'Select-String' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\BaoBao>adb -s 127.0.0.1:58526 logcat | findstr "picoCTF"
05-08 17:41:28.856 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:42:46.548 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:42:49.388 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:43:03.052 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:43:03.366 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:43:03.532 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:43:03.682 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
05-08 17:43:03.830 3635 3635 I PICO : picoCTF{a.moose.once.bit.my.sister}
```



Mở one.apk lên ta được PicoCTF, ở đây ta thử nhập bất kì sau đó dung logcat của adb để tìm các keyword có liên quan đến CTF

Flag : **picoCTF{a.moose.once.bit.my.sister}**

FLAG2



Sau khi dùng jd-gui để kiểm tra file (file classes.dex đã được chuyển đổi thành .jar) ta thấy có 1 file tên flag nằm ở .picoCTF

Tiếp tục sử dụng apktool, ta tìm đến đường dẫn có chứa file flagstaffHill và đọc nó

```

1.class public Lcom/hellocmu/picocft/FlagstaffHill;
2.super Ljava/lang/Object;
3.source "FlagstaffHill.java"
4
5
6.# direct methods
7.method public constructor <init>()V
8    .locals 0
9
10   .line 6
11   invoke-direct {p0}, Ljava/lang/Object;→<init>()V
12
13   return-void
14 .end method
15
16.method public static native fenugreek(Ljava/lang/String;)Ljava/lang/String;
17 .end method
18
19.method public static getFlag(Ljava/lang/String;Landroid/content/Context;)Ljava/lang/String;
20 .locals 2
21 .param p0, "input"    # Ljava/lang/String;
22 .param p1, "ctx"      # Landroid/content/Context;
23
24 .line 11
25   const v0, 0x7f0b002f

```

Ở đây ta thấy có 1 địa chỉ là **0x7f0b002f**, kết hợp với code trước đó xem ở jd-gui thì có vẻ nó có liên quan tới **paramContext**

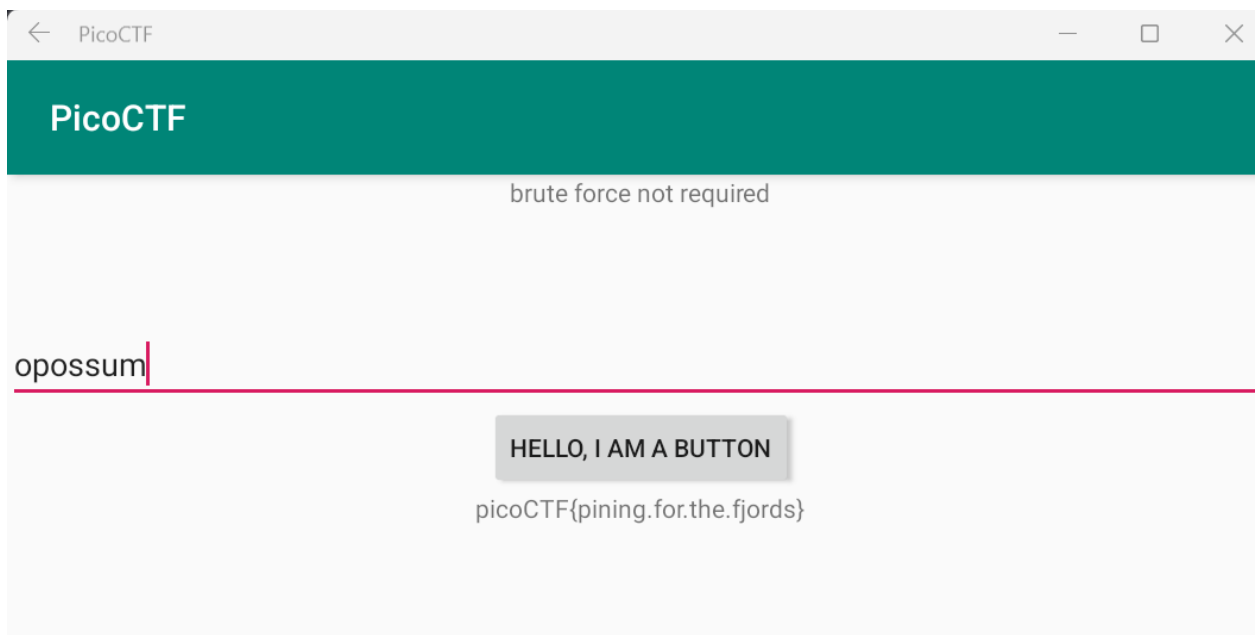
Dòng `public static String getFlag(String paramString, Context paramContext)` { : Đây là khai báo của phương thức `getFlag`, một phương thức tĩnh nhận một chuỗi (`paramString`) và một đối tượng `Context` (`paramContext`) làm tham số.

Dòng `return paramString.equals(paramContext.getString(2131427375)) ? fenugreek(paramString) : "NOPE"`

```
(bao@kali)-[~/Desktop/Baitapluientap/two]
$ grep -r "0x7f0b002f"
smali/com/hellocmu/picoctf/R$string.smali:.field public static final password:I = 0x7f0b002f
smali/com/hellocmu/picoctf/FlagstaffHill.smali:    const v0, 0x7f0b002f
res/values/public.xml:    <public type="string" name="password" id="0x7f0b002f" />

(bao@kali)-[~/Desktop/Baitapluientap/two]
$ grep -r "password"
smali/androidx/core/view/accessibility/AccessibilityNodeInfoCompat.smali:    .param p1, "password"    # Z
smali/androidx/core/view/accessibility/AccessibilityNodeInfoCompat.smali:    const-string v2, "; password: "
smali/com/hellocmu/picoctf/R$string.smali:.field public static final password:I = 0x7f0b002f
smali/com/hellocmu/picoctf/FlagstaffHill.smali:    .local v0, "password":Ljava/lang/String;
res/values/public.xml:    <public type="string" name="password" id="0x7f0b002f" />
res/values/strings.xml:    <string name="password">opossum</string>
```

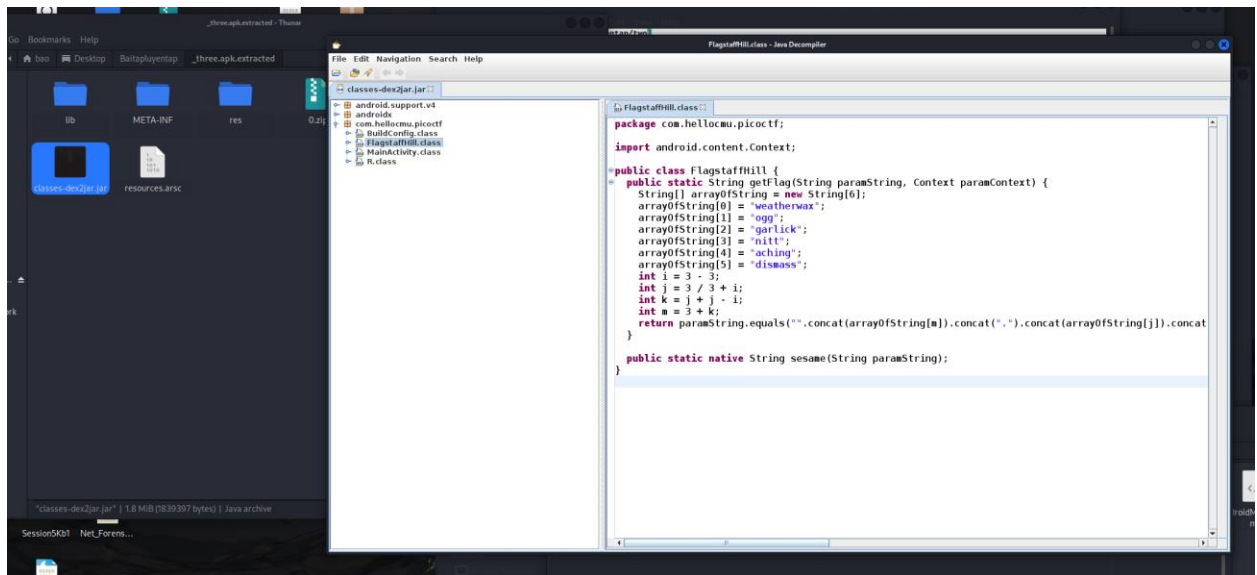
Ta tìm kiếm các file có chứa chuỗi **0x7f0b002f**, thì thấy có liên quan tới password
Tiếp tục tìm kiếm với từ khóa password thì dc pass là **opossum**



FLAG : **picoCTF{pining.for.the.fjords}**

FLAG3

Dùng jd-gui để đọc code của file **flagstaffHill**



Hãy xem xét từng phần trong chuỗi:

arrayOfString[m]: Đây là phần tử thứ m trong mảng arrayOfString.

`.concat(".")`: Chuỗi này thêm một dấu chấm "." vào sau phần tử trước.

arrayOfString[j]: Đây là phần tử thứ j trong mảng arrayOfString.

`.concat(".")`: Thêm một dấu chấm "." vào sau phần tử trước.

`arrayOfString[i]`: Đây là phần tử thứ `i` trong mảng `arrayOfString`.

`.concat(".")`: Thêm một dấu chấm "." vào sau phần tử trước.

`arrayOfString[m + i - j]`: Đây là phần tử thứ $m + i - j$ trong mảng `arrayOfString`.

`.concat(".")`: Thêm một dấu chấm "." vào sau phần tử trước.

arrayOfString[3]: Đây là phần tử thứ 3 trong mảng arrayOfString.

`.concat(".")`: Thêm một dấu chấm "." vào sau phần tử trước.

`arrayOfString[k]`: Đây là phần tử thứ k trong mảng `arrayOfString`.

Do đó, kết quả của chuỗi được tạo ra là sự kết hợp của các phần tử này, mỗi phần tử được ngăn cách bởi một dấu chấm "."

⇒ Vậy nên ta chỉ cần bỏ 2 biến đi rồi xuất thẳng ra mật khẩu :

```

Main.java 42ckvevht
NEW JAVA RUN

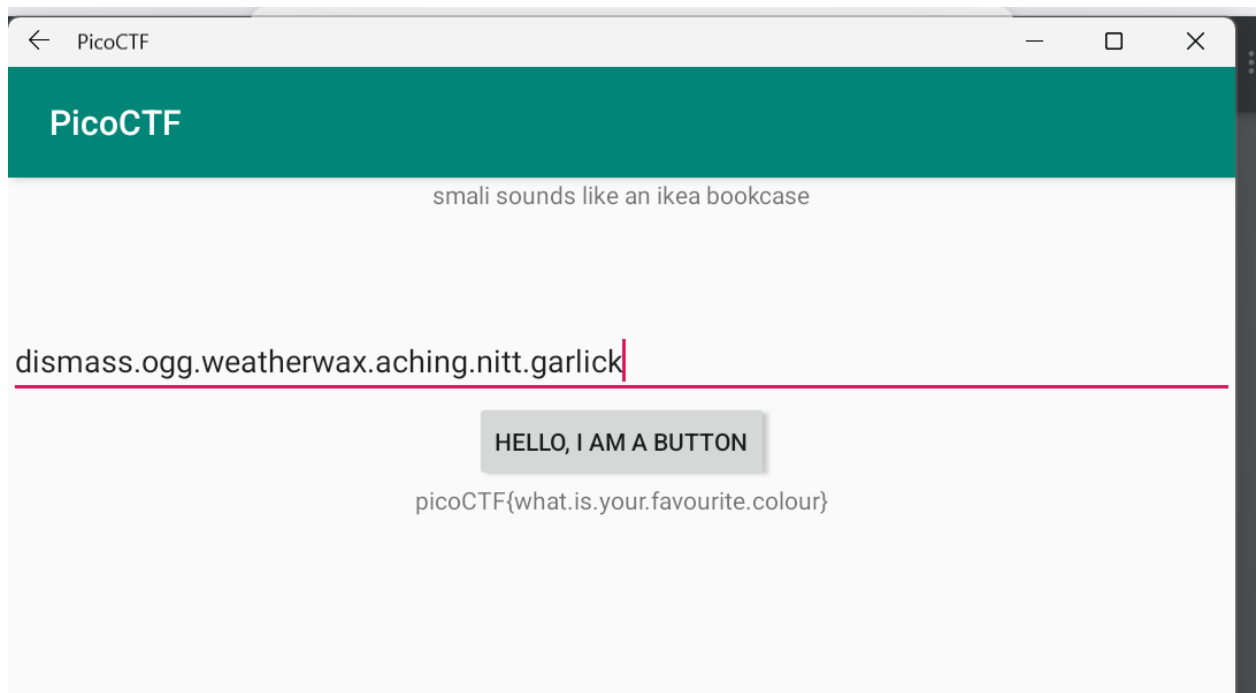
1 public class Main {
2     public static void main(String[] args) {
3         // Gọi phương thức getFlag và in ra kết quả
4         String result = getFlag();
5         System.out.println(result);
6     }
7
8     public static String getFlag() {
9         // Phần code của phương thức getFlag
10        String[] arrayOfString = new String[6];
11        arrayOfString[0] = "weatherwax";
12        arrayOfString[1] = "ogg";
13        arrayOfString[2] = "garlick";
14        arrayOfString[3] = "nitt";
15        arrayOfString[4] = "aching";
16        arrayOfString[5] = "dismiss";
17        int i = 3 - 3;
18        int j = 3 / 3 + i;
19        int k = j + j - i;
20        int m = 3 + k;
21        String result = "";
22        System.out.println(result);
23        return result;
24    }
25
26    public static native String sesame(String paramString);

```

STDIN
Input for the program (Optional)

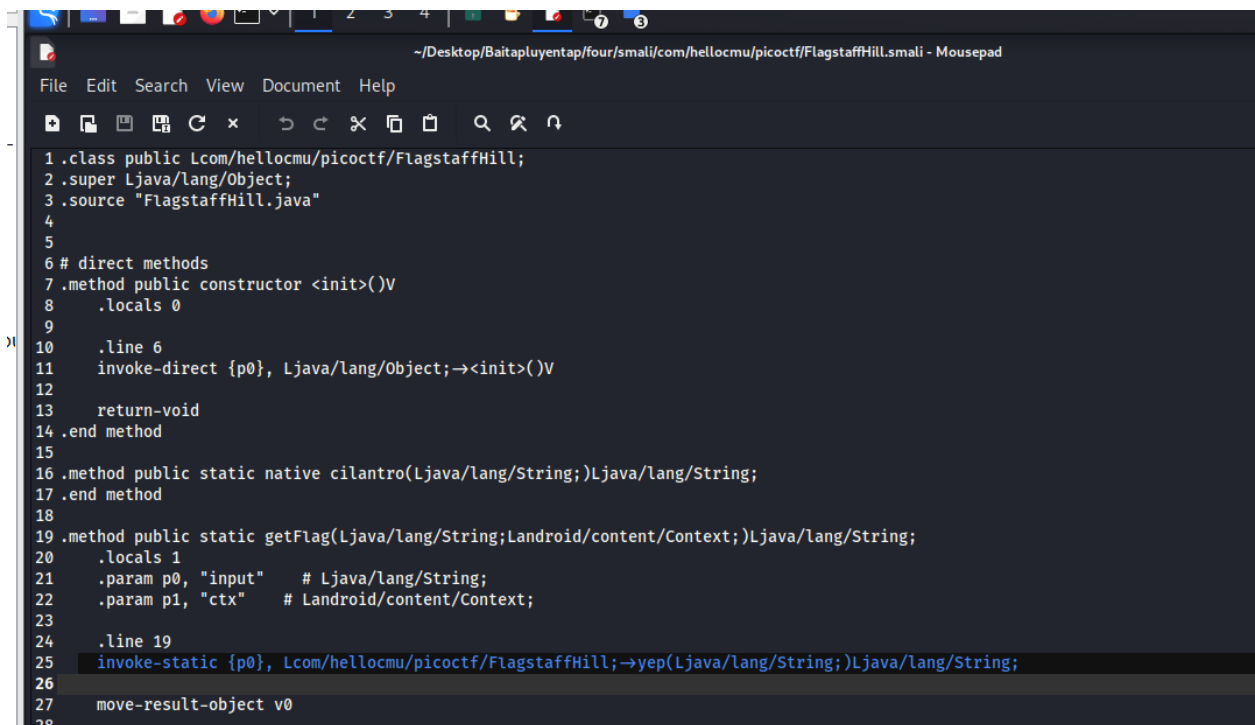
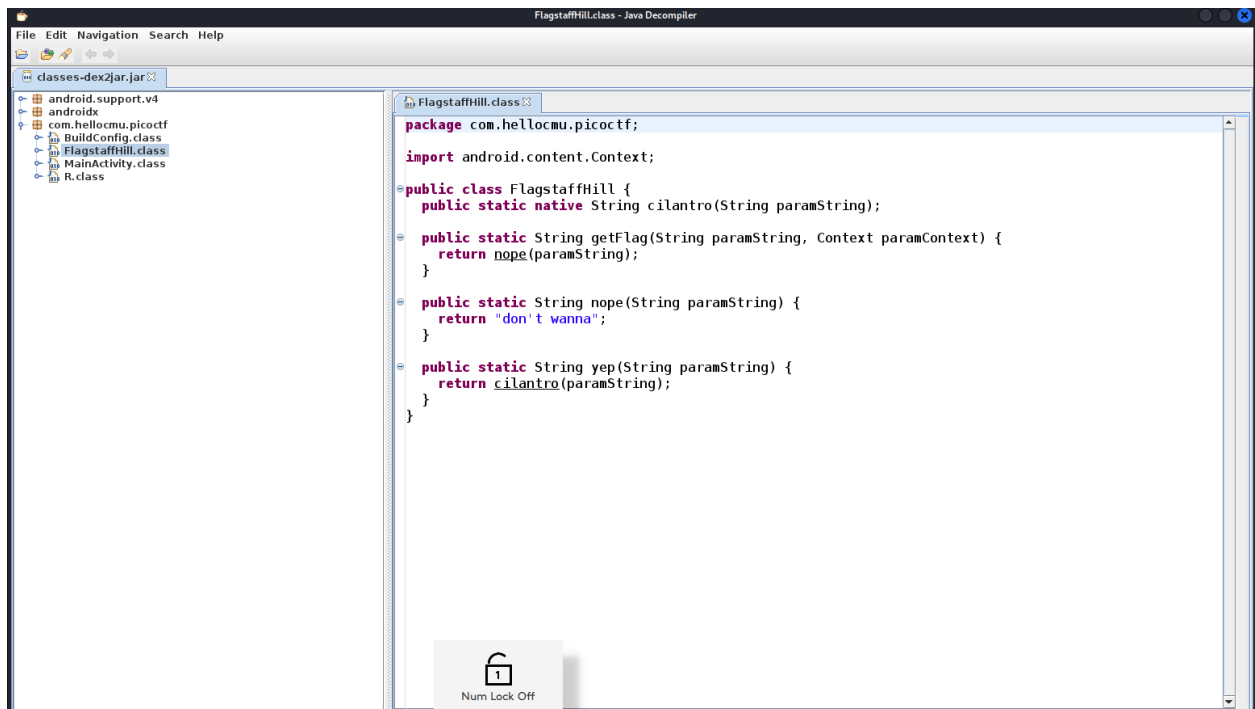
Output:
dismiss.ogg.weatherwax.aching.nitt.garlick
dismiss.ogg.weatherwax.aching.nitt.garlick

FEEDBACK



Flag: picoCTF{what.is.your.favourite.colour}

FLAG4



Sửa dòng 25 (nope -> yep)

```
(bao@kali)-[~/Desktop/Baitapluientap]
$ java -jar apktool_2.9.3.jar b NewFolder/four
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
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: NewFolder/four/dist/four.apk
```

Recompiler lại file four.apk

```
(bao@kali)-[~/Desktop/Baitapluientap]
$ keytool -genkey -v -keystore fourv2.keystore -alias fourv2 -keyalg RSA -keysize 2048 -validity 10000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: bao
What is the name of your organizational unit?
[Unknown]: vietnam
What is the name of your organization?
[Unknown]: vietnam
What is the name of your City or Locality?
[Unknown]: vietnam
What is the name of your State or Province?
[Unknown]: vietnam
What is the two-letter country code for this unit?
[Unknown]: vietnam
Is CN=bao, OU=vietnam, O=vietnam, L=vietnam, ST=vietnam, C=vietnam correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=bao, OU=vietnam, O=vietnam, L=vietnam, ST=vietnam, C=vietnam
[Storing fourv2.keystore]
```

Tạo chữ ký

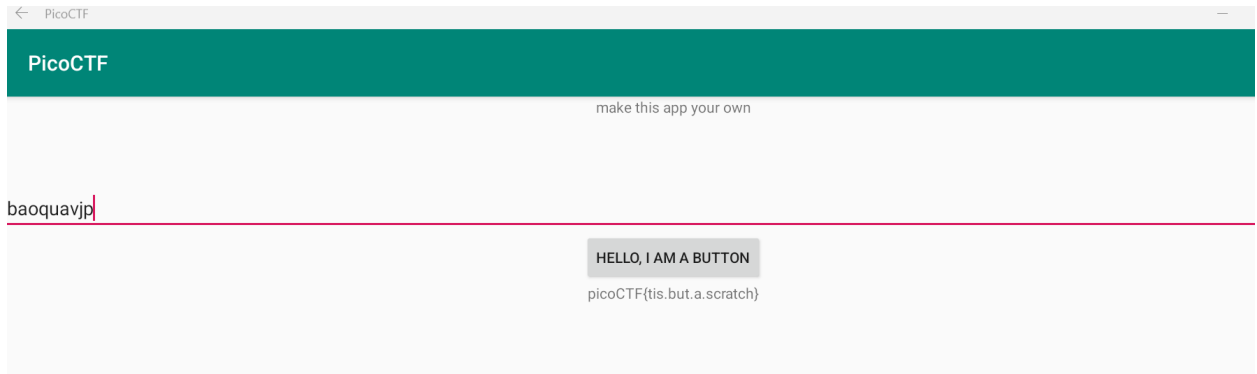
```
at com.android.apksigner.ApkSignerTool.main(ApkSignerTool.java:85)
I: Checking whether resources has changed ...
(bao@kali)-[~/Desktop/Baitapluientap]
$ apksigner sign --ks fourv2.keystore four.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:
(bao@kali)-[~/Desktop/Baitapluientap]
```

Ký lại vào file four.apk này

```
C:\Users\BaoBao>adb -s 127.0.0.1:58526 uninstall com.hellocmu.picoctf
Success
```

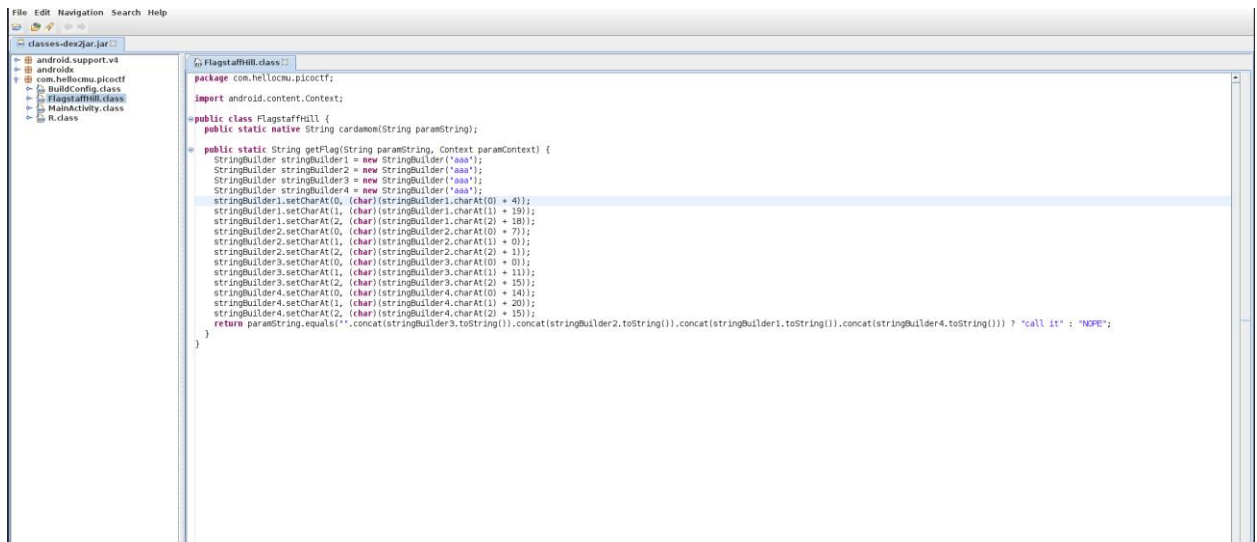
Gỡ đi chữ ký trước đó bên windows

KẾT QUẢ :



Flag: picoCTF{tis.but.a.scratch}

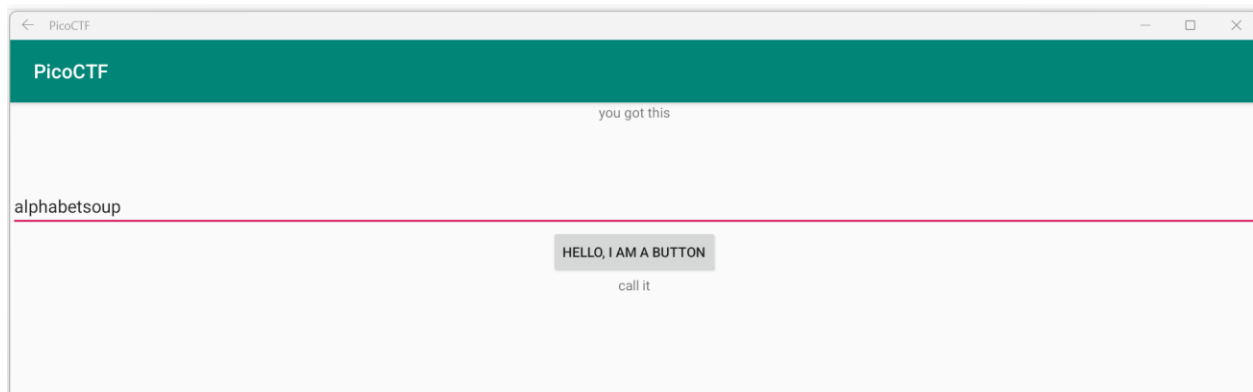
FLAG5



Sửa code để trả về kết quả :



Ta được output là : **alphabetsoup**



Không trả về flag , ta xem lại file snail của FlagStaff

```

22  .param p1, "ctx"    # Landroid/content/Context;
23
24  .line 12
25  new-instance v0, Ljava/lang/StringBuilder;
26

```

V0 là biến liên quan tới flag

```

230  move-result v5
231
232  if-eqz v5, :cond_0
233
234  const-string v5, "call it"
235
236  return-object v5
237
238  .line 37
239  :cond_0
240  const-string v5, "NOPE"
241
242  return-object v5
243 .end method

```

Ở dòng này trả về flag không đúng nên ta sẽ sửa lại

```

229
230  move-result v5
231
232  invoke-static {p0}, Lcom/helloctmu/picoctf/FlagstaffHill;→cardamom(L2java/lang/String;)Ljava/lang/String;
233
234  move-result-object v0
235
236  return-object v0
237
238  .line 37
239  :cond_0

```

Build lại file

```

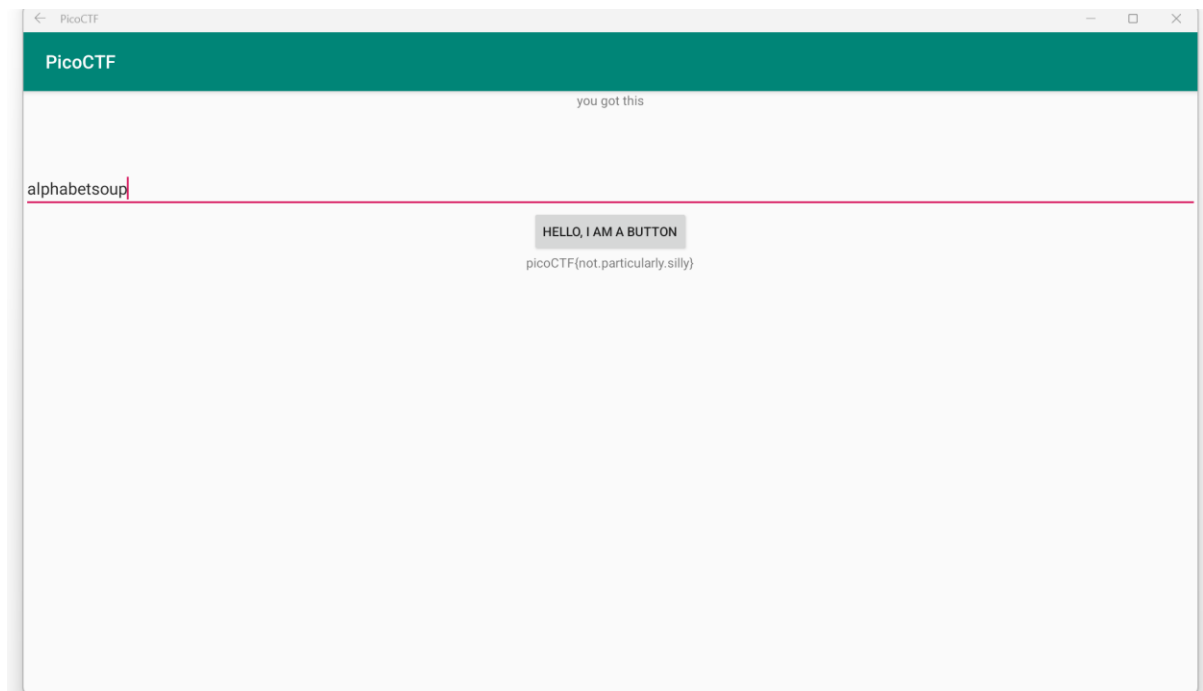
(bao@kali)-[~/Desktop/Baitapluientap]
$ java -jar apktool 2.9.3.jar b five
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: five/dist/five.apk
(bao@kali)-[~/Desktop/Baitapluientap]
$

```

Tạo lại chữ ký như ở bài trên và ký lại vào file apk mới

```
➜ $ cd ..  
  
(bao@kali)-[~/Desktop/Baitapluventap]  
$ apksigner sign --ks fivev2.keystore five2.apk  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
Keystore password for signer #1:  
  
(bao@kali)-[~/Desktop/Baitapluventap]
```

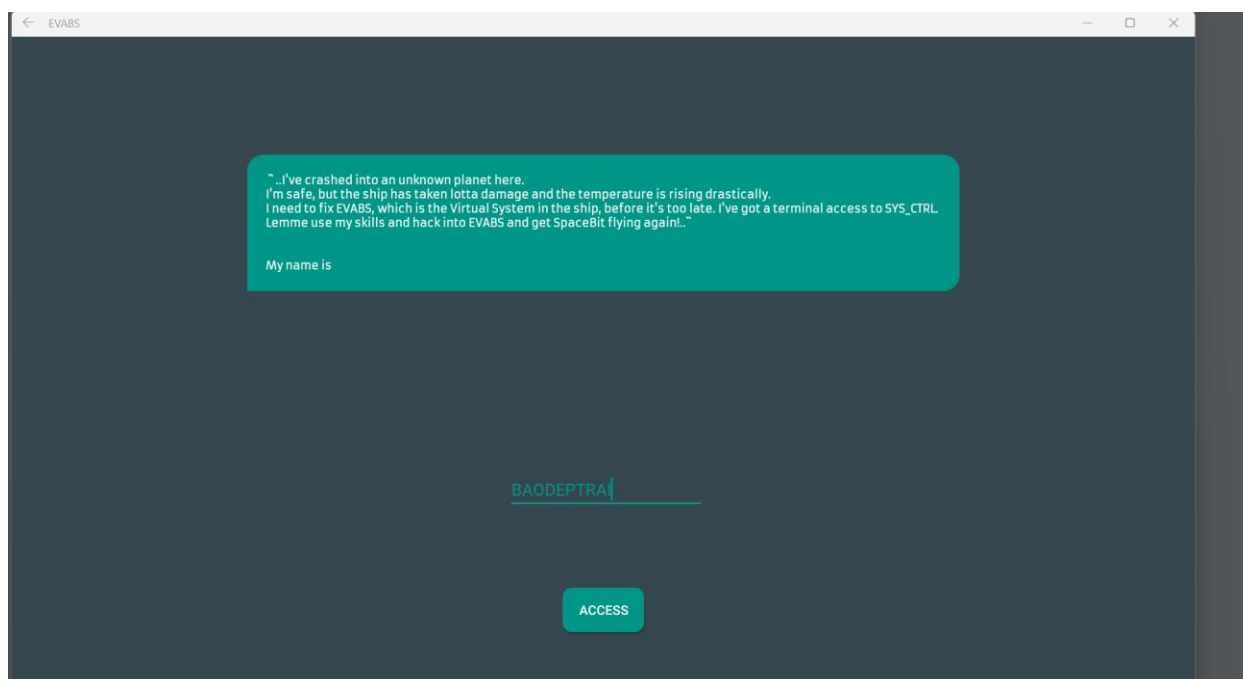
Kết quả



FLAG : picoCTF{not.particularly.silly}

EVABS

Vào ứng dụng



CHALLENGE1:

Pass có dạng : EVABS{

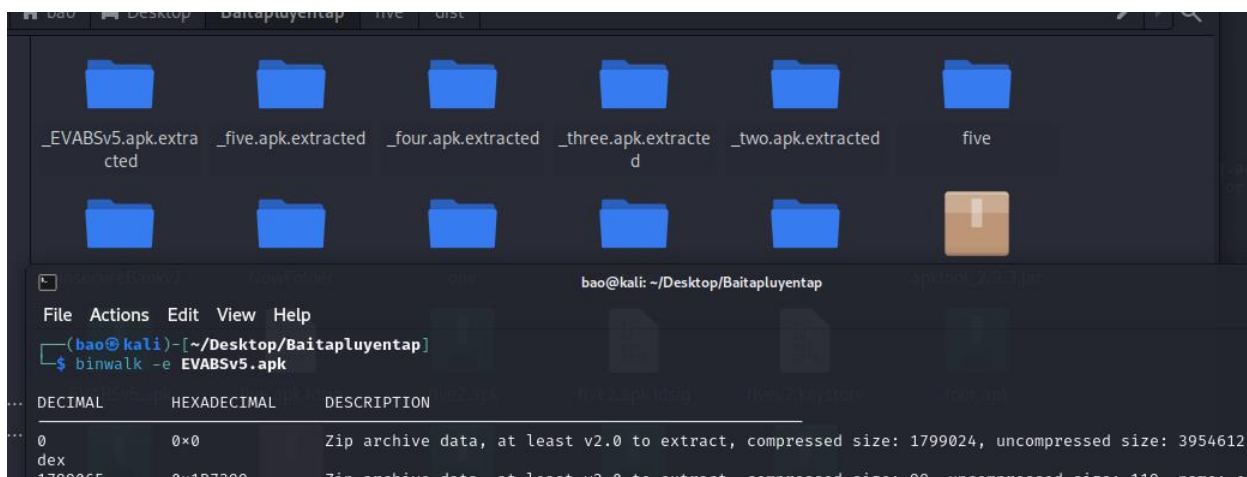
Nên ta sẽ tìm kiếm logcat với keyword như vậy

```
C:\Users\BaoBao>adb logcat | findstr "EVABS{"
05-08 19:43:27.382 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:43:39.472 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:43:56.321 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:35.761 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.083 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.221 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.365 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.511 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.647 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.801 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:36.969 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:39.555 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:39.698 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:40.627 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:40.797 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:42.779 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:42.940 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:45:43.093 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:48:33.455 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:48:33.607 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:48:52.950 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:52:57.097 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:52:57.259 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
05-08 19:52:57.395 10228 10228 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
```

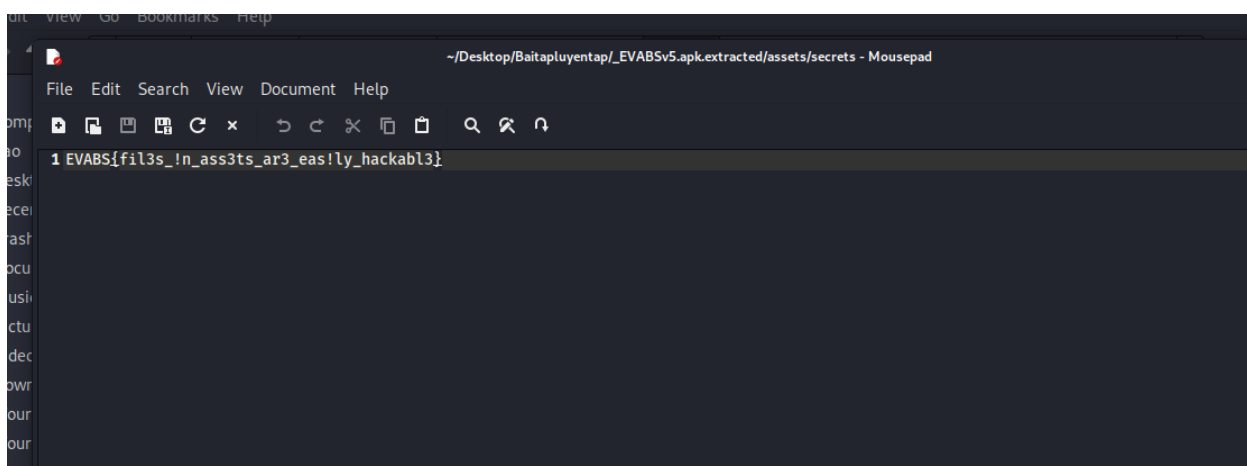
FLAG : EVABS{logging_info_never_safe}

CHALLENGE2:

Extract file EVABSV5.apk

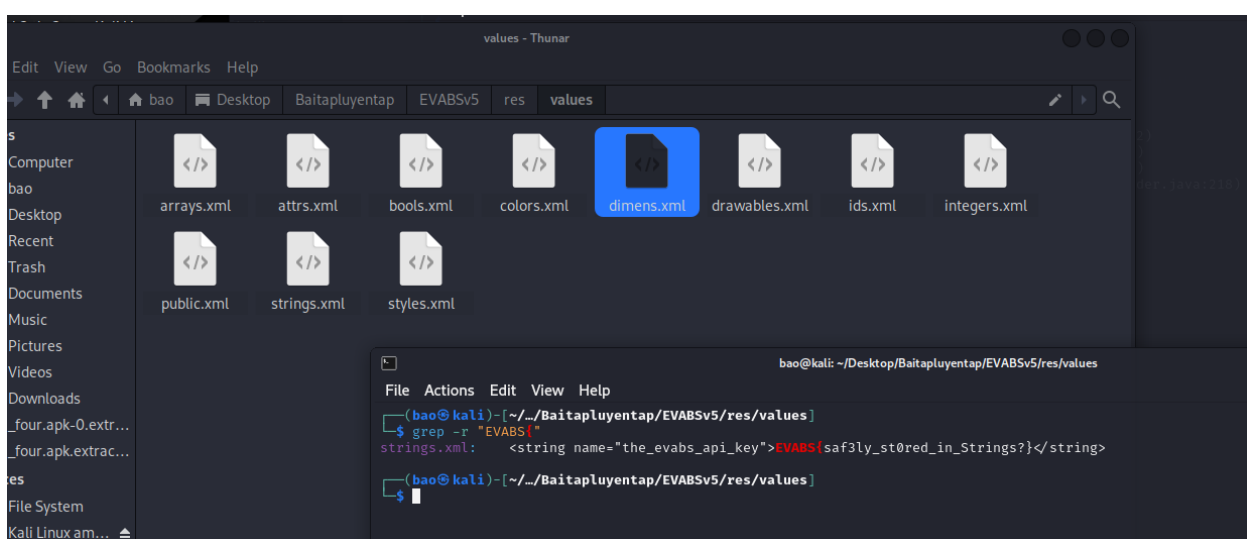


Kiểm tra folder asset



FLAG : EVABS{fil3s_!n_ass3ts_ar3_eas!ly_hackabl3}

CHALLENGE3:



FLAG : EVABS{saf3ly_st0red_in_Strings?}

CHALLENGE4:

```
(bao@kali)-[~/.../Baitapluientap/EVABSv5/res/values]
$ cd ..

(bao@kali)-[~/Desktop/Baitapluientap/EVABSv5/res]
$ grep -r "EVABS{"
raw/link.txt:EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}
layout/activity_flagcheck.xml:    <EditText android:textColor="@color/colorWh
```

FLAG : EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}

CHALLENGE5:

TA VÀO

/data/data/com.revo.evabs/shared, sau đó ta sử dụng lệnh grep -r "EVABS{" thì ta có được flag

NHƯNG KHÔNG TÌM THẤY GÌ CẢ!

```
vbox86p:/data/data/com.revo.evabs # ls
cache code_cache shared_prefs
vbox86p:/data/data/com.revo.evabs # grep -r "EVABS{"
1|vbox86p:/data/data/com.revo.evabs # |
```

CHALLENGE6: KHÔNG TÌM THẤY DATABASE

CHALLENGE7:

Đầu tiên thực hiện kiểm tra AndroidManifest.xml

Dòng này trong tệp Android Manifest (AndroidManifest.xml) định nghĩa một hoạt động (activity) trong ứng dụng Android của bạn. Cụ thể, dòng này định nghĩa một hoạt động có tên là ExportedActivity và thiết lập thuộc tính exported thành true. bạn.

Trong trường hợp này, ExportedActivity được đặt là exported="true", điều này có nghĩa là hoạt động này có thể được gọi từ các ứng dụng hoặc thành phần khác trên thiết bị.

```
android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/AppTheme">
<activity android:exported="true" android:name="com.revo.evabs.ExportedActivity"/>
<activity android:name="com.revo.evabs.Frida1"/>
<activity android:name="com.revo.evabs.FileRead"/>
<activity android:name="com.revo.evabs.DebugMe"/>
```

Ta sẽ thực hiện trigger để xem thông tin về các ExportedActivity với lệnh như console bên dưới, và chương trình trên điện thoại ta có được flag

KẾT QUẢ :

```
org.chromium.webview_shell
:/data/data # cd com.revo.evabs
:/data/data/com.revo.evabs # ls
cache code_cache shared_prefs
:/data/data/com.revo.evabs # cd shared_prefs
:/data/data/com.revo.evabs/shared_prefs # ls
PREFERENCE.xml
:/data/data/com.revo.evabs/shared_prefs #
PS D:\BaoMatweb\Lab4\Lab4\Baitapluientap> adb -s 192.168.56.102
vbox86p:/ # syu
/system/bin/sh: syu: inaccessible or not found
127|vbox86p:/ # su
:/ # cd /data/data/com.revo.evabs/shared_prefs
:/data/data/com.revo.evabs/shared_prefs # ls
PREFERENCE.xml
:/data/data/com.revo.evabs/shared_prefs #
PS D:\BaoMatweb\Lab4\Lab4\Baitapluientap> am start com.revo.eva
am : The term 'am' is not recognized as the name of a cmdlet, f
spelling of the name, or if a path was included, verify that th
At line:1 char:1
+ am start com.revo.evabs/com.revo.evabs.ExportedActivity
+ ~~~
+ CategoryInfo          : ObjectNotFound: (am:String) [], C
+ FullyQualifiedErrorId : CommandNotFoundException

PS D:\BaoMatweb\Lab4\Lab4\Baitapluientap> adb -s 192.168.56.102:5555 shell am start com.revo.evabs/com.revo.evabs.Export
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.revo.evabs/.ExportedAct
ivity }
PS D:\BaoMatweb\Lab4\Lab4\Baitapluientap>
```

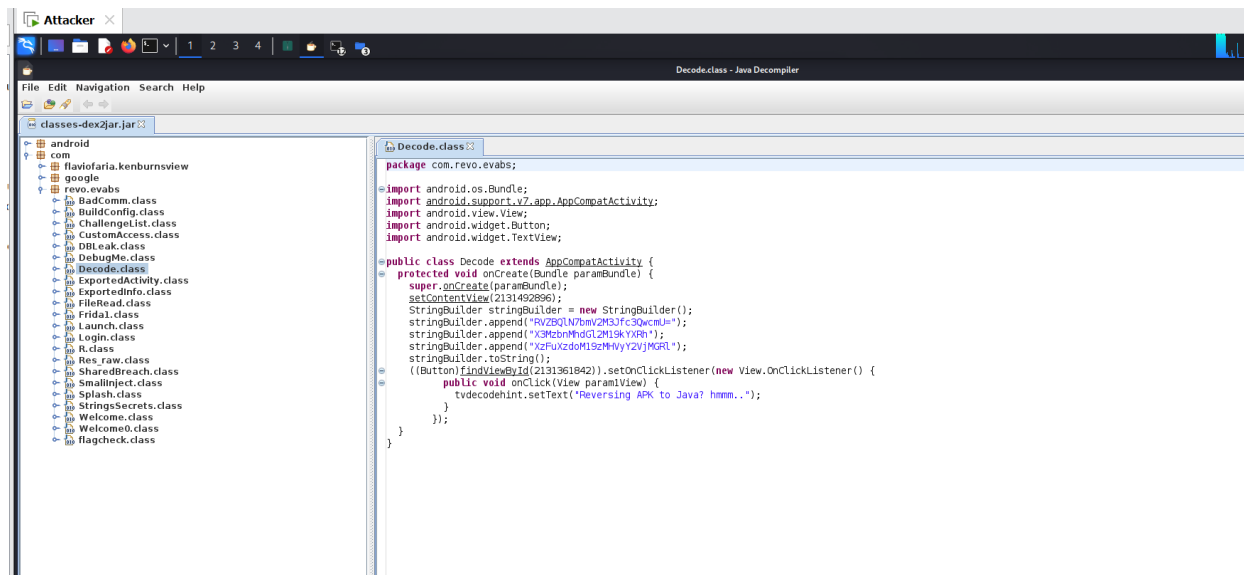
SYS_CTRL: ERROR: Secret service
breached. Data compromised:
EVABS{exp0rted_activities_ar3_harmful}r

Free for personal use

FLAG : EVABS{exp0rted_activities_ar3_harmful}

CHALLENGE8:

Sử dụng javagui để decompile file .apk và xem code DECODE



Kết quả sau khi decode

base64

RVZBQ1N7bmV2M3Jfc3QwcmVzX3MzbnMhdG12M19kYXRhXzFuXzdoM19zMHVyY2VjMGR1

Base64 Standard

Auto detection (works like a charm, however sometimes may fail for short strings)

Strict Decoding

No (ignore invalid characters and force decoding value as Base64).

Character Encoding

Auto detection (an experimental feature that may fail for "exotic" encodings)

Decode Base64

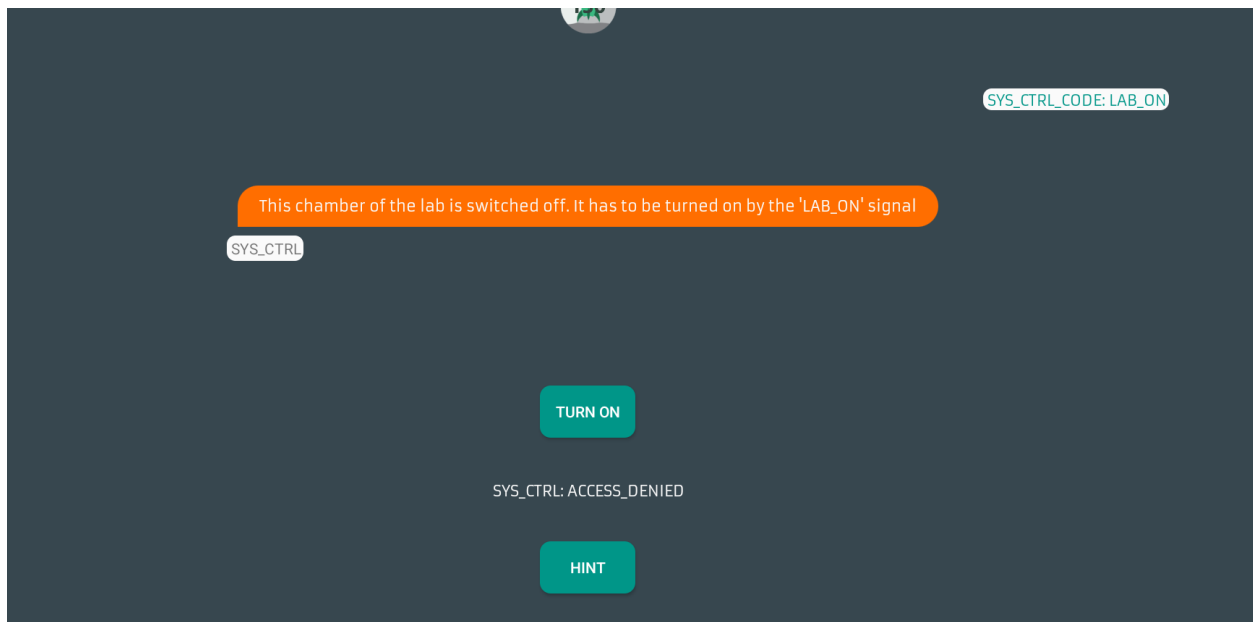
Text

EVABS{nev3r_st0res_s3ns!tiv3_data_1n_7h3_s0urcec0de

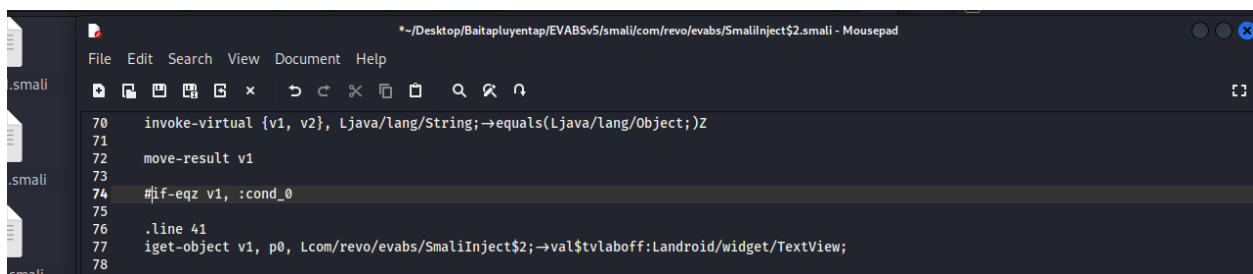

```
(bao@kali)-[~/Desktop/Baitapluientap]
$ java -jar apktool_2.9.3.jar b EVABSV5
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: EVABSV5/dist/EVABSV5.apk

(bao@kali)-[~/Desktop/Baitapluientap]
$
```

KẾT QUẢ:



ON nhưng không có gì :VVV



Quay lại file smali, ta thấy dòng này đang cản trở việc in flag nên mình sẽ # nó đi

LÝ DO :

```

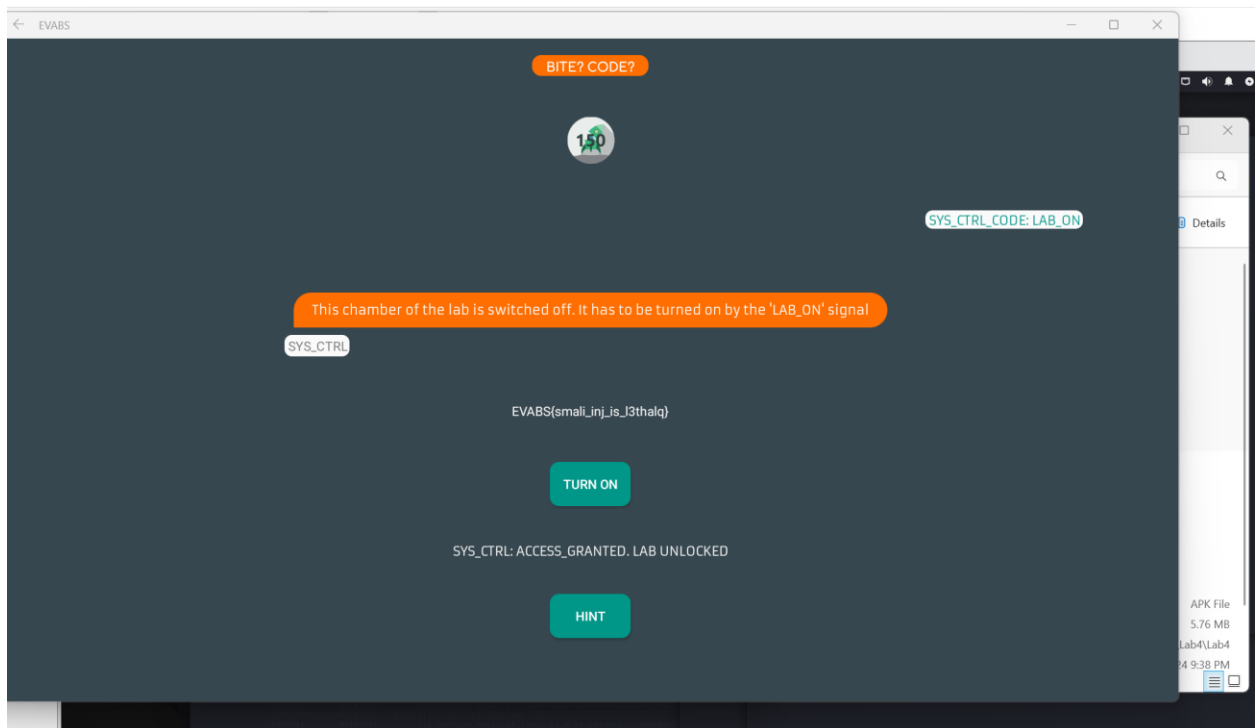
        tvsmalihint.setText("apktool? Editing smali? Repackaging?");
    }
});
button1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1View) {
        String str = SmaliInject.this.stringFromSmali();
        if (SmaliInject.this.SIGNAL.equals("LAB_ON")) {
            tvlaboff.setText("SYS_CTRL_CODE: LAB_ON");
            labstat.setText("SYS_CTRL: ACCESS_GRANTED. LAB UNLOCKED");
            TextView textView = tvflag;
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("EVABS{");
            stringBuilder.append(str);
            stringBuilder.append("}");
            textView.setText(stringBuilder.toString());
        } else {
            tvlaboff.setText("SYS_CTRL_CODE: LAB_OFF");
            labstat.setText("SYS_CTRL: ACCESS_DENIED");
        }
    }
});
}

public native String stringFromSmali();
}

```

Button này sẽ luôn trả về ACCESS_DENIED!

KẾT QUẢ :



FLAGS : EVABS{smali_inj_is_I3thalq}

CHALLENGE10: WEB KHÔNG CÒN TỒN TẠI NÊN BỎ

CHALLENGE11:

```

package com.revo.evabs;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class CustomAccess extends AppCompatActivity {
    public final String EVABS_SENSOR_KEY = "com.revo.evabs.action.SENSOR_KEY";

    static {
        System.loadLibrary("native-lib");
    }

    private void GetSensorKey() {
        String str = ((EditText)findViewById(2131361891)).getText().toString();
        if ((new String(new char[] {
            'c', 'u', 's', 't', '0', 'm', '_', 'p', '3', 'r',
            'm' })).equals(str)) {
            Toast.makeText((Context) this, "SYS_CTRL: CRED5 ACCEPTED. SENSOR_KEY SENT", 1).show();
            Intent intent = new Intent("com.revo.evabs.action.SENSOR_KEY");
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("EVABS(");
            stringBuilder.append(stringFromJNI());
            stringBuilder.append(")");
            intent.putExtra("android.intent.extra.TEXT", stringBuilder.toString());
            intent.setType("text/plain");
            startActivity(intent);
        } else {
            Toast.makeText((Context) this, "SYS_CTRL: WRONG_CRED5. SENSOR_KEY LOOKED", 1).show();
        }
    }

    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2131492893);
        ((Button)findViewById(2131361835)).setOnClickListener(new View.OnClickListener() {
            public void onClick(View param1View) {
                CustomAccess.this.GetSensorKey();
            }
        });
        final TextView tvhintcust = (TextView)findViewById(2131362091);
        ((Button)findViewById(2131361841)).setOnClickListener(new View.OnClickListener() {
            public void onClick(View param1View) {
                tvhintcust.setText("Can you trick a custom action?");
            }
        });
    }

    public native String stringFromJNI();
}

```

```

private void GetSensorKey() {
    String str = ((EditText)findViewById(2131361891)).getText().toString();
    if ((new String(new char[] {
        'c', 'u', 's', 't', '0', 'm', '_', 'p', '3', 'r',
        'm' })).equals(str)) {
        Toast.makeText((Context) this, "SYS_CTRL: CRED5 ACCEPTED. SENSOR_KEY SENT", 1).show()
    }
}

```

cust0m_p3rm ta sẽ tạm lưu lại thông tin này, vì nó giống password

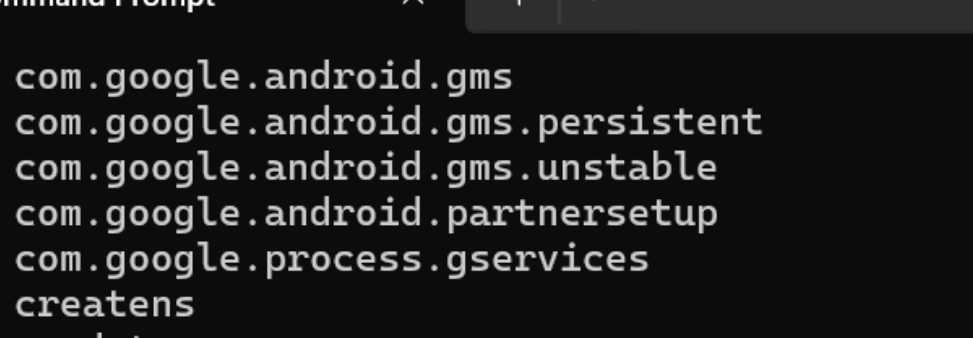
Chạy Frida_server phía máy android

```

PS D:\BaoMatweb\Lab4\Lab4\Baitapluuyentap> adb -s 192.168.56.102:5555 shell
vbox86p:/ # su
:/ # cd /data/data/tmp/
sh: cd: /data/data/tmp: No such file or directory
2|:/ # /data/local/tmp
sh: /data/local/tmp: can't execute: Is a directory
126|:/ # cd /data/local/tmp
:/data/local/tmp # ls
frida-server                                frida-server-16.2.1-linux-x86_64
frida-server-16.2.1-android-x86_64
:/data/local/tmp # cd frida-server-16.2.1-android-x86_64
:/data/local/tmp/frida-server-16.2.1-android-x86_64 # ls
frida-server-16.2.1-android-x86_64
v frida-server-16.2.1-android-x86_64 frida_server <
:/data/local/tmp/frida-server-16.2.1-android-x86_64 # ls
frida_server
:/data/local/tmp/frida-server-16.2.1-android-x86_64 # chmod 755 frida_server
:/data/local/tmp/frida-server-16.2.1-android-x86_64 # ls
frida_server
:/data/local/tmp/frida-server-16.2.1-android-x86_64 # ./frida_server

```

Phía windows



The screenshot shows a Windows Command Prompt window titled "Command Prompt". It contains a list of processes with their PIDs and names. The process "frida_server" with PID 7715 is highlighted. The list of processes is as follows:

PID	Process Name
5722	com.google.android.gms
5730	com.google.android.gms.persistent
6024	com.google.android.gms.unstable
7075	com.google.android.partnersetup
1729	com.google.process.gservices
191	createns
395	credstore
398	diskiod
7715	frida_server
483	gatekeeperd
405	genybaseband
400	gpuservice
533	hostapd_noaidl
98	hwservicemanager
459	incidentd
1	init

```
> BaoMatweb > Lab4 > bai11.py > ...
9 intent.putExtra.overload("java.lang.String", "java.lang.String").implementation = function(var_1, var_2) {
10     send("[+] Flag: " + var_2);
11 };
12 """
13
14 # print the message
15 def printGetInfor(message, data):
16     print(message)
17
18 # attach the process with the specific PID (change the PID accordingly)
19 process = frida.get_usb_device().attach(111)
20 |
21 # create the script with Java code
22 script = process.create_script(javaCode)
23
24 # add the message handler to print the message
25 script.on("message", printGetInfor)
26
27 # start hooking
28 print("[*] Hooking")
29
30 # load the script
31 script.load()
32
33 # keep the script running to read the result
34 sys.stdin.read()
```

PROBLEMS 62 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
return f(*args, **kwargs)
~~~~~
File "C:\Users\BaoBao\AppData\Local\Programs\Python\Python312\Lib\site-packages\frida\core.py", line 1010, in attach
    return Session(self._impl.attach(self_pid_of(target), **kwargs)) # type: ignore
           ~~~~~
frida.ProcessNotFoundError: unable to find process with pid 111111
$ C:\Users\BaoBao> & c:/Users/BaoBao/AppData/Local/Programs/Python/Python312/python.exe d:/BaoMatweb/Lab4/bai11.py
[*] Hooking
type: 'send', 'payload': '[*] Starting hooks android.content.Intent.putExtra'
type: 'error', 'description': "TypeError: cannot read property 'use' of null", 'stack': "TypeError: cannot read property 'use' of null\n    at use (frida/node_modules/frida-java-bridge/index.js:258)\n    at <eval> (</script1.js:3)", 'fileName': 'frida/node_modules/frida-java-bridge/index.js', 'lineNumber': 258, 'columnNumber': 1}
```