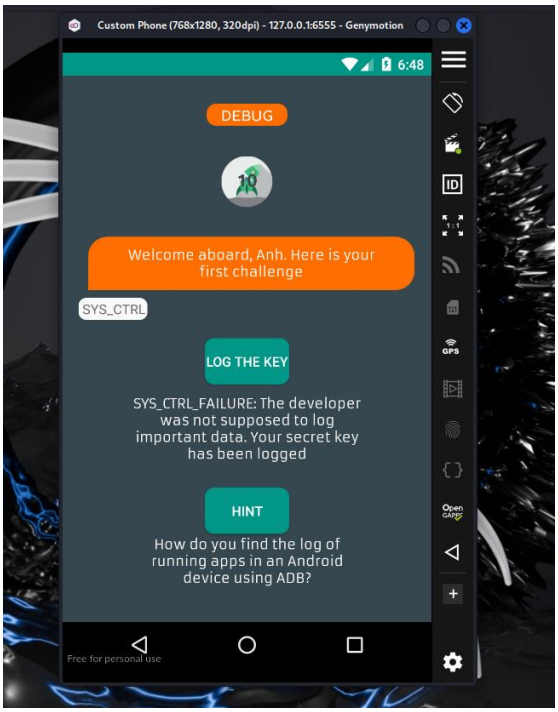
BÁO CÁO CHI TIẾT

Challenges 1 Hoàn thành 12 levels.

Level 1:



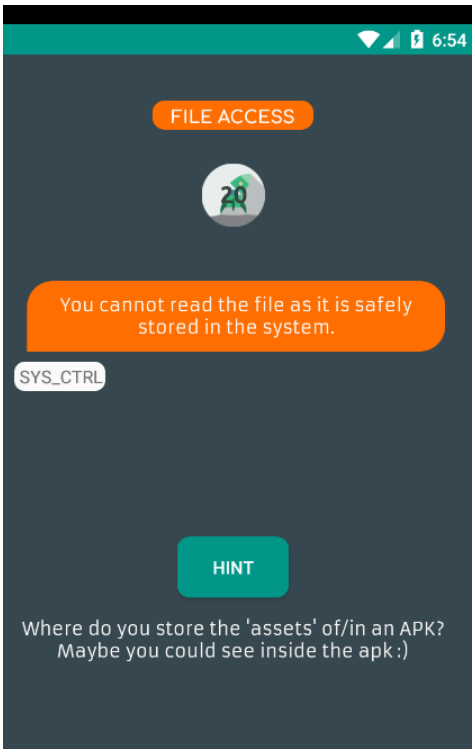
Với hint như trên, ta dùng *adb logcat* và *grep* EVABS

```
05-22 18:47:18.686  6532  7587 W android.os.Debug: failed to get memory consumpt
ion info: -1
05-22 18:47:19.483   482   482 W batteryd: type=1400 audit(0.0:35318): avc: gran
ted { read } for path="/dev/fuse" dev="tmpfs" ino=2231 scontext=u:r:init:s0 tcon
text=u:object_r:fuse_device:s0 tclass=chr_file
^C

┌──(kali㉿kali)-[~/NT213/Lab4/Lab4]
└─$ adb logcat | grep EVABS
05-22 18:46:53.221  7490  7490 D ** SYS_CTRL **: : EVABS{logging_info_never_safel}
```

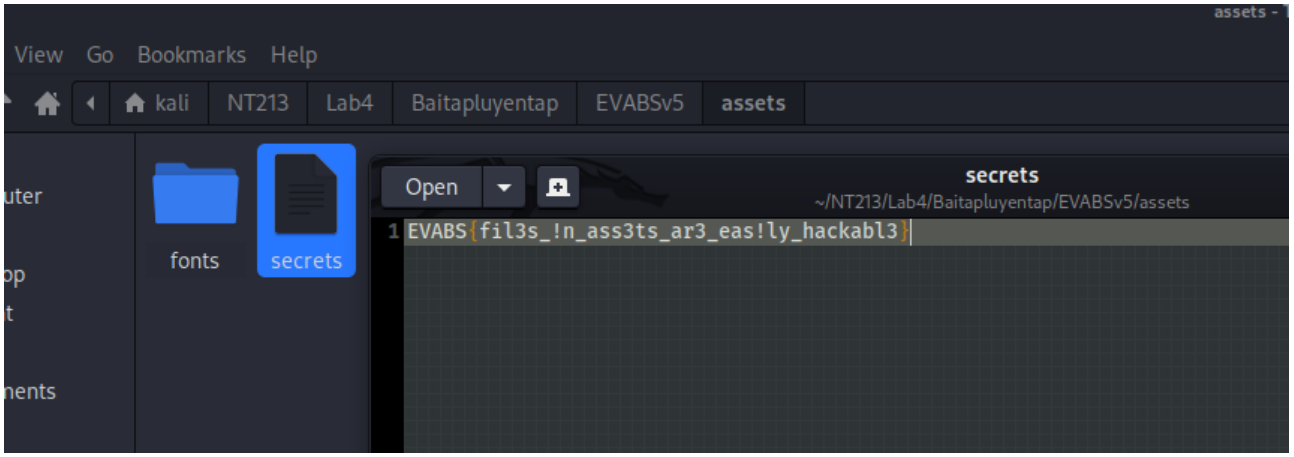
⇨ Ta có được flag: **EVABS{logging_info_never_safel}**

Level 2:

Với hint này



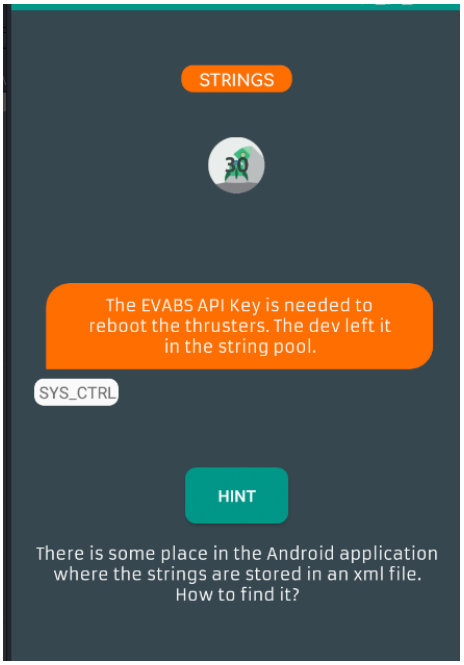
Ta giải nén file apk và vào folder assets, tại đây có luôn flag



⇨ **EVABS{fil3s_!n_ass3ts_ar3_eas!ly_hackabl3}**

Level 3:

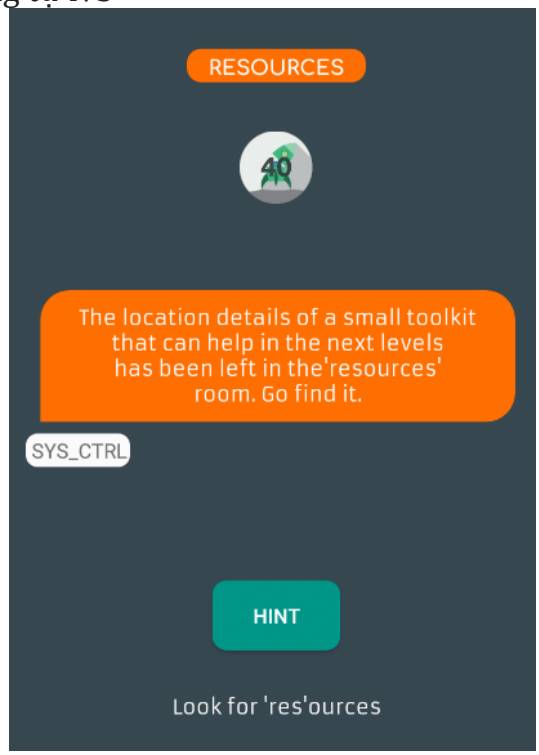
Với hint rằng strings được lưu trong xml file



Ta decompile file apk, cat và grep flag

⇨ Ta được flag: **EVABS{saf3ly_st0red_in_Strings?}**

Level 4:
Với hint này thì làm tương tự lv3

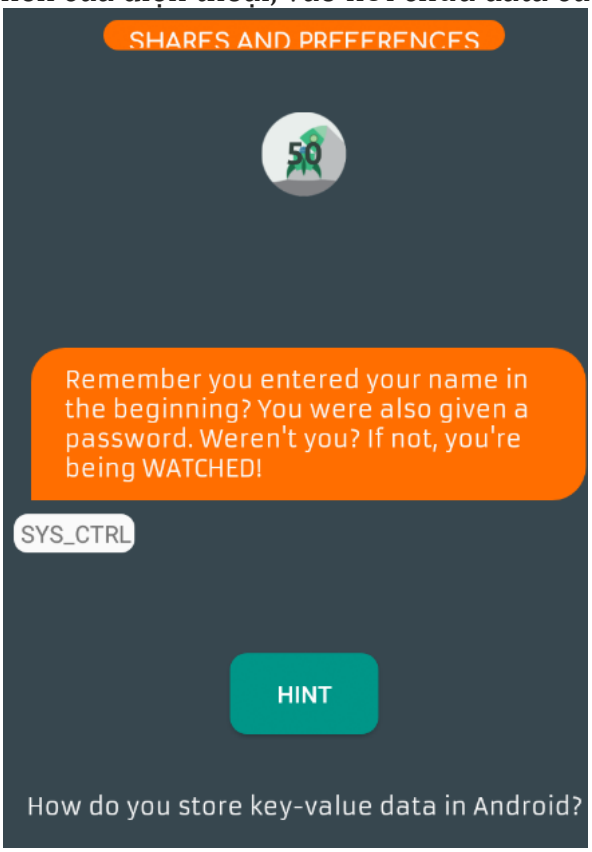


Vào folder res và thực hiện grep

```
┌──(kali㉿kali)-[~/…/Lab4/Baitapluyentap/EVABSv5/res]
└─$ grep "EVABS{" */*
layout/activity_flagcheck.xml:    <EditText android:textColor="@color/colorWhite" android:id="@id/e
marginTop="8.0dip" android:layout_marginRight="8.0dip" android:layout_marginBottom="8.0dip" android
tEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.503" app:layout_constraintStart_toSta
layout/activity_flagcheck.xml:    <TextView android:textSize="10.0dip" android:textColor="@color/co
roid:layout_height="24.0dip" android:layout_marginLeft="15.0dip" android:layout_marginTop="8.0dip"
xt_here}." app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent
layout-v17/activity_flagcheck.xml:    <EditText android:textColor="@color/colorWhite" android:id="@
out_marginTop="8.0dip" android:layout_marginRight="8.0dip" android:layout_marginBottom="8.0dip" and
p" android:layout_marginEnd="8.0dip" app:layout_constraintBottom_toBottomOf="parent" app:layout_con
aintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.401" />
layout-v17/activity_flagcheck.xml:    <TextView android:textSize="10.0dip" android:textColor="@colo
 android:layout_height="24.0dip" android:layout_marginLeft="15.0dip" android:layout_marginTop="8.0d
e_t3xt_here}." android:textAlignment="center" android:layout_marginStart="8.0dip" android:layout_ma
StartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.96
raw/link.txt:EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}
values/strings.xml:    <string name="the_evabs_api_key">EVABS{saf3ly_st0red_in_Strings?}</string>
```

⇨ Ta được flag: **EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}**

Level 5:
Với hint này thì ta vào shell của điện thoại, vào nơi chứa data của ứng dụng

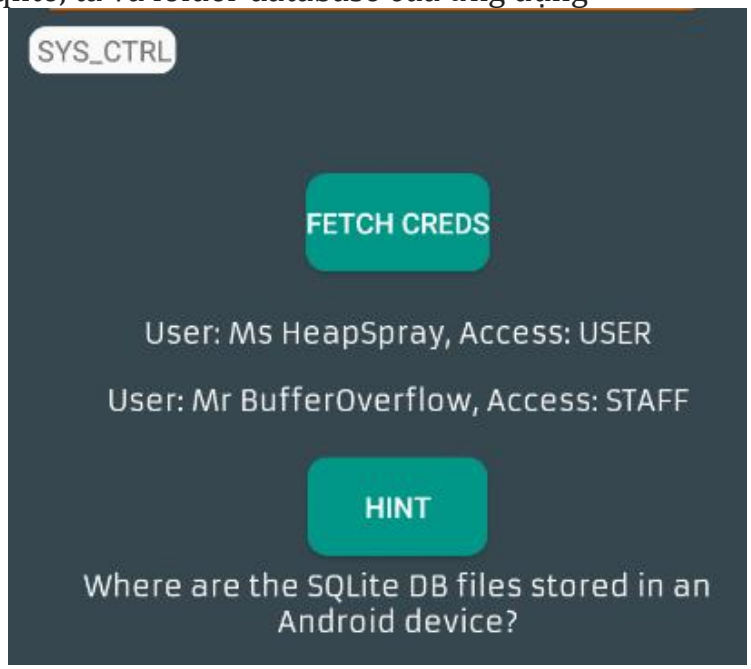


Dùng lệnh grep, ta được flag

```
┌──(kali㉿kali)-[~/…/Lab4/Baitapluyentap/EVABSv5/res]
└─$ adb shell
genymotion:/ # cd data/data/com.revo.evabs/
genymotion:/data/data/com.revo.evabs # ls
cache  code_cache  lib  shared_prefs
genymotion:/data/data/com.revo.evabs # grep "EVABS{" */*
shared_prefs/DETAILS.xml:    <string name="password">EVABS{shar3d_pr3fs_c0uld_be_c0mpromiz3ds}</string>
genymotion:/data/data/com.revo.evabs #
```

⇨ **EVABS{shar3d_pr3fs_c0uld_be_c0mpromiz3ds}**

Level 6:
Với hint là dùng sqlite, ta và folder database của ứng dụng



Dùng sqlite3 xem file db và có được flag

```
┌──(kali㉿kali)-[~/…/Lab4/Baitapluyentap/EVABSv5/res]
└─$ adb shell
genymotion:/ # cd /data/data/com.revo.evabs/
genymotion:/data/data/com.revo.evabs # ls
cache  code_cache  databases  lib  shared_prefs
genymotion:/data/data/com.revo.evabs # cd databases/
genymotion:/data/data/com.revo.evabs/databases # ls
MAINFRAME_ACCESS  MAINFRAME_ACCESS-journal
genymotion:/data/data/com.revo.evabs/databases # sqlite3 M
MAINFRAME_ACCESS          MAINFRAME_ACCESS-journal
genymotion:/data/data/com.revo.evabs/databases # sqlite3 MAINFRAME_ACCESS
SQLite version 3.18.2 2017-07-21 07:56:09
Enter ".help" for usage hints.
sqlite> .tables
CREDS             android_metadata
sqlite> select * from CREDS;
Dr.l33t|EVABS{sqlite_is_not_safe}E|ADMIN
Mr BufferOverflow|0xNotSecureSQLite_|STAFF
Ms HeapSpray|SQLite_expl0it|USER
sqlite>
```

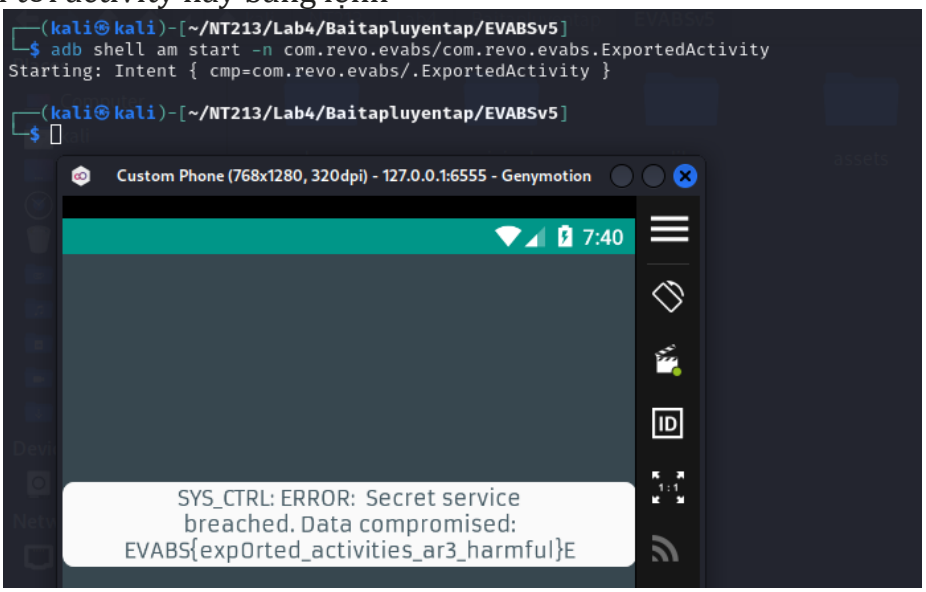
⇨ **EVABS{sqlite_is_not_safe}**

Level 7:

Exported activities là những hành động trong ứng dụng Android có thể được truy cập và sử dụng bởi các ứng dụng khác. Các hành động này được khai báo trong tệp AndroidManifest.xml. Kiểm tra file ta thấy exported = “true”

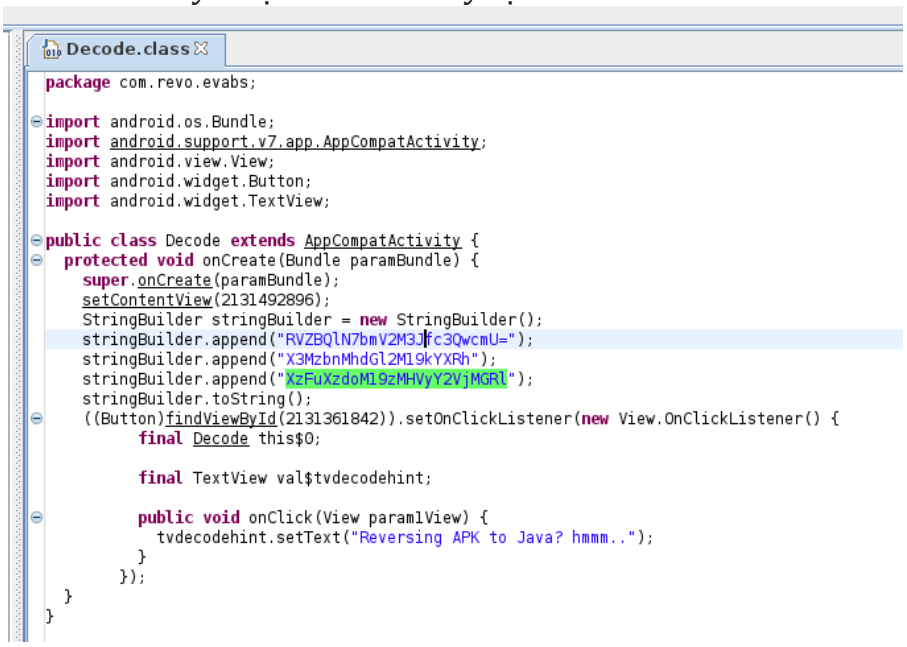
```
  AppTheme">
4         <activity android:exported="true" android:name="com.revo.evabs.ExportedActivity" />
5         <activity android:name="com.revo.evabs.Frida1" />
6         <activity android:name="com.revo.evabs.FileRead" />
7         <activity android:name="com.revo.evabs.DebugMe" />
```

Ta có thể gọi tới activity này bằng lệnh



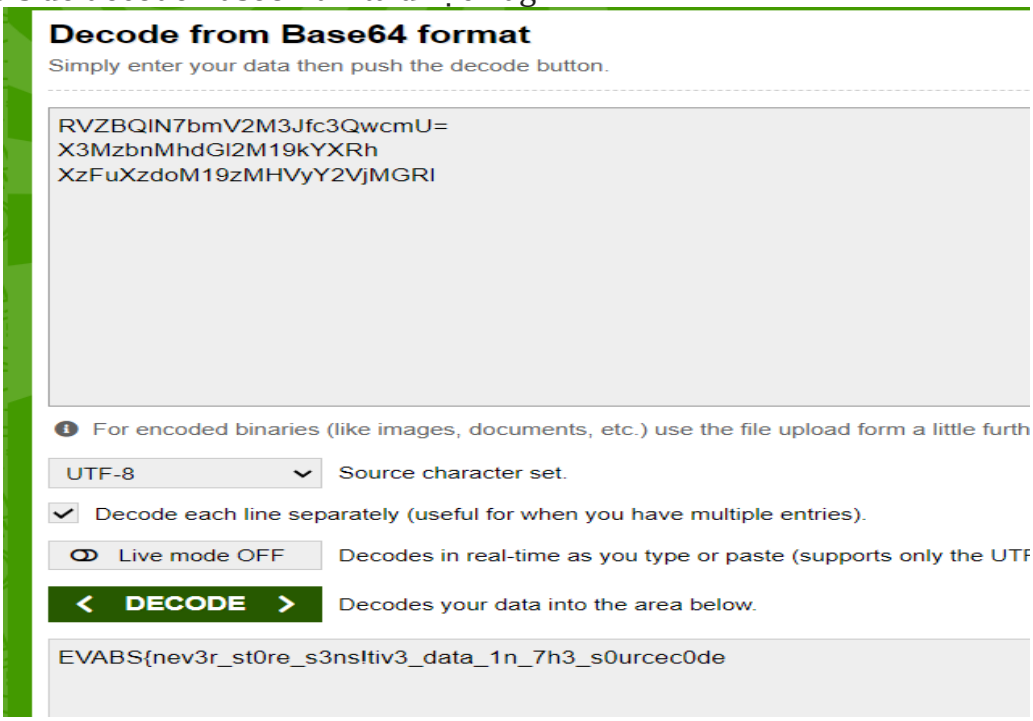
Lúc này ứng dụng hiện ra flag: EVABS{exp0rted_activities_ar3_harmful}

Level 8:
Unzip file apk, dùng tools d2j để chuyển file classes.dex thành file jar và dùng jd-gui để đọc. Tại lớp Decode ta thấy được chuỗi các ký tự



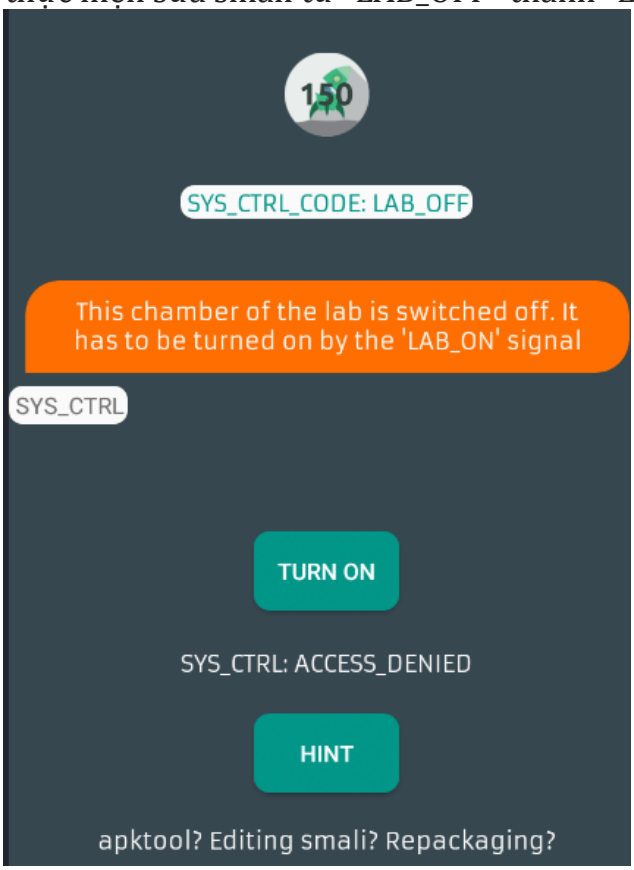
```
Decode.class
package com.revo.evabs;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class Decode extends AppCompatActivity {
  protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131492896);
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("RVZBQlN7bmV2M3Jfc3QwcmU=");
    stringBuilder.append("X3MzbnMhdGl2M19kYXRh");
    stringBuilder.append("XzFuXzdoM19zMHVyY2VjMGRl");
    stringBuilder.toString();
    ((Button)findViewById(2131361842)).setOnClickListener(new View.OnClickListener() {
          final Decode this$0;

          final TextView val$tvdecodehint;

          public void onClick(View param1View) {
            tvdecodehint.setText("Reversing APK to Java? hmmm..");
          }
        });
  }
}
```

Dùng tools để decode Base64 thì ta được flag

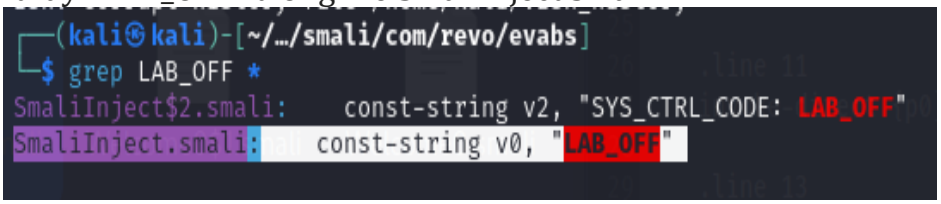


=> EVABS{nev3r_st0re_s3ns!tiv3_data_1n_7h3_s0urcec0de}

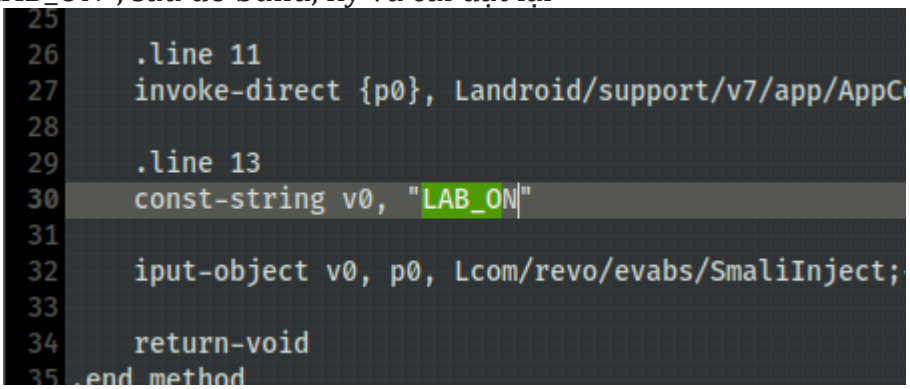
Level 9:
Yêu cầu của bài này là thực hiện sửa smali từ “LAB_OFF” thành “LAB_ON”

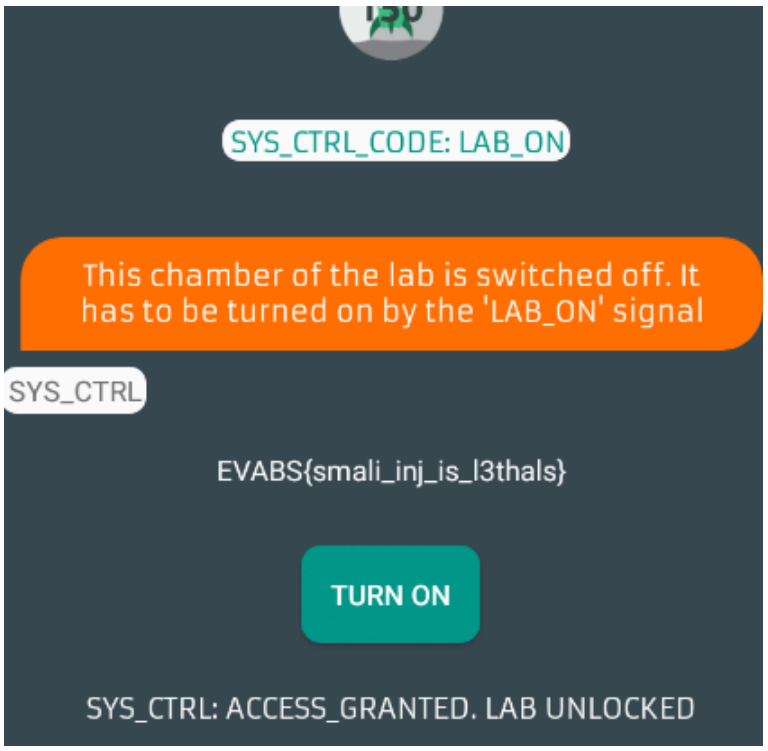


Tìm kiếm và thấy “LAB_OFF” trong file SmaliInject.smali



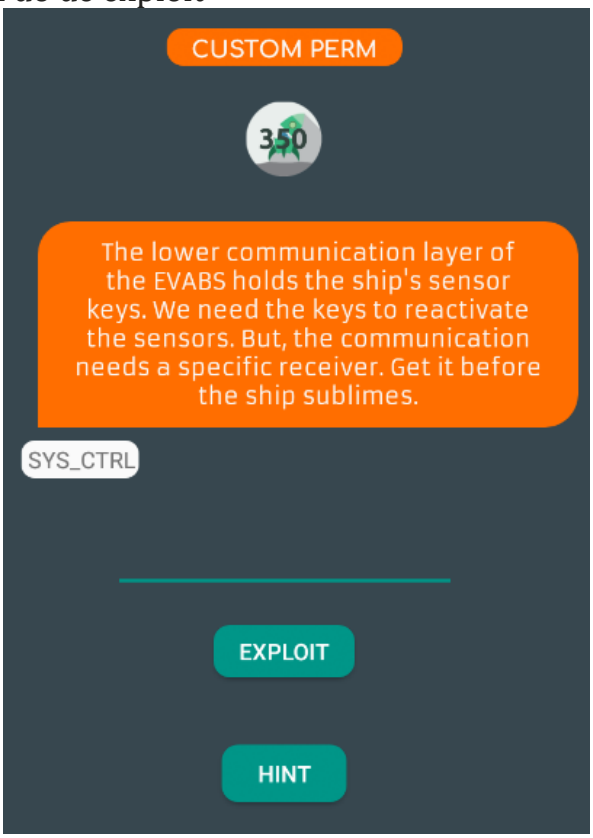
Sửa thành “LAB_ON”, sau đó build, ký và cài đặt lại





Ta có được flag: **EVABS{smali_inj_is_l3thals}**

Level 11:
Yêu cầu này cần nhập gì đó để exploit



Vào đọc code thì thấy cần nhập "cust0m_p3rm". Nhưng khi nhập xong vẫn chưa có flag

```
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: CREDS 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_CREDS. SENSOR_KEY LOCKED", 1).show();
  }
}
```

Tuy nhiên ta thấy được flag sẽ được stringBuilder nối thành chuỗi
Đọc code smali ta thấy ký tự cuối được nối stringBuilder nối vào v5. Ta thực hiện ghi log lại bằng code "invoke-static {v5}, Ljava/lang/System; ->loadLibrary(Ljava/lang/String;)V"

```
114
115     const-string v6, "}"
116
117
118     invoke-virtual {v5, v6}, Ljava/lang/StringBuilder;→append(Ljava/lang/String;)Ljava/lang/StringBuilder;
119
120     invoke-virtual {v5}, Ljava/lang/StringBuilder;→toString()Ljava/lang/String;
121
122     move-result-object v5
123     invoke-static {v5}, Ljava/lang/System;→loadLibrary(Ljava/lang/String;)V
124     const-string v6, "android.intent.extra.TEXT"
```

Sau đó đọc logcat và thấy được flag

```
┌──(kali㉿kali)-[~/NT213/Lab4/Baitapluyentap]
└─$ adb logcat | grep EVABS
05-22 20:41:54.099  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 20:42:00.511  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 20:42:34.131  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 20:44:26.854  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 20:44:33.774  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 20:47:35.435  9757  9757 D Here's The soooper flaggie: EVABS{You've been tricked}
05-22 21:14:56.271 10925 10925 E AndroidRuntime: java.lang.UnsatisfiedLinkError: dalvik.system.PathClassLoader
[DexPathList[[zip file "/data/app/com.revo.evabs-mNM4M1dsfLeZDj6zy6SGDg==/base.apk"],nativeLibraryDirectories=
[/data/app/com.revo.evabs-mNM4M1dsfLeZDj6zy6SGDg==/lib/x86, /data/app/com.revo.evabs-mNM4M1dsfLeZDj6zy6SGDg==/
base.apk!/lib/x86, /system/lib, /system/vendor/lib]]] couldn't find "libEVABS{always_ver1fy_packag3sa}.so"
```

⇨ **EVABS{always_ver1fy_packag3sa}**

Level 12:

Từ code ta thấy flag sẽ được ghi log nếu x = a * b = 25 * 2 > random(70) + 150

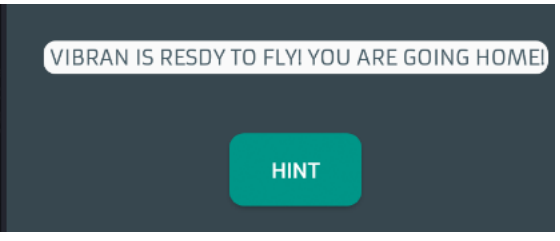
```
public class Fridal extends AppCompatActivity implements View.OnClickListener {
  int a = 25;

  int b = 2;

  int x;

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

  public void onClick(View paramView) {
    TextView textView2 = (TextView)findViewById(2131361996);
    TextView textView1 = (TextView)findViewById(2131362132);
    TextView textView4 = (TextView)findViewById(2131362134);
    TextView textView3 = (TextView)findViewById(2131362142);
    textView1.setText(String.valueOf(this.a));
    textView4.setText(String.valueOf(this.b));
    this.x = this.a * this.b;
    int i = (new Random()).nextInt(70);
    textView3.setText(String.valueOf(this.x));
    if (this.x > i + 150) {
      textView2.setText("VIBRAN IS RESDY TO FLY! YOU ARE GOING HOME!");
      Log.d("CONGRATZ!", stringFromJNI());
    } else {
      textView2.setText("Co-ordinates Not Found!");
    }
  }
```

Vào code smali sửa giá trị random(70) + 150 sao cho < 50

```
126     .line 46
127     .local v4, "r":Ljava/util/Random;
128     const/16 v5, 0x5
129
130     invoke-virtual {v4, v5}, Ljava/util/Random;->nextInt(I)I
131
132     move-result v5
133
134     add-int/lit16 v5, v5, 0x5
135
```

Build lại và có flag

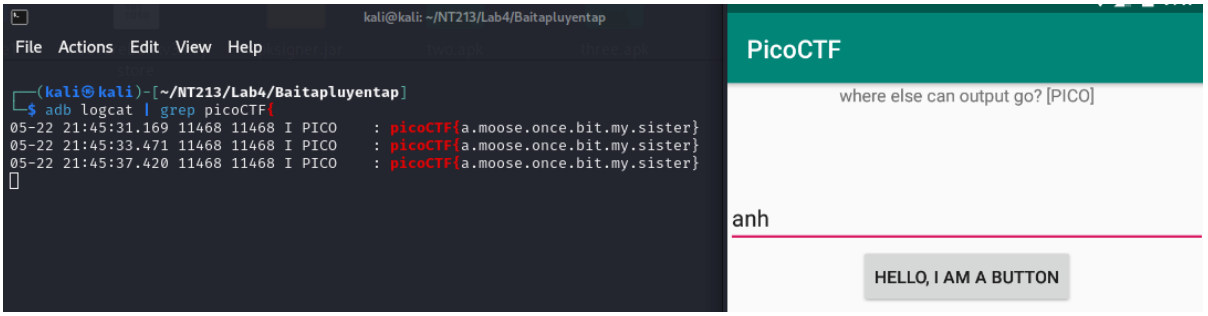
```
┌──(kali㉿kali)-[~/NT213/Lab4/Baitapluyentap]
└─$ adb logcat | grep EVABS
05-22 21:14:56.271 10925 10925 E AndroidRuntime: java.lang.Unsatisfi
[DexPathList[[zip file "/data/app/com.revo.evabs-mNM4M1dsfLeZDj6zy6S
[/data/app/com.revo.evabs-mNM4M1dsfLeZDj6zy6SGDg==/lib/x86, /data/ap
base.apk!/lib/x86, /system/lib, /system/vendor/lib]]] couldn't find
05-22 21:39:10.976 11317 11317 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
```



⇨ **EVABS{a_dynam1c_h00k}**

Challenges 2 Hoàn thành 5 challenges

One.apk

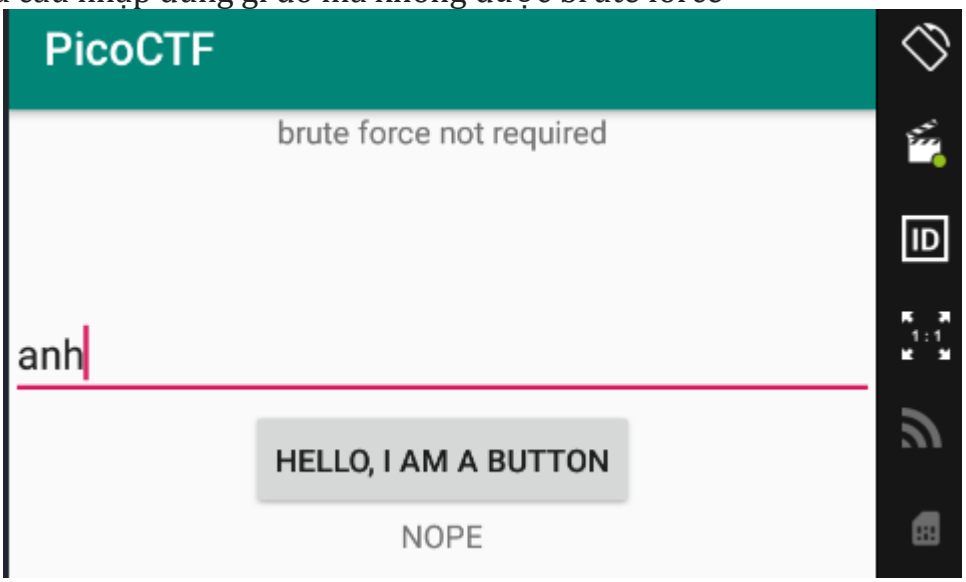


Bấm nút và vào logcat xem thì thấy flag:

⇨ **picoCTF{a.moose.once.bit.my.sister}**

Two.apk

Bài này yêu cầu nhập đúng gì đó mà không được brute force



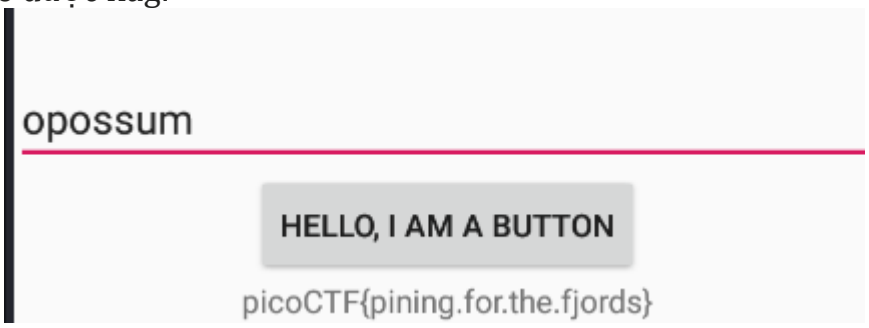
Vào đọc code thì tại code smali thấy được password

```
26
27    invoke-virtual {p1, v0}, Landroid/content/Context;→getString(I)Ljava/lang/String;
28
29    move-result-object v0
30
31    .line 12
32    .local v0, "password":Ljava/lang/String;
33    invoke-virtual {p0, v0}, Ljava/lang/String;→equals(Ljava/lang/Object;)Z
34
35    move-result v1
36
37    if-eqz v1, :cond_0
```

Thử grep password thì thấy

```
┌──(kali㉿kali)-[~/NT213/Lab4/Baitapluyentap (2)/two]
└─$ grep -r "password"
smali/com/hellocmu/picoctf/FlagstaffHill.smali:    .local v0, "password":Ljava/lang/String;
smali/com/hellocmu/picoctf/R$string.smali:.field public static final password:I = 0x7f0b002f
smali/androidx/core/view/accessibility/AccessibilityNodeInfoCompat.smali:    .param p1, "password"    # Z
smali/androidx/core/view/accessibility/AccessibilityNodeInfoCompat.smali:    const-string v2, "; password: "
res/values/strings.xml:    <string name="password">opossum</string>
res/values/public.xml:    <public type="string" name="password" id="0x7f0b002f" />

┌──(kali㉿kali)-[~/NT213/Lab4/Baitapluyentap (2)/two]
└─$ sS
```

Nhập vào và có được flag:



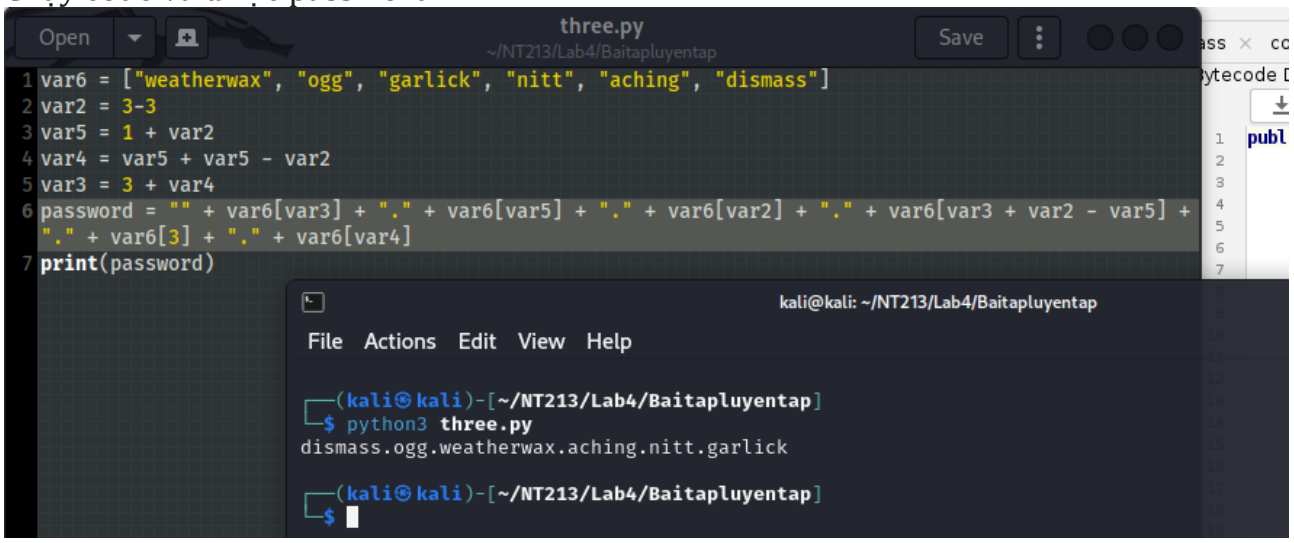
⇨ **picoCTF{pining.for.the.fjords}**

Three.apk

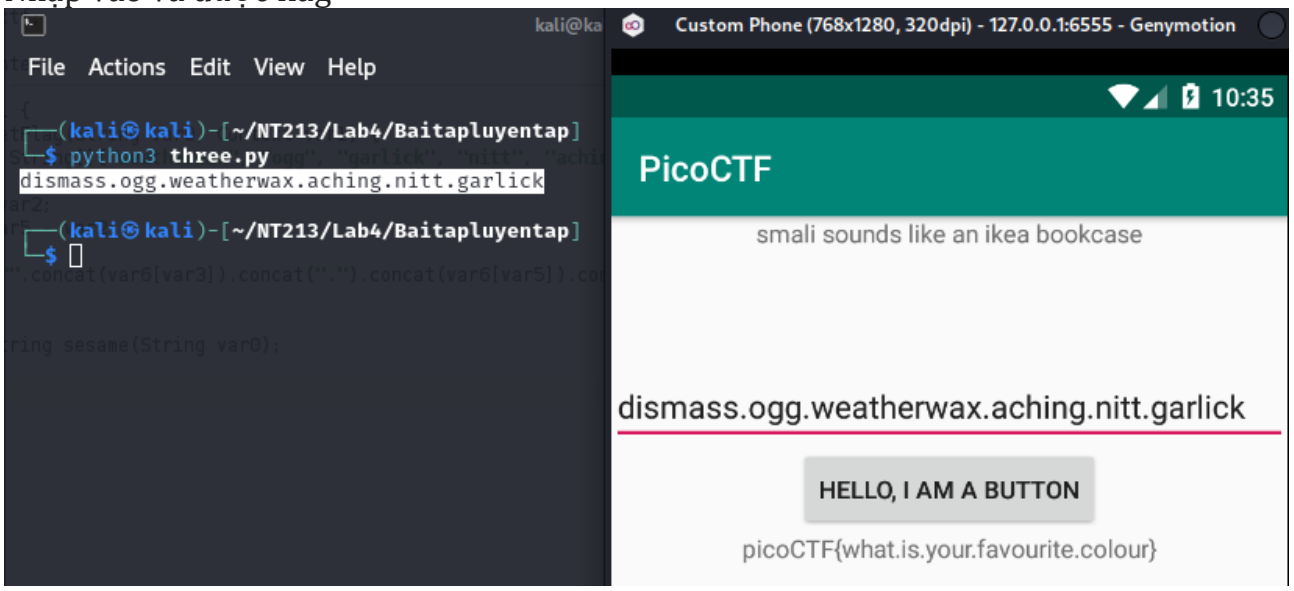
Từ code ta thấy cần nhập password đúng theo thứ tự để được flag

```
package com.hellocmu.picoctf;

import android.content.Context;

public class FlagstaffHill {
   public static String getFlag(String var0, Context var1) {
      String[] var6 = new String[]{"weatherwax", "ogg", "garlick", "nitt", "aching", "dismass"};
      int var2 = 3 - 3;
      int var5 = 3 / 3 + var2;
      int var4 = var5 + var5 - var2;
      int var3 = 3 + var4;
      return var0.equals("".concat(var6[var3]).concat(".").concat(var6[var5]).concat(".").concat(var6[var2]).c
   }

   public static native String sesame(String var0);
}
```

Chạy code và được password

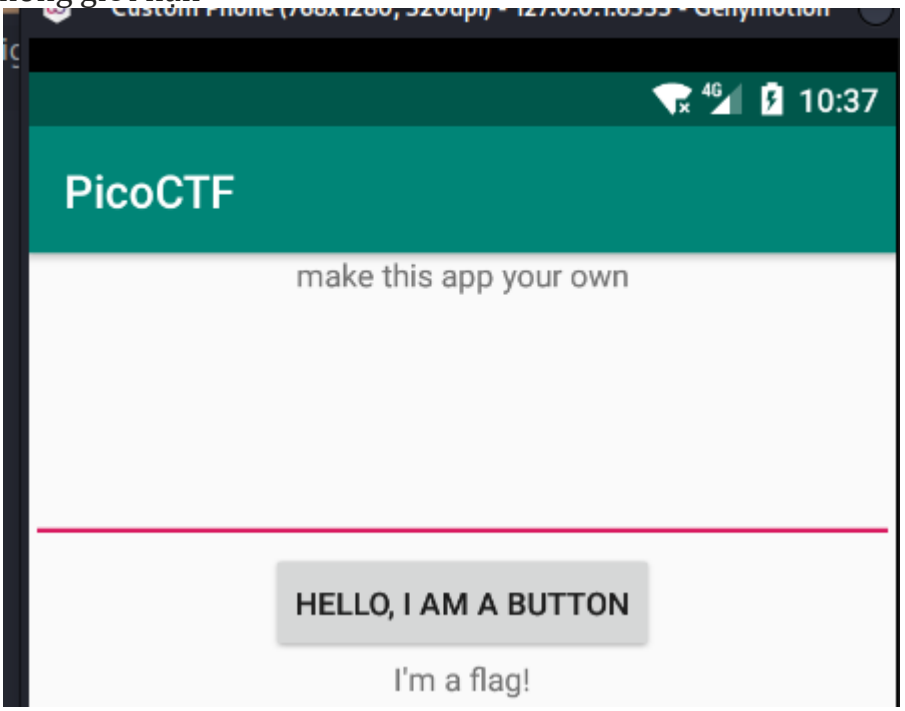


Nhập vào và được flag



⇨ **picoCTF{what.is.your.favourite.colour}**

Four.apk
Lab này thì không giới hạn



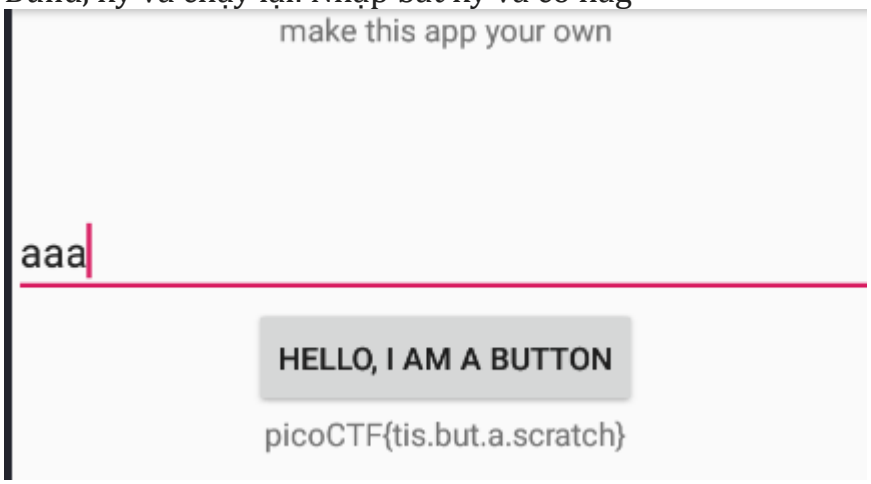
Từ code có thể flag sẽ xuất hiện khi gọi hàm yep

```
1  package com.hellocmu.picoctf;
2
3  import android.content.Context;
4
5  public class FlagstaffHill {
6     public static native String cilantro(String var0);
7
8     public static String getFlag(String var0, Context var1) {
9        return nope(var0);
10    }
11
12    public static String nope(String var0) {
13       return "don't wanna";
14    }
15
16    public static String yep(String var0) {
17       return cilantro(var0);
18    }
19 }
20
```

Vào code smali đổi nơi gọi hàm nope hành yep

```
    .line 19
    invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;→yep(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v0

    .line 20
    .local v0, "flag":Ljava/lang/String;
    return-object v0
.end method

.method public static nope(Ljava/lang/String;)Ljava/lang/String;
    .locals 1
    .param p0, "input"    # Ljava/lang/String;
```

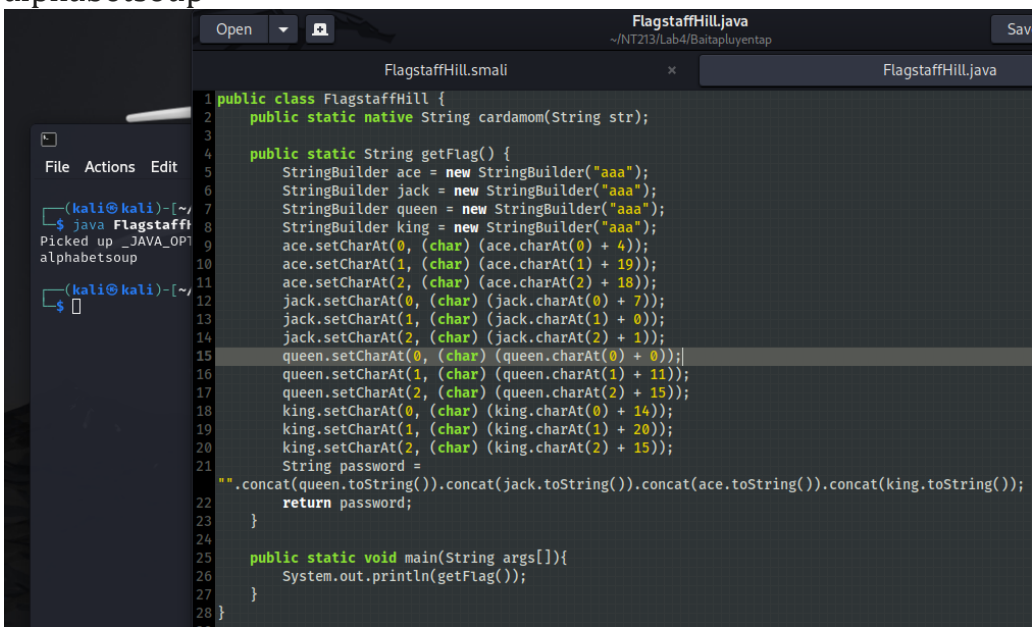
Build, ký và chạy lại. Nhập bất kỳ và có flag



⇨ **picoCTF{tis.but.a.scratch}**

Five.apk

Ứng dụng sẽ thực hiện tạo password tại FlagstaffHill với code bên dưới, ta được alphabetsoup



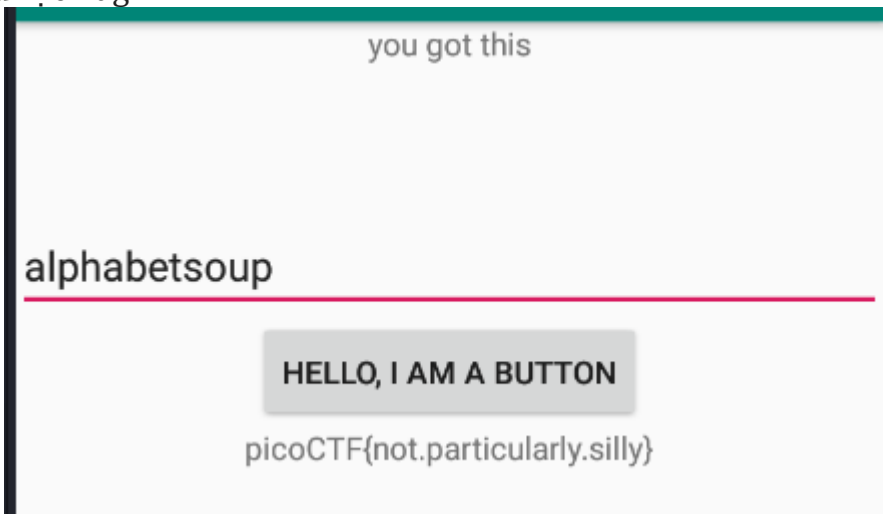
Tuy nhiên nó chỉ cho ra kết quả là “call it” thay vì gọi tới cardamom chứa flag

```
ntext;

l {
tring cardamom(String var0);

etFlag(String var0, Context var1) {
 = new StringBuilder("aaa");
 = new StringBuilder("aaa");
 = new StringBuilder("aaa");
 = new StringBuilder("aaa");
char)(var3.charAt(0) + 4));
char)(var3.charAt(1) + 19));
char)(var3.charAt(2) + 18));
char)(var4.charAt(0) + 7));
char)(var4.charAt(1) + 0));
char)(var4.charAt(2) + 1));
char)(var5.charAt(0) + 0));
char)(var5.charAt(1) + 11));
char)(var5.charAt(2) + 15));
char)(var2.charAt(0) + 14));
char)(var2.charAt(1) + 20));
char)(var2.charAt(2) + 15));
"".concat(var5.toString()).concat(var4.toString()).concat(var3.toString()).concat(var2.toString())) ? "call it" : "NOPE";
```

Ta thực hiện sửa code smali sao cho nó sẽ gọi đến cardamom. Thực hiện build và ký lại

```
220     move-result-object v5
221
222     invoke-virtual {v4, v5}, Ljava/lang/String;→concat(Ljava/lang/String;)Ljava/lang/String;
223
224     move-result-object v4
225
226     .line 36
227     .local v4, "password":Ljava/lang/String;
228     invoke-virtual {p0, v4}, Ljava/lang/String;→equals(Ljava/lang/Object;)Z
229
230     move-result v5
231
232     invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;→cardamom(Ljava/lang/String;)Ljava/lang/String;
233
234     move-result-object v0
235
236     return-object v0
237
```

Cuối cùng ta được flag:



⇨ **picoCTF{not.particularly.silly}**
