

# BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng

Kỳ báo cáo: Lab 05

Tên chủ đề: Ôn tập

GV: Ngô Khánh Khoa

Ngày báo cáo: 30/05/2023

## 1. THÔNG TIN CHUNG:

Lớp: NT213.N21.ANTT.2

STT	Họ và tên	MSSV	Email
1	Phạm Phúc Đức	20520162	20520162@gm.uit.edu.vn

## 2. NỘI DUNG THỰC HIỆN:<sup>1</sup>

STT	Công việc	Kết quả tự đánh giá
1	<a href="#">Challenges 1</a>	11/12
2	<a href="#">Challenges 2</a>	100%

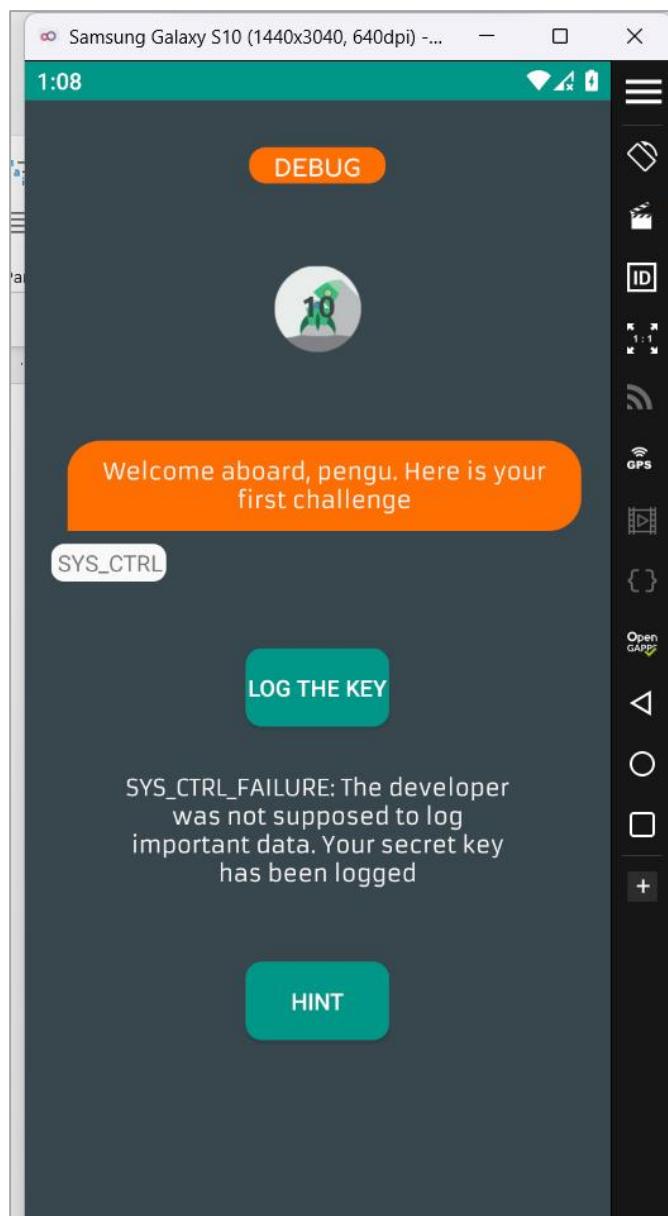
Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

# BÁO CÁO CHI TIẾT

## 1. Challenges 1

### 1.1. Level 1 – Debug me



Hình 1: level 1

Level yêu cầu chúng ta tìm đọc log nên chúng ta sẽ đọc nội dung từ logcat:

```

PS C:\Program Files\Genymobile\Genymotion\tools> .\adb shell ps | grep 'evabs'
u0_a162      3484    298 12808156 177408 ep_poll   75119dc3390a S com.revo.evabs
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb logcat --pid 3484
----- beginning of main
06-14 01:22:28.391 3484 3511 I OpenGLRenderer: Davey! duration=1291ms; Flags=1, IntendedVsync=1074205890484, Vsync=1074205890484, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1074206501750, AnimationStart=1074206536392, PerformTraversalsStart=1074206583325, DrawStart=1074531220674, SyncQueued=1074547215724, SyncStart=1074718354234, IssueDrawCommandsStart=1074718835580, SwapBuffers=1075659273820, FrameCompleted=1075668290329, DequeueBufferDuration=370438, QueueBufferDuration=680813, GpuCompleted=0,
06-14 01:22:28.393 3484 3484 I Choreographer: Skipped 86 frames! The application may be doing too much work on its main thread.
06-14 01:22:28.418 3484 3511 I OpenGLRenderer: Davey! duration=1470ms; Flags=0, IntendedVsync=1074222557150, Vsync=1075655890426, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1075670819142, AnimationStart=1075670849593, PerformTraversalsStart=1075671073365, DrawStart=1075671139015, SyncQueued=1075671374241, SyncStart=1075672497847, IssueDrawCommandsStart=1075672691727, SwapBuffers=1075684738789, FrameCompleted=1075693949177, DequeueBufferDuration=477714, QueueBufferDuration=632762, GpuCompleted=0,
06-14 01:23:30.624 3484 3511 I OpenGLRenderer: Davey! duration=1007ms; Flags=1, IntendedVsync=1136822554646, Vsync=1136889221310, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1136900915977, AnimationStart=1136900943634, PerformTraversalsStart=1136900983304, DrawStart=1137257736467, SyncQueued=1137261489458, SyncStart=1137332927588, IssueDrawCommandsStart=1137332985696, SwapBuffers=1137894072091, FrameCompleted=1137901316866, DequeueBufferDuration=343619, QueueBufferDuration=660140, GpuCompleted=0,
06-14 01:23:30.627 3484 3484 I Choreographer: Skipped 59 frames! The application may be doing too much work on its main thread.
06-14 01:23:30.650 3484 3511 I OpenGLRenderer: Davey! duration=1020ms; Flags=0, IntendedVsync=1136905887976, Vsync=1137889221270, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1137904739927, AnimationStart=1137904780435, PerformTraversalsStart=1137905012308, DrawStart=1137905082708, SyncQueued=1137906210784, SyncStart=1137907660969, IssueDrawCommandsStart=1137907793667, SwapBuffers=1137914647052, FrameCompleted=1137927384984, DequeueBufferDuration=282997, QueueBufferDuration=762947, GpuCompleted=0,
06-14 01:32:27.584 3484 3511 I OpenGLRenderer: Davey! duration=1369ms; Flags=1, IntendedVsync=1673305866520, Vsync=167330199852, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1673344777356, AnimationStart=167344809204, PerformTraversalsStart=1673344860607, DrawStart=1673678271418, SyncQueued=1673685824612, SyncStart=1673872894198, IssueDrawCommandsStart=1673873385322, SwapBuffers=1674854390450, FrameCompleted=1674861999517, DequeueBufferDuration=438603, QueueBufferDuration=593371, GpuCompleted=0,
06-14 01:32:27.587 3484 3484 I Choreographer: Skipped 90 frames! The application may be doing too much work on its main thread.
06-14 01:32:29.877 3484 3499 I com.revo.evabs: JIT allocated 76KB for compiled code of void android.view.View.<init>(android.content.Context, android.util.AttributeSet, int, int)
06-14 01:33:11.320 3484 3511 I OpenGLRenderer: Davey! duration=1767ms; Flags=1, IntendedVsync=1716689198118, Vsync=1716689198118, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1716691681218, AnimationStart=1716691730945, PerformTraversalsStart=1716691791567, DrawStart=1716987740633, SyncQueued=1716999292939, SyncStart=1717140960386, IssueDrawCommandsStart=1717141018773, SwapBuffers=1718589885878, FrameCompleted=1718598088876, DequeueBufferDuration=339988, QueueBufferDuration=667403, GpuCompleted=60074497276,
06-14 01:33:11.324 3484 3484 I Choreographer: Skipped 113 frames! The application may be doing too much work on its main thread.
06-14 01:33:15.259 3484 3484 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
|
```

Hình 2: Xem logcat của ứng dụng bằng pid

Phía dưới cùng:

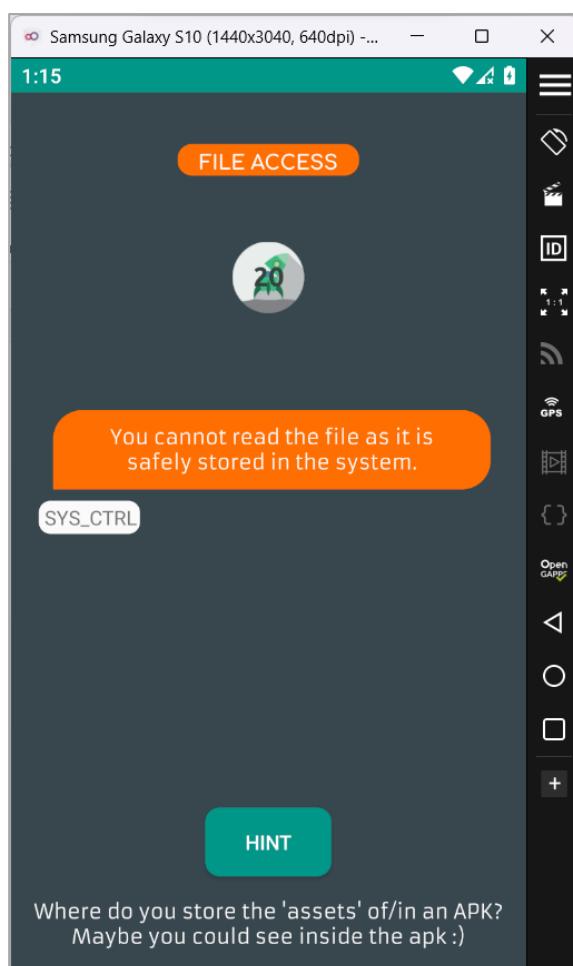
```

06-14 01:23:30.650 3484 3511 I OpenGLRenderer: Davey! duration=1020ms; Flags=0, IntendedVsync=1136905887976, Vsync=1137889221270, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1137904739927, AnimationStart=1137904780435, PerformTraversalsStart=1137905012308, DrawStart=1137905082708, SyncQueued=1137906210784, SyncStart=1137907660969, IssueDrawCommandsStart=1137907793667, SwapBuffers=1137914647052, FrameCompleted=1137927384984, DequeueBufferDuration=282997, QueueBufferDuration=762947, GpuCompleted=0,
06-14 01:32:27.584 3484 3511 I OpenGLRenderer: Davey! duration=1369ms; Flags=1, IntendedVsync=1673305866520, Vsync=167330199852, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1673344777356, AnimationStart=167344809204, PerformTraversalsStart=1673344860607, DrawStart=1673678271418, SyncQueued=1673685824612, SyncStart=1673872894198, IssueDrawCommandsStart=1673873385322, SwapBuffers=1674854390450, FrameCompleted=1674861999517, DequeueBufferDuration=438603, QueueBufferDuration=593371, GpuCompleted=0,
06-14 01:32:27.587 3484 3499 I com.revo.evabs: JIT allocated 76KB for compiled code of void android.view.View.<init>(android.content.Context, android.util.AttributeSet, int, int)
06-14 01:33:11.320 3484 3511 I OpenGLRenderer: Davey! duration=1767ms; Flags=1, IntendedVsync=1716689198118, Vsync=1716689198118, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1716691681218, AnimationStart=1716691730945, PerformTraversalsStart=1716691791567, DrawStart=1716987740633, SyncQueued=1716999292939, SyncStart=1717140960386, IssueDrawCommandsStart=1717141018773, SwapBuffers=1718589885878, FrameCompleted=1718598088876, DequeueBufferDuration=339988, QueueBufferDuration=667403, GpuCompleted=60074497276,
06-14 01:33:11.324 3484 3484 I Choreographer: Skipped 113 frames! The application may be doing too much work on its main thread.
06-14 01:33:15.259 3484 3484 D ** SYS_CTRL **: : EVABS{logging_info_never_safe}
|
```

Hình 3: Flag

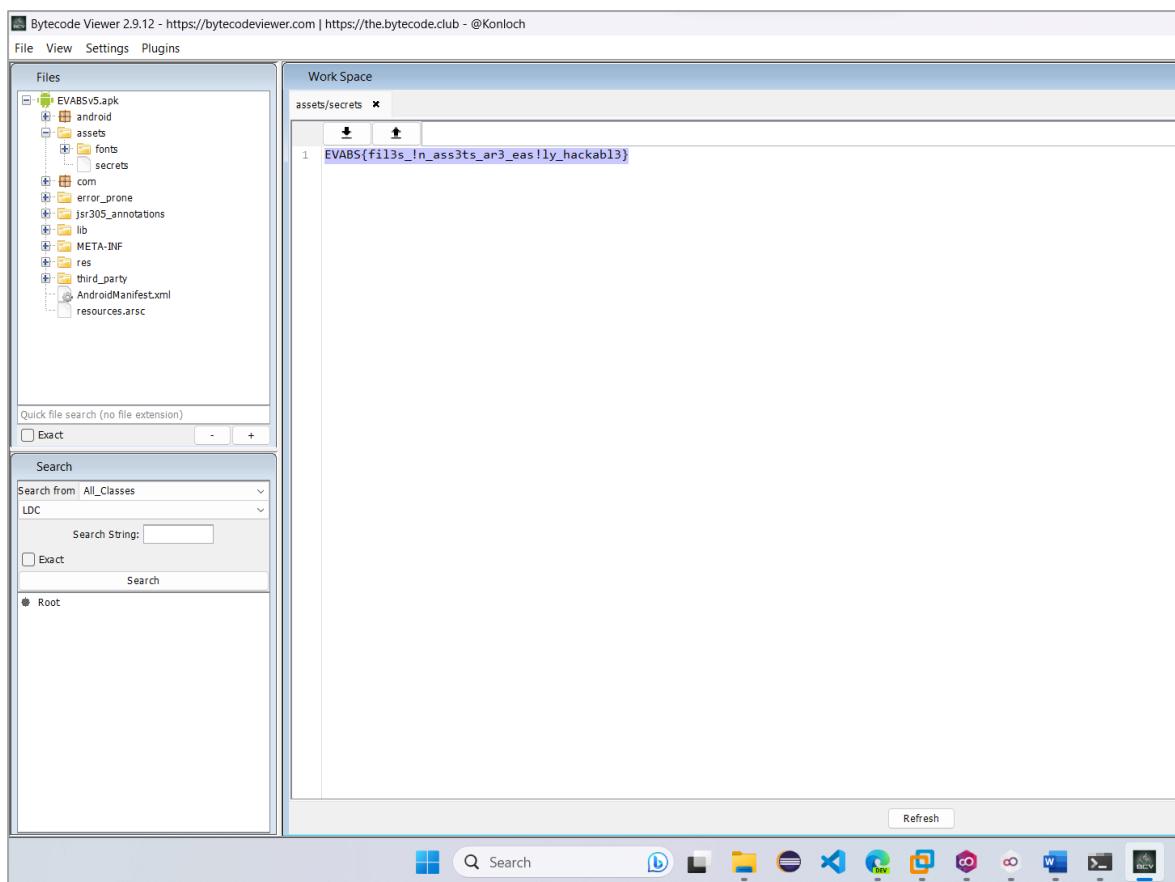
➔ EVABS{logging\_info\_never\_safe}

## 1.2. Level 2 – File Access



Hình 4: Gợi ý câu 2

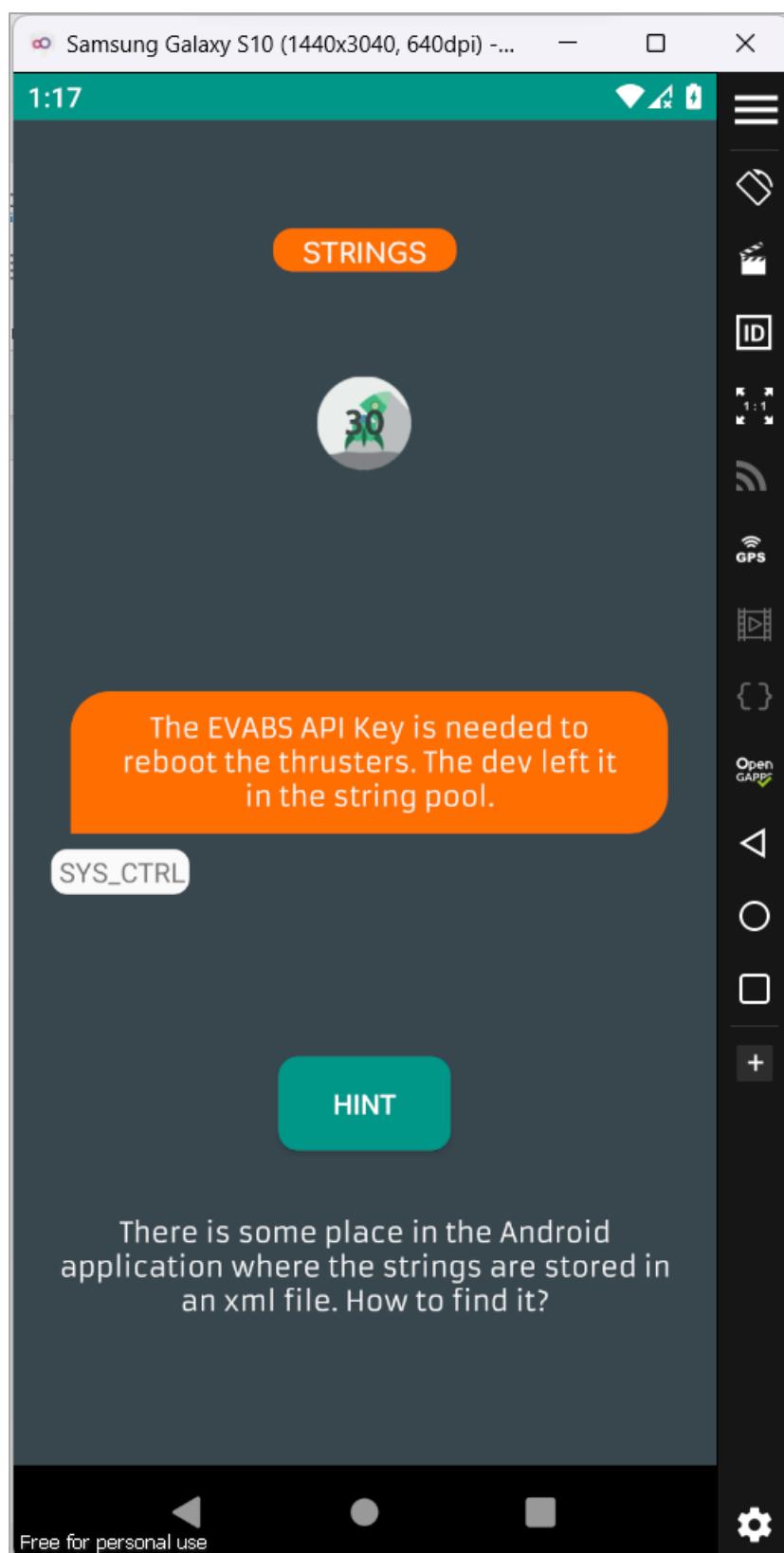
Theo như gợi ý, chúng ta sẽ đi tìm trong tệp assets khi decompile tệp .apk:



Hình 5: Tìm được flag

➔ EVABS{fil3s\_!n\_ass3ts\_ar3\_eas!ly\_hackabl3}

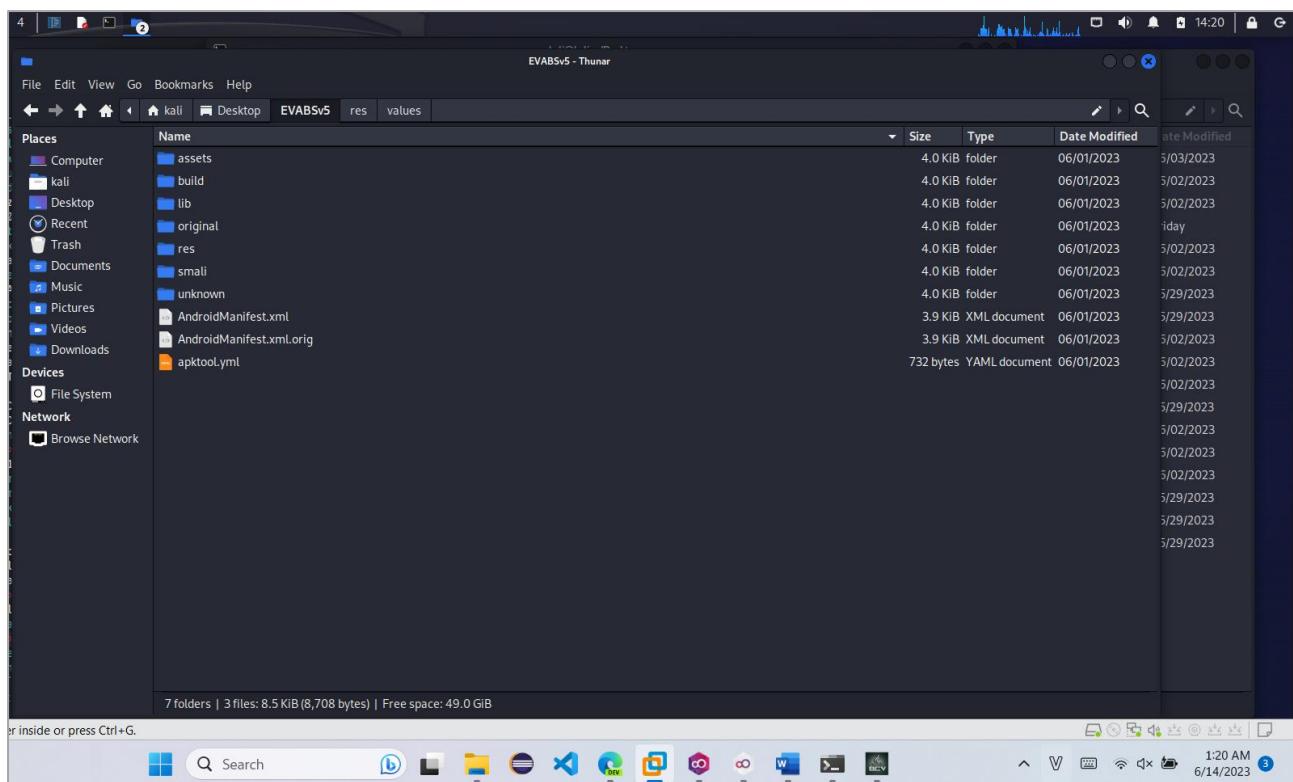
### 1.3. Level 3 - String



Hình 6: Gợi ý level 3

Gợi ý chỉ ra flag nằm trong tệp xml chứa các chuỗi -> có thể là tệp string.xml và nó luôn nằm trong thư mục values

Sử dụng apktool của kali để decompile (ByteCode Viewer không xem được tệp values)



Hình 7: Decompile

Truy cập tới res/values/string.xml:

```

97      <string name="status_bar_notification_info_overflow">999+</string>
98      <string name="the_evabs_api_key">EVABS{saf3ly_st0red_in_Strings?}</string>
99      <string name="title_activity_home">Home</string>
100     <string name="title_activity_launch">Launch</string>
101     <string name="title_activity_login">Sign in</string>
102     <string name="title_activity_splash">Splash</string>
103     <string name="title_activity_test">Test</string>
104 </resources>
105

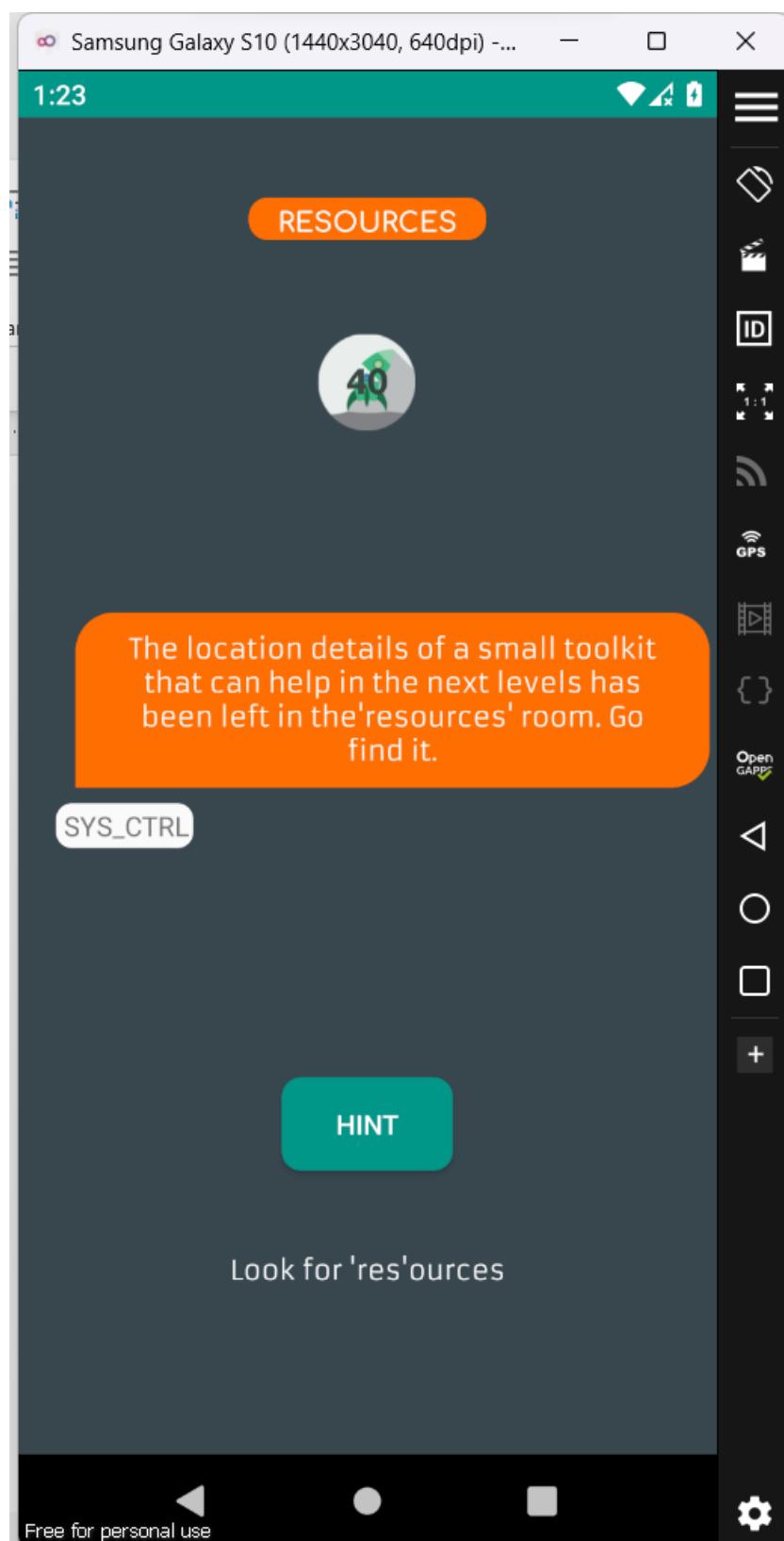
```

x api| ↕ Match case Regular expression 5 occurrences

Hình 8: Tìm kiếm bằng từ khóa "api"

→ EVABS{saf3ly\_st0red\_in\_Strings?}

#### 1.4. Level 4 - Resources



Hình 9: Level 4

Resource đang nói đến cũng là tệp res, chúng ta tìm tới file có chứa “small toolkit” kia.

Theo như phân tích bằng BCV, tệp res\_raw.class (gồm cả \$1) sẽ liên quan tới level này:

```

File View Settings Plugins
Files
Work Space
JD-GUI Decompiler - Editable: false
package com.revo.evabs;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.TextView;
class Res_raw$1
implements View.OnClickListener
{
    Res_raw$1(Res_raw paramRes_raw, TextView paramTextView) {}
    public void onClick(View paramView)
    {
        this.val$tv.setText("Look for 'res'ources");
    }
}

```

Hình 10: Nội dung res\_raw\$1.class

Dựa theo tên lớp, truy cập vào tệp res/raw:

```

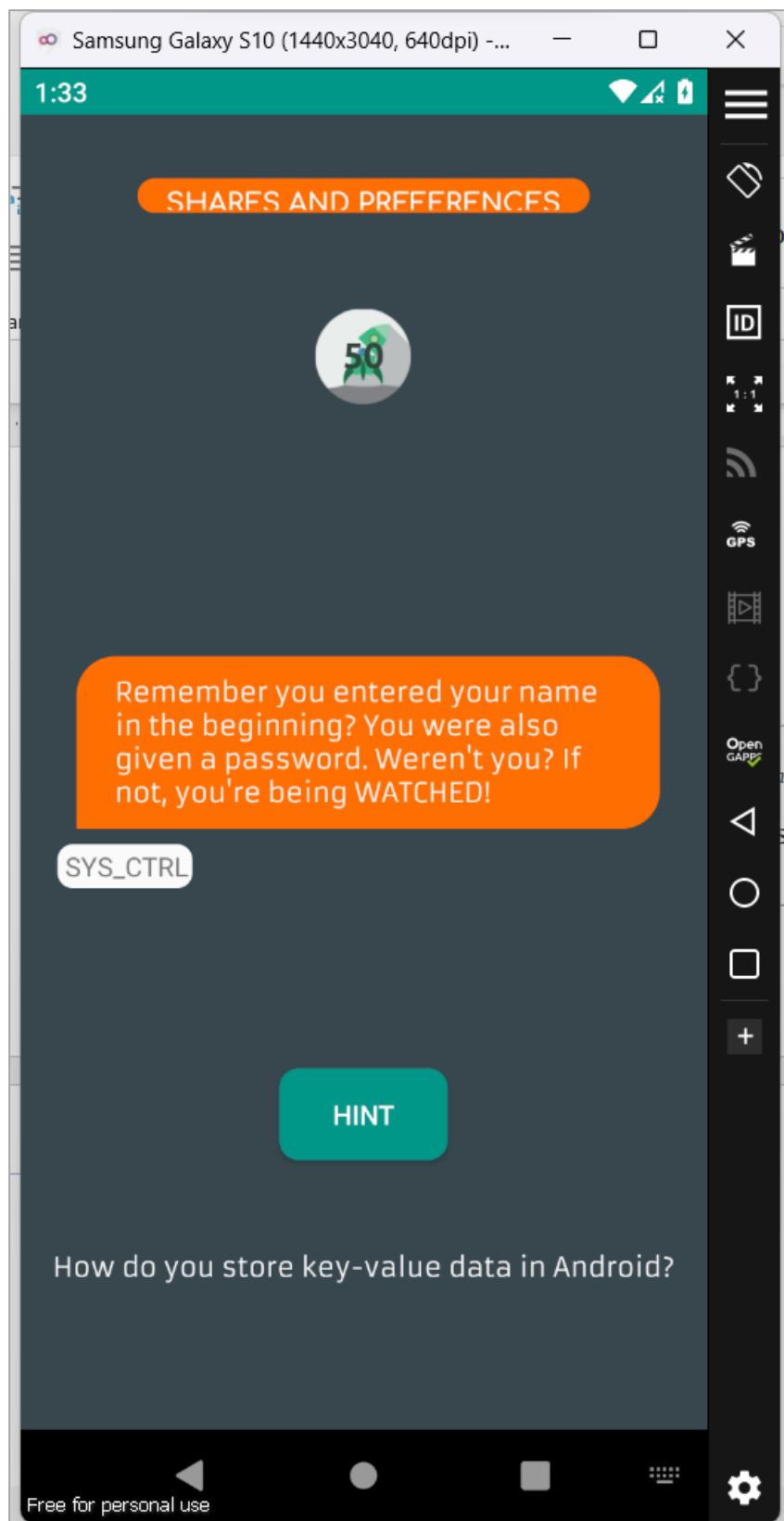
Bytecode Viewer 2.9.12 - https://bytecodeviewer.com | https://the.bytecode.club - @Konloch
File View Settings Plugins
Files
Work Space
com/revo/evabs/Res_raw$1.class x com/revo/evabs/Res_raw.class x com/revo/evabs/R$raw.class x res/raw/link.txt x
www.github.com/abhi-r3v0/Adhrit
# This Toolkit will help you fix EVABS
EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}

```

Hình 11: Tìm thấy flag

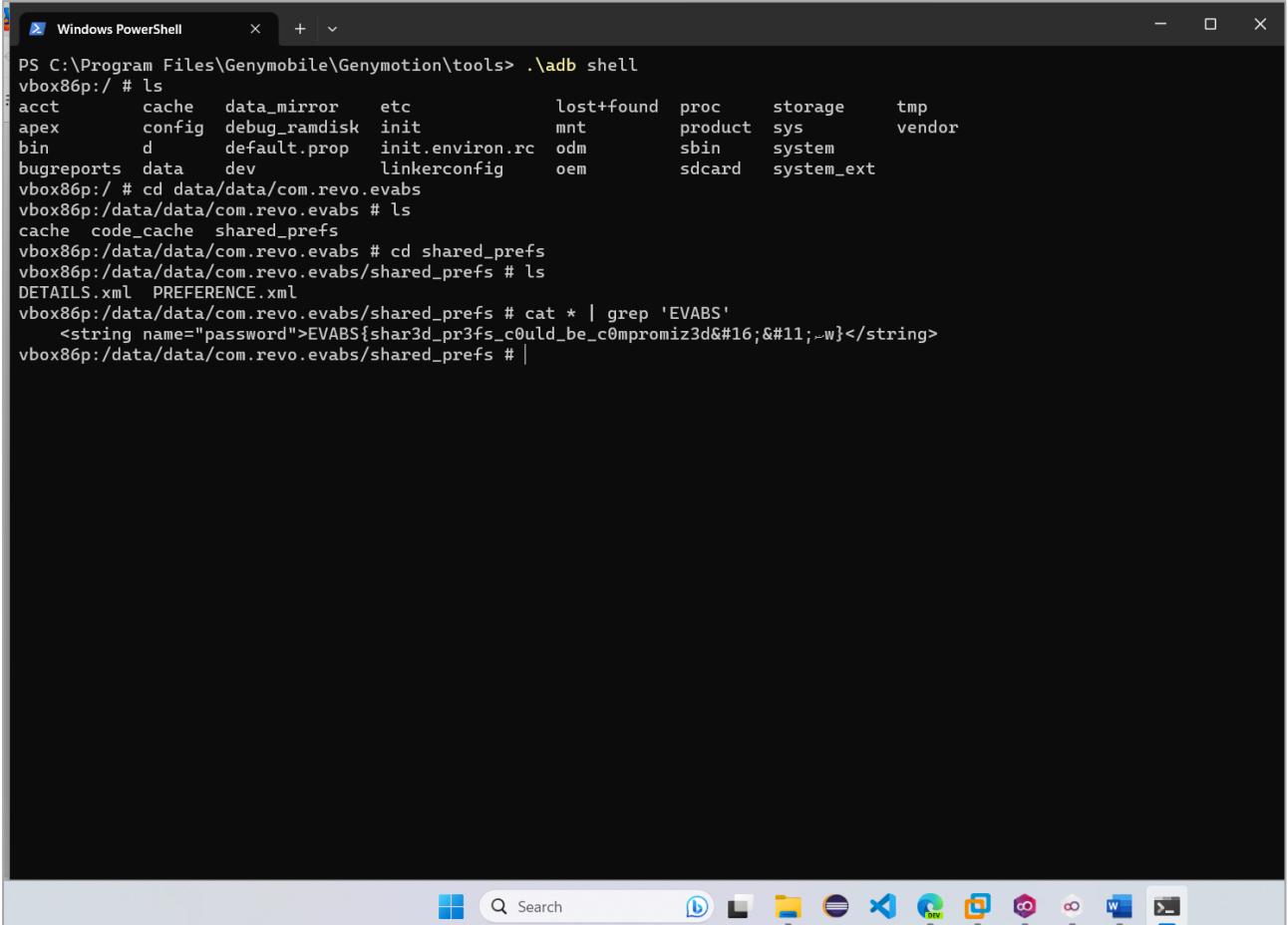
➔ EVABS{th!s\_plac3\_is\_n0t\_as\_s3cur3\_as\_it\_l00ks}

### **1.5. Level 5 – Shares and Preferences**



Hình 12: Level 5

Thường thì dữ liệu các ứng dụng sẽ được lưu vào data/data/<app package> nên chúng ta sẽ mở các file trong đó để tìm mật khẩu.



```

Windows PowerShell
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb shell
vbox86p:/ # ls
acct      cache  data_mirror  etc          lost+found  proc    storage   tmp
apex      config  debug_ramdisk  init        mnt       product  sys     vendor
bin       d       default.prop  init.environ.rc  odm      sbin     system
bugreports data    dev           linkerconfig  oem      sdcard   system_ext
vbox86p:/ # cd data/data/com.revo.evabs
vbox86p:/data/data/com.revo.evabs # ls
cache  code_cache shared_prefs
vbox86p:/data/data/com.revo.evabs # cd shared_prefs
vbox86p:/data/data/com.revo.evabs/shared_prefs # ls
DETAILS.xml  PREFERENCE.xml
vbox86p:/data/data/com.revo.evabs/shared_prefs # cat * | grep 'EVABS'
<string name="password">EVABS{shar3d_pr3fs_c0uld_be_c0mpromiz3d&#16;‐w}</string>
vbox86p:/data/data/com.revo.evabs/shared_prefs #

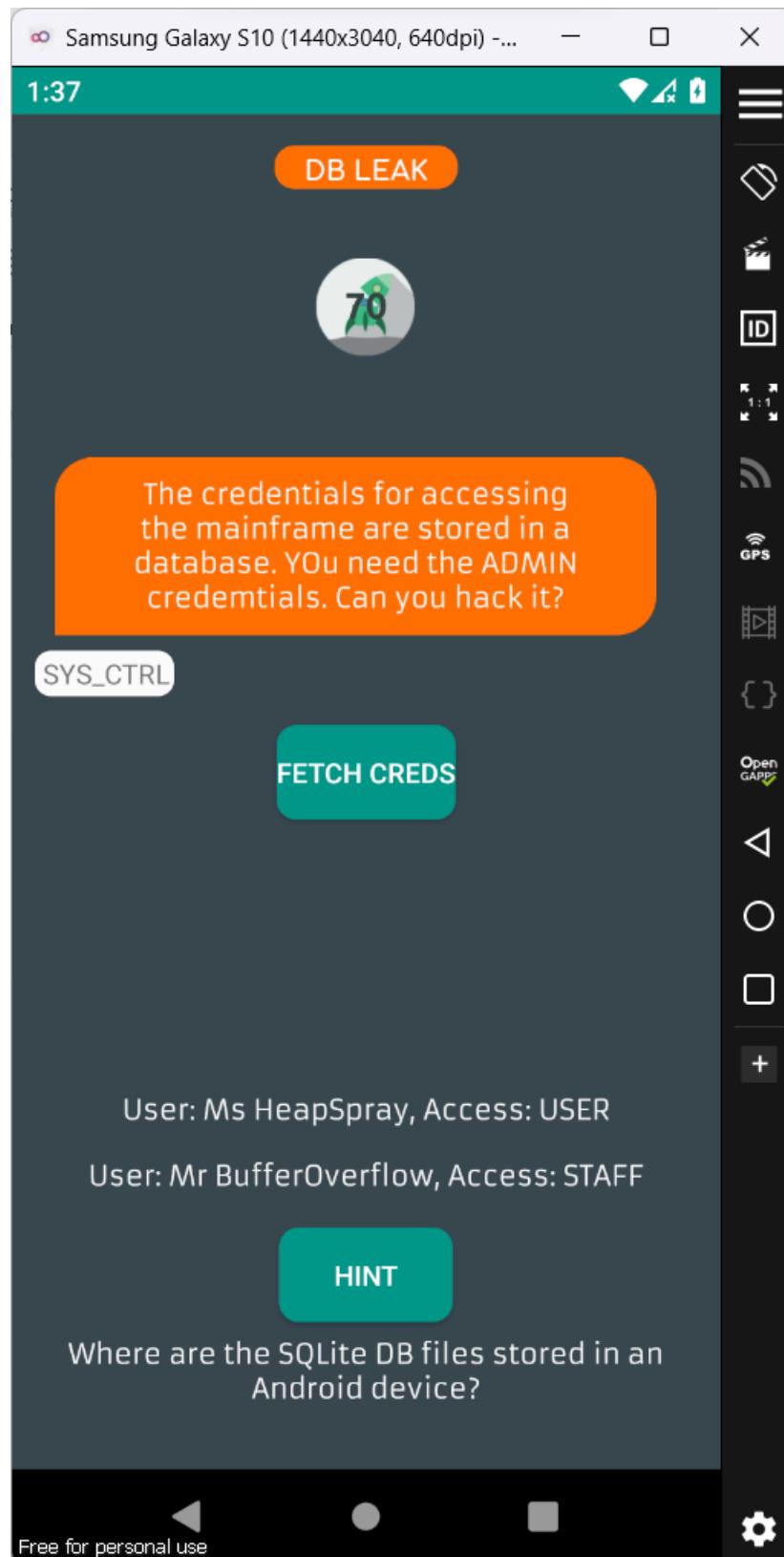
```

Hình 13: Lấy flag

Hai tệp đầu là cache và code\_cache là 2 tệp thường thấy và log sẽ không ghi vào đây, chúng ta sẽ mở file shared\_prefs và lấy key.

➔ Flag: EVABS{shar3d\_pr3fs\_c0uld\_be\_c0mpromiz3d&#16;‐w}

### 1.6. Level 6 – DB leak



Hình 14: Level 6

Level 6 yêu cầu chúng ta truy cập vào SQLite trong hệ thống, thực hiện truy cập lại vào data/data/com.revo.evabs:

```
vbox86p:/data/data/com.revo.evabs # ls
cache code_cache databases shared_prefs
vbox86p:/data/data/com.revo.evabs # cat database
cat: database: No such file or directory
1|vbox86p:/data/data/com.revo.evabs # sqlite3 databases
Error: unable to open database "databases": unable to open database file
1|vbox86p:/data/data/com.revo.evabs # cat databases
cat: databases: Is a directory
1|vbox86p:/data/data/com.revo.evabs # ls
cache code_cache databases shared_prefs
vbox86p:/data/data/com.revo.evabs # ls databases
MAINFRAME_ACCESS MAINFRAME_ACCESS-journal
vbox86p:/data/data/com.revo.evabs # sqlite3 MAINFRAME_ACCESS
SQLite version 3.28.0 2020-05-06 18:46:38
Enter ".help" for usage hints.
sqlite> .tables
sqlite> .exit
vbox86p:/data/data/com.revo.evabs # cd databases
vbox86p:/data/data/com.revo.evabs/databases # sqlite3 MAINFRAME_ACCESS
SQLite version 3.28.0 2020-05-06 18:46:38
Enter ".help" for usage hints.
sqlite> .tables
CREDS          android_metadata
sqlite> SELECT * FROM CREDS
...> .exit
...> exit
...> exit
...> exit
1|vbox86p:/data/data/com.revo.evabs/databases # sqlite3 MAINFRAME_ACCESS
SQLite version 3.28.0 2020-05-06 18:46:38
Enter ".help" for usage hints.
sqlite> .tables
CREDS          android_metadata
sqlite> SELECT * FROM CREDS;
Dr.l33t|EVABS{sqlite_is_not_safe}@@
u|ADMIN
Mr BufferOverflow|0xNotSecureSQLite_|STAFF
Ms HeapSpray|SQLite_expl0it|USER
sqlite> |
```

Hình 15: Xem giá trị có trong bảng

Chúng ta có thể biết được nó nằm trong bảng CREDS dó khi phân tích mã nguồn, chúng ta thấy tài khoản Dr.l33t có quyền ADMIN:

```

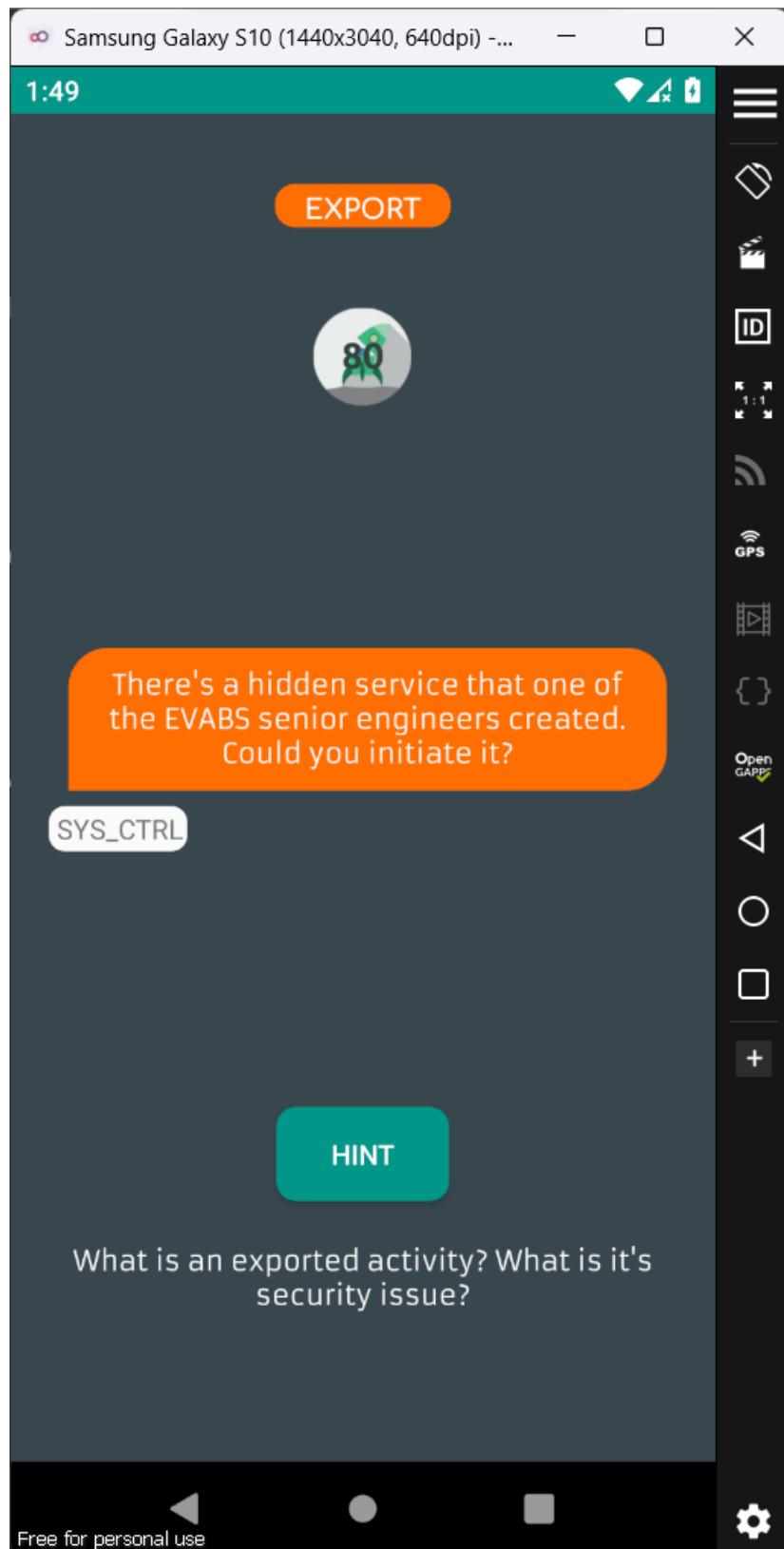
JD-GUI Decompiler - Editable: false
14 {
15     System.loadLibrary("native-lib");
16 }
17
18 protected void onCreate(Bundle paramBundle)
19 {
20     super.onCreate(paramBundle);
21     setContentView(2131492894);
22     String str1 = stringFromJNI();
23     SQLiteDatabase localSQLiteDatabase = openOrCreateDatabase("MAINFRAME_ACCESS", 0, null);
24     localSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS CREDS(admin VARCHAR, pass VARCHAR, access VARCHAR);");
25     StringBuilder localStringBuilder = new StringBuilder();
26     localStringBuilder.append("INSERT INTO CREDS VALUES('Dr.133t', '')");
27     localStringBuilder.append(str1);
28     localStringBuilder.append("' , 'ADMIN');");
29     localSQLiteDatabase.execSQL(localStringBuilder.toString());
30     localSQLiteDatabase.execSQL("INSERT INTO CREDS VALUES('Mr BufferOverflow', '0xNotSecureSQLite_', 'STAFF');");
31     localSQLiteDatabase.execSQL("INSERT INTO CREDS VALUES('Ms HeapSpray', 'SQLite_expl0it', 'USER');");
32     Cursor localCursor = localSQLiteDatabase.rawQuery("SELECT * FROM CREDS", null);
33     localCursor.moveToFirst();
34     String str2 = localCursor.getString(0);
35     String str3 = localCursor.getString(2);
36     localCursor.moveToFirst();
37     String str4 = localCursor.getString(0);
38     String str5 = localCursor.getString(2);
39     TextView localTextView1 = (TextView) findViewById(2131362078);
40     TextView localTextView2 = (TextView) findViewById(2131362079);
41     Button localButton = (Button) findViewById(2131361839);
42     DBLeak.1 local1 = new DBLeak.1(this, localTextView1, str2, str3, localTextView2, str4, str5);
43     localButton.setOnClickListener(local1);
44     TextView localTextView3 = (TextView) findViewById(2131362080);
45     ((Button) findViewById(2131361836)).setOnClickListener(new DBLeak.2(this, localTextView3));
46 }
47
48 public native String stringFromJNI();
49 }

```

Hình 16: Phân tích

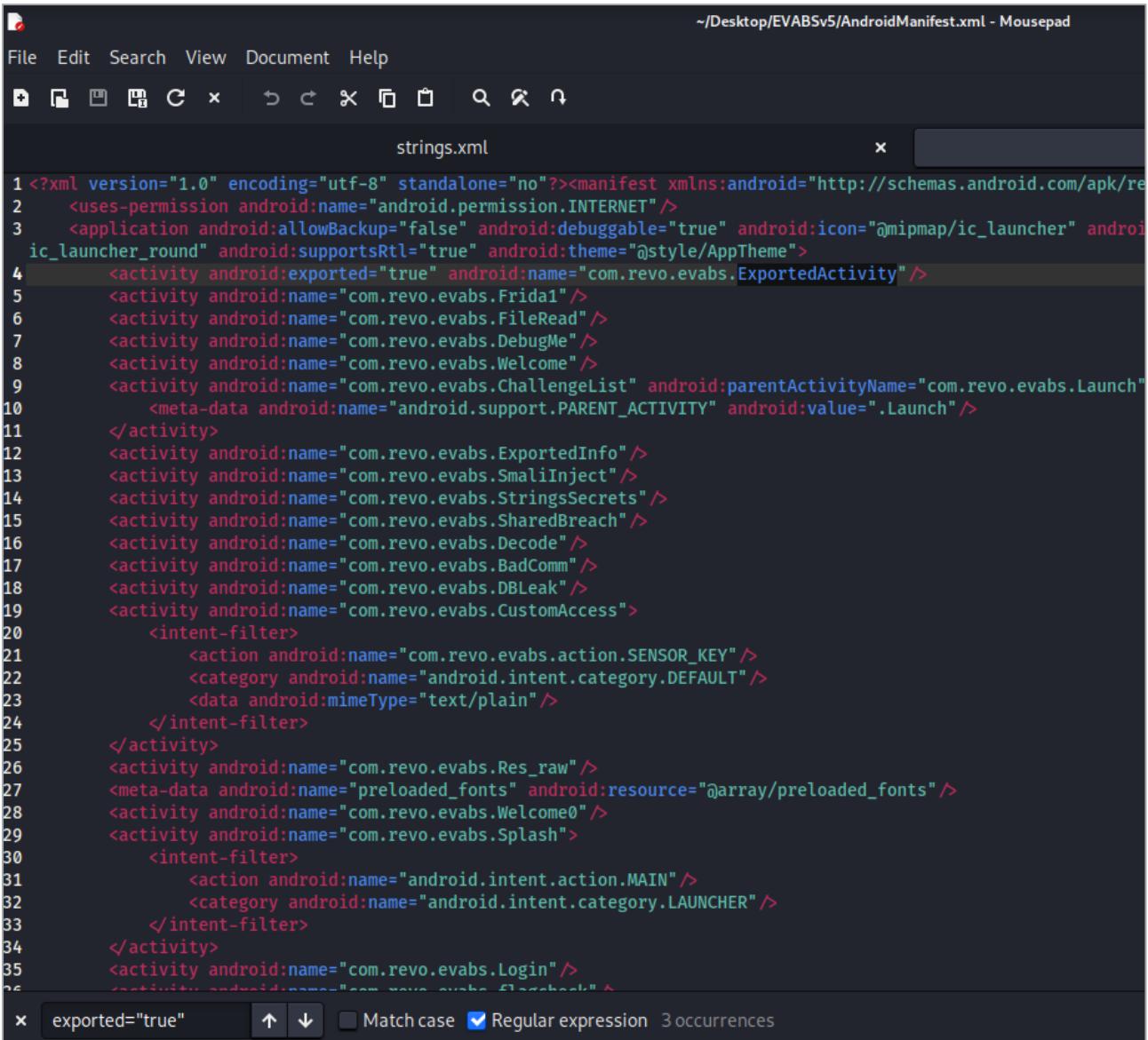
➔ flag: EVABS{sqlite\_is\_not\_safe}

### 1.7. Level 7 - Export



Hình 17: Yêu cầu level 7

Dựa theo gợi ý, chúng ta sẽ tìm tới activity có gán nhãn là exported trong tệp Manifest.xml:



```

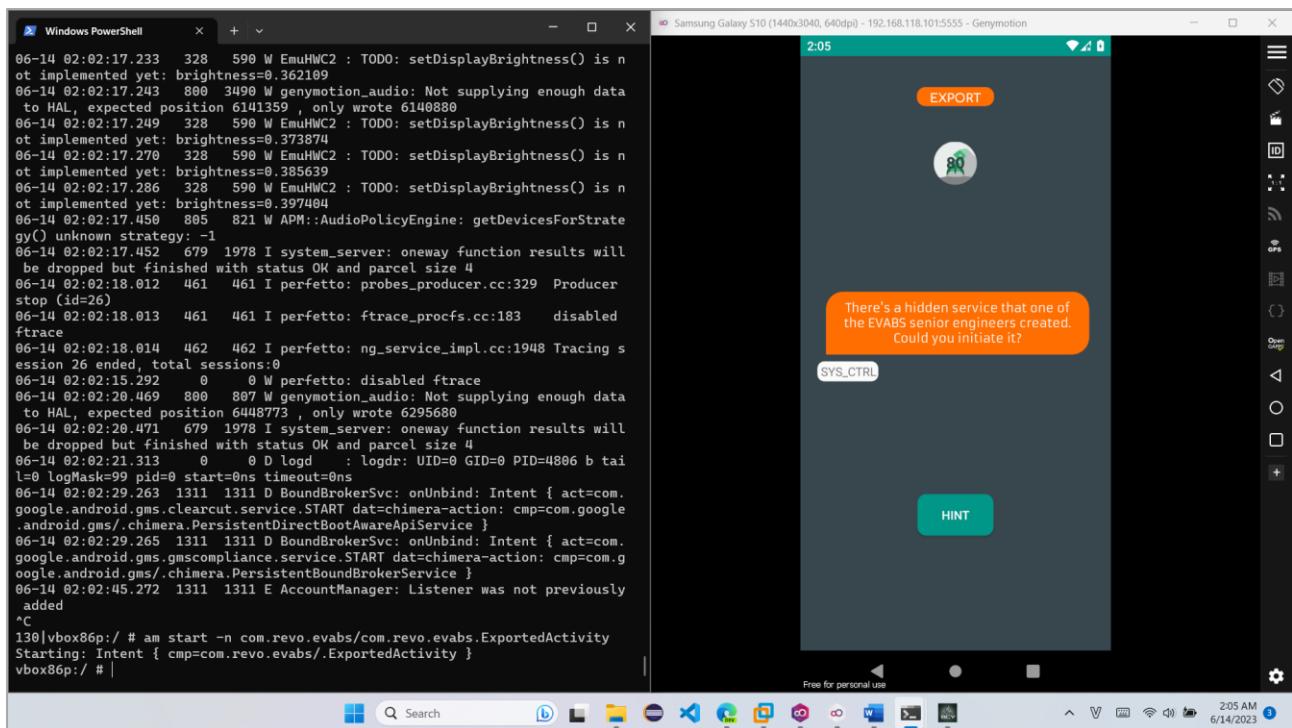
1 <?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/re
2     <uses-permission android:name="android.permission.INTERNET"/>
3     <application android:allowBackup="false" android:debuggable="true" android:icon="@mipmap/ic_launcher" android:i
ic_launcher_round" android:supportsRtl="true" android:theme="@style/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"/>
8         <activity android:name="com.revo.evabs.Welcome"/>
9         <activity android:name="com.revo.evabs.ChallengeList" android:parentActivityName="com.revo.evabs.Launch"
10            <meta-data android:name="android.support.PARENT_ACTIVITY" android:value=".Launch"/>
11        </activity>
12        <activity android:name="com.revo.evabs.ExportedInfo"/>
13        <activity android:name="com.revo.evabs.SmalInject"/>
14        <activity android:name="com.revo.evabs.StringsSecrets"/>
15        <activity android:name="com.revo.evabs.SharedBreach"/>
16        <activity android:name="com.revo.evabs.Decode"/>
17        <activity android:name="com.revo.evabs.BadComm"/>
18        <activity android:name="com.revo.evabs.DBLeak"/>
19        <activity android:name="com.revo.evabs.CustomAccess">
20            <intent-filter>
21                <action android:name="com.revo.evabs.actionSENSOR_KEY"/>
22                <category android:name="android.intent.category.DEFAULT"/>
23                <data android:mimeType="text/plain"/>
24            </intent-filter>
25        </activity>
26        <activity android:name="com.revo.evabs.Res_raw"/>
27        <meta-data android:name="preloaded_fonts" android:resource="@array/preloaded_fonts"/>
28        <activity android:name="com.revo.evabs.Welcome0"/>
29        <activity android:name="com.revo.evabs.Splash">
30            <intent-filter>
31                <action android:name="android.intent.action.MAIN"/>
32                <category android:name="android.intent.category.LAUNCHER"/>
33            </intent-filter>
34        </activity>
35        <activity android:name="com.revo.evabs.Login"/>
36    </application>

```

x **exported="true"**    Match case  Regular expression 3 occurrences

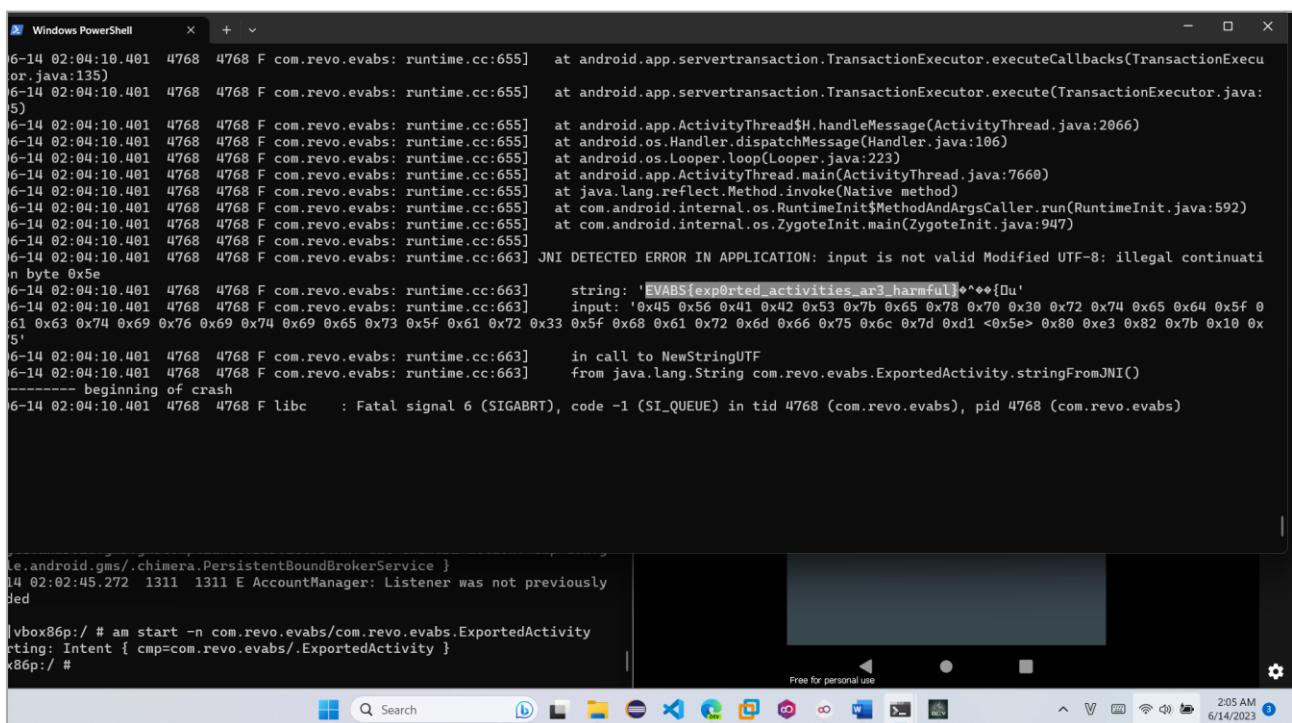
Hình 18: Kết quả tại dòng 4

Do nó được gán nhãn là exported="true" nên chúng ta sẽ thử Hijacking tới nó:



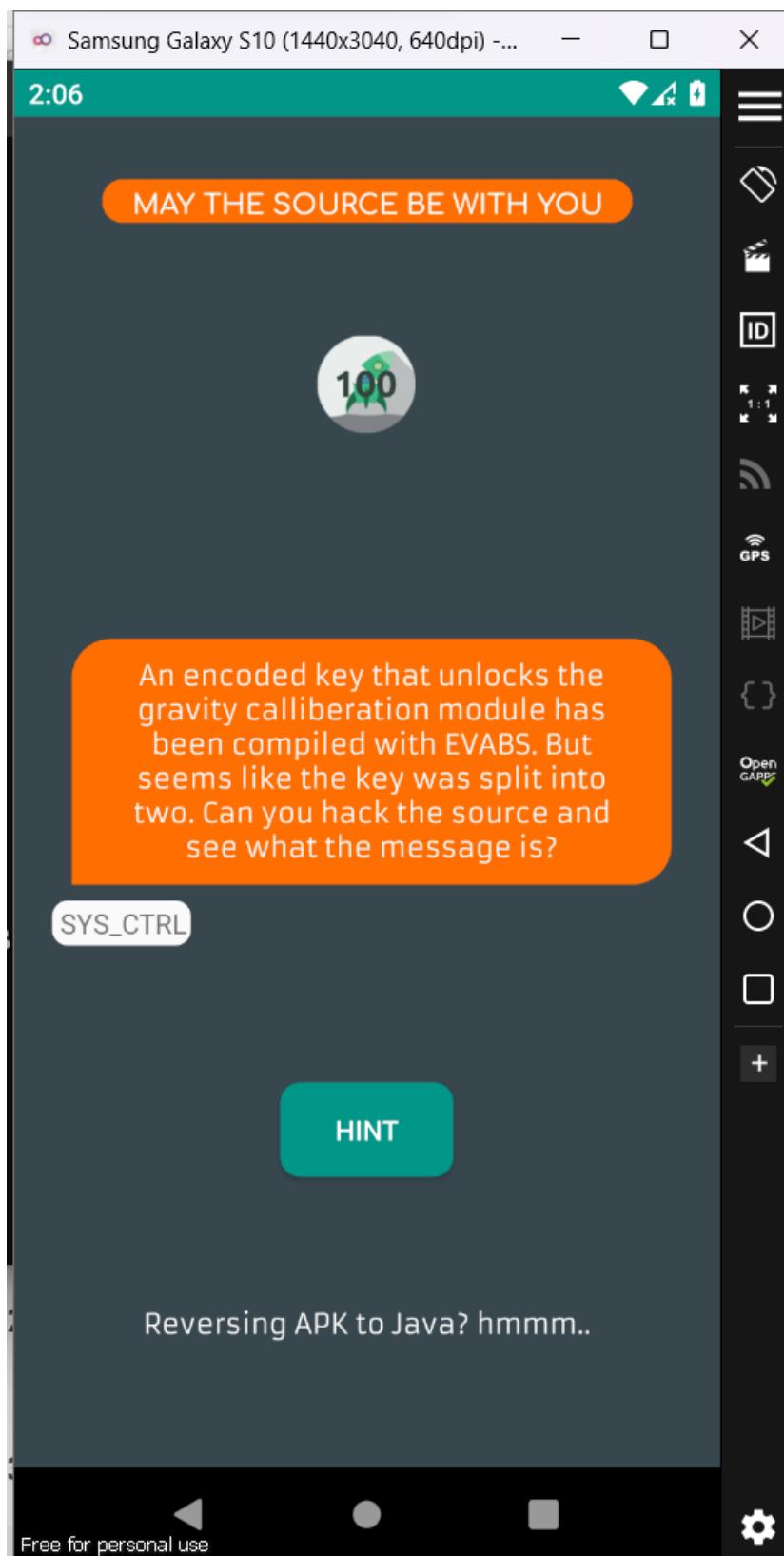
Hình 19: Không xuất hiện gì

Tuy nhiên khi xem lại logcat, chúng ta thấy:



➔ flag: EVABS{exp0rted\_activities\_ar3\_harmful}

### 1.8. Level 8 - Decode



Hình 20: Nội dung level 8

Yêu cầu của level 8 sẽ là đọc code sau đó lấy các chuỗi trong đó giải mã mà ghép chúng lại với nhau sẽ được flag:

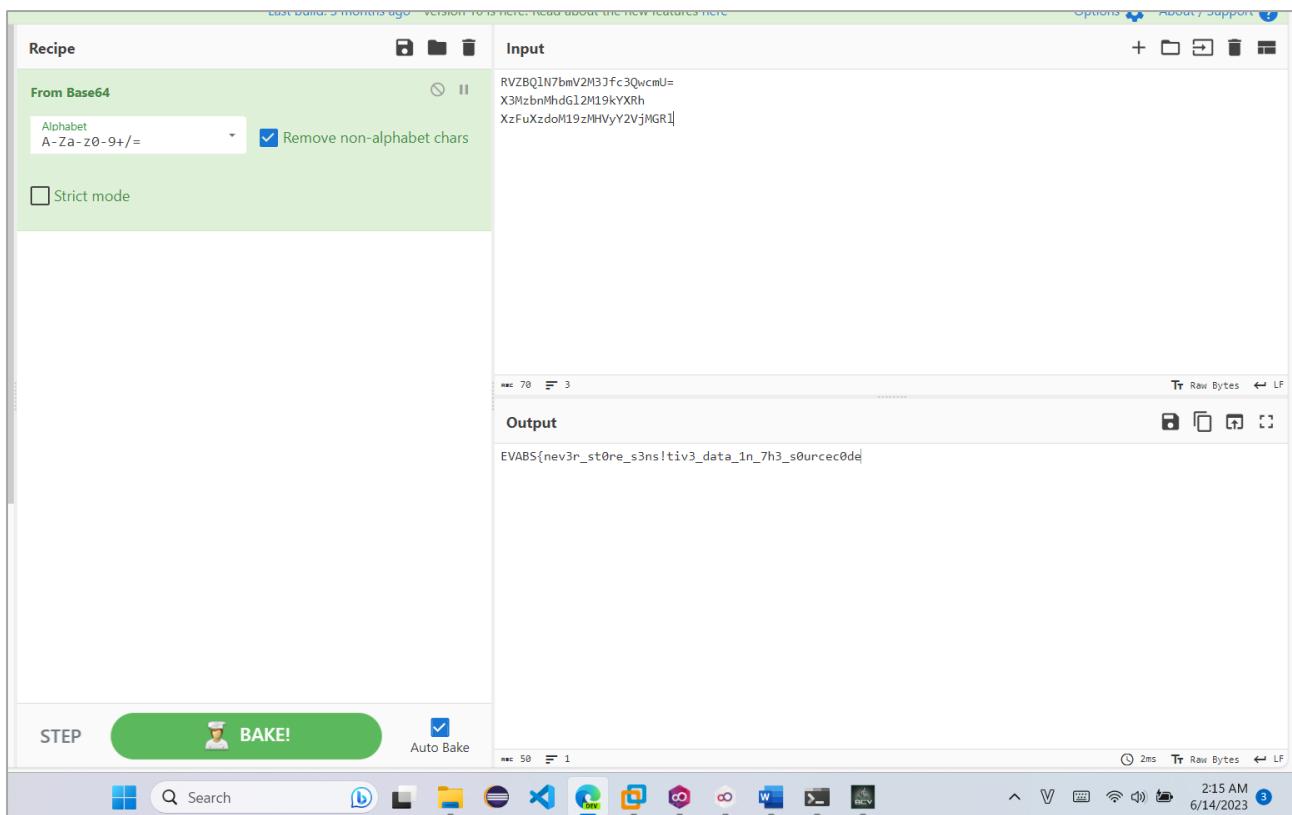
```

JD-GUI Decompiler - Editable: false
1 package com.revo.evabs;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.widget.Button;
6 import android.widget.TextView;
7
8 public class Decode
9     extends AppCompatActivity
10 {
11     protected void onCreate(Bundle savedInstanceState)
12     {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15         String localStringBuilder = new StringBuilder();
16         localStringBuilder.append("RVZBQ1N7bmV2M3Jfc3QwcmU=");
17         localStringBuilder.append("X3MzbnMhdGl2M19kYXRh");
18         localStringBuilder.append("XzFuXzdoM19zMHVyY2VjMGRL");
19         localStringBuilder.toString();
20         ((Button) findViewById(R.id.button)).setOnClickListener(new Decode.OnCreateListener(this, (TextView) findViewById(R.id.textView)));
21     }
22 }
23

```

Hình 21: Nội dung Decode.class

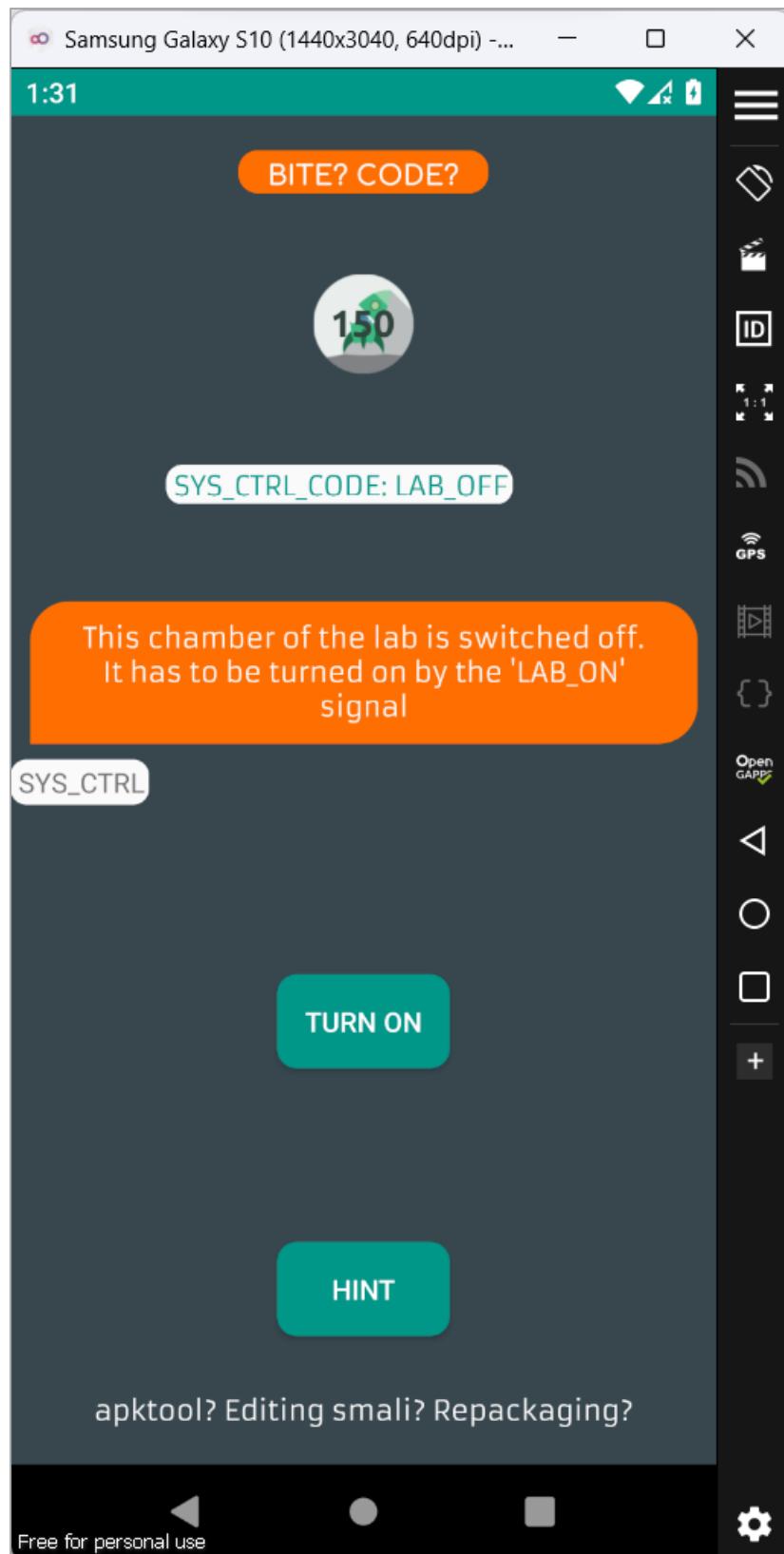
Chuỗi có dấu = nên có thể là base64:



Hình 22: Lấy được flag

→ EVABS{nev3r\_st0re\_s3ns!tiv3\_data\_1n\_7h3\_s0urcece0de}

### 1.9. Level 9 – Smali injection



Hình 23: Level 9

Level 9 yêu cầu chúng ta sửa code smali và sửa LAB\_OFF thành LAB\_ON để lấy flag.

Thực hiện sửa code:

```

23 .method public constructor <init>()V
24     .locals 1
25
26     .line 11
27     invoke-direct {p0}, Landroid/support/v7/app/AppCompatActivity;→<init>()V
28
29     .line 13
30     const-string v0, "LAB_OFF"
31
32     iput-object v0, p0, Lcom/revo/evabs/SmaliInject;→SIGNAL:Ljava/lang/String;
33
34     return-void
35 .end method

```

```

23 .method public constructor <init>()V
24     .locals 1
25
26     .line 11
27     invoke-direct {p0}, Landroid/support/v7/app/AppCompatActivity;→<init>()V
28
29     .line 13
30     const-string v0, " LAB_ON"
31
32     iput-object v0, p0, Lcom/revo/evabs/SmaliInject;→SIGNAL:Ljava/lang/String;
33
34     return-void
35 .end method

```

Hình 24: Sửa code smali

Cuối cùng re-build và ký tệp apk:

```

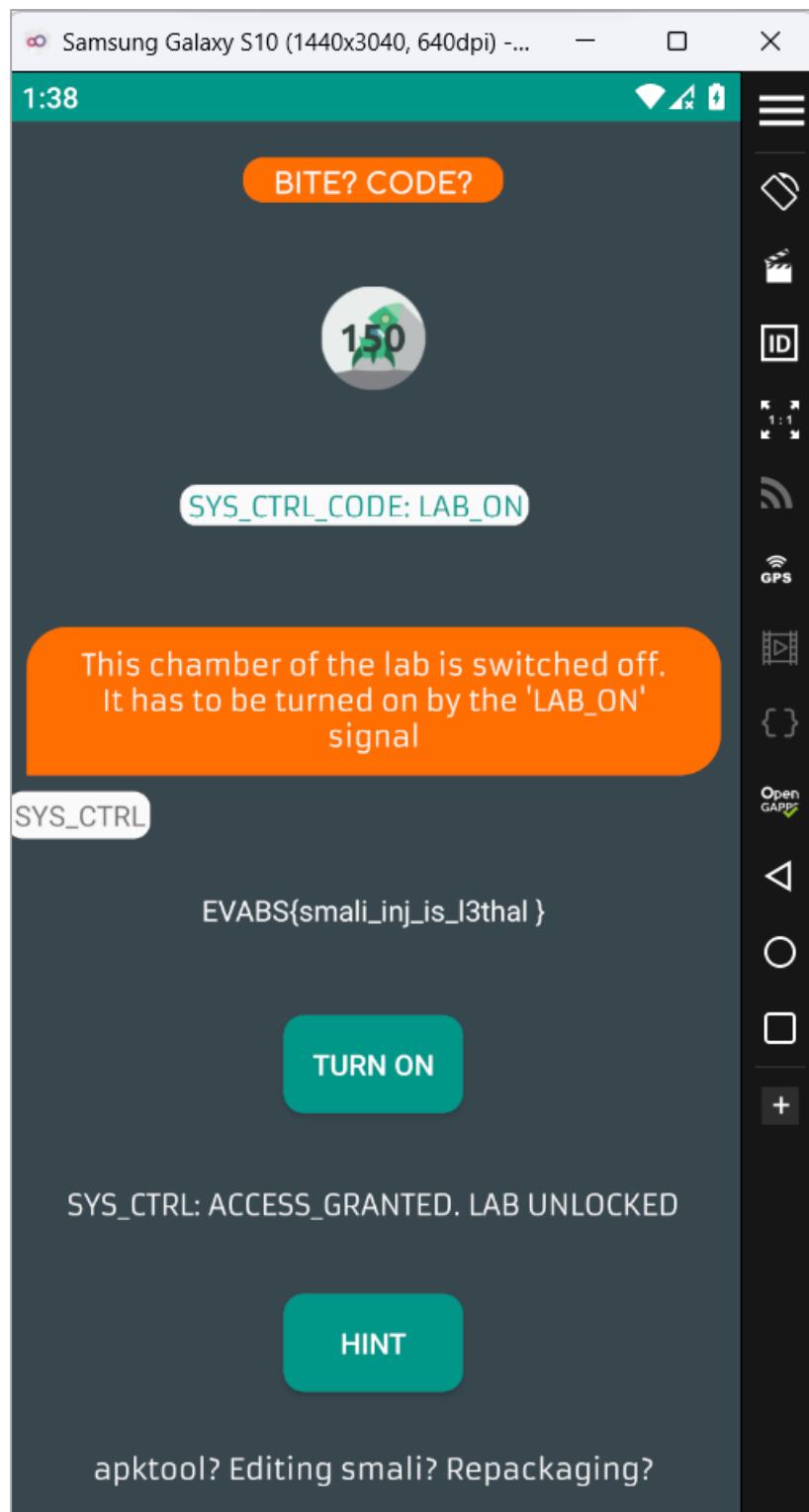
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/evabsv5]
$ apktool b EVABSV5 -o EVABSV6.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.7.0-dirty
I: Checking whether sources has changed ...
I: Checking whether resources has changed ...
I: Copying raw resources ...
I: Building apk file ...
I: Copying unknown files/dir ...
I: Built apk into: EVABSV6.apk
7.8 MB Android package Today
(kali㉿kali)-[~/Desktop/evabsv5]
$ apksigner sign --ks evabs.keystore EVABSV6.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:
Exception in thread "main" java.io.FileNotFoundException: EVABSV6.apk (No such file or directory)
    at java.base/java.io.RandomAccessFile.open0(Native Method)
    at java.base/java.io.RandomAccessFile.open(RandomAccessFile.java:344)
    at java.base/java.io.RandomAccessFile.<init>(RandomAccessFile.java:259)
    at java.base/java.io.RandomAccessFile.<init>(RandomAccessFile.java:213)
    at com.android.apksig.Apksigner.sign(ApkSigner.java:190)
    at com.android.apksigner.ApkSignerTool.sign(ApkSignerTool.java:356)
    at com.android.apksigner.ApkSignerTool.main(ApkSignerTool.java:85)

(kali㉿kali)-[~/Desktop/evabsv5]
$ apksigner sign --ks evabs.keystore EVABSV6.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:

```

Hình 25: Rebuild apk

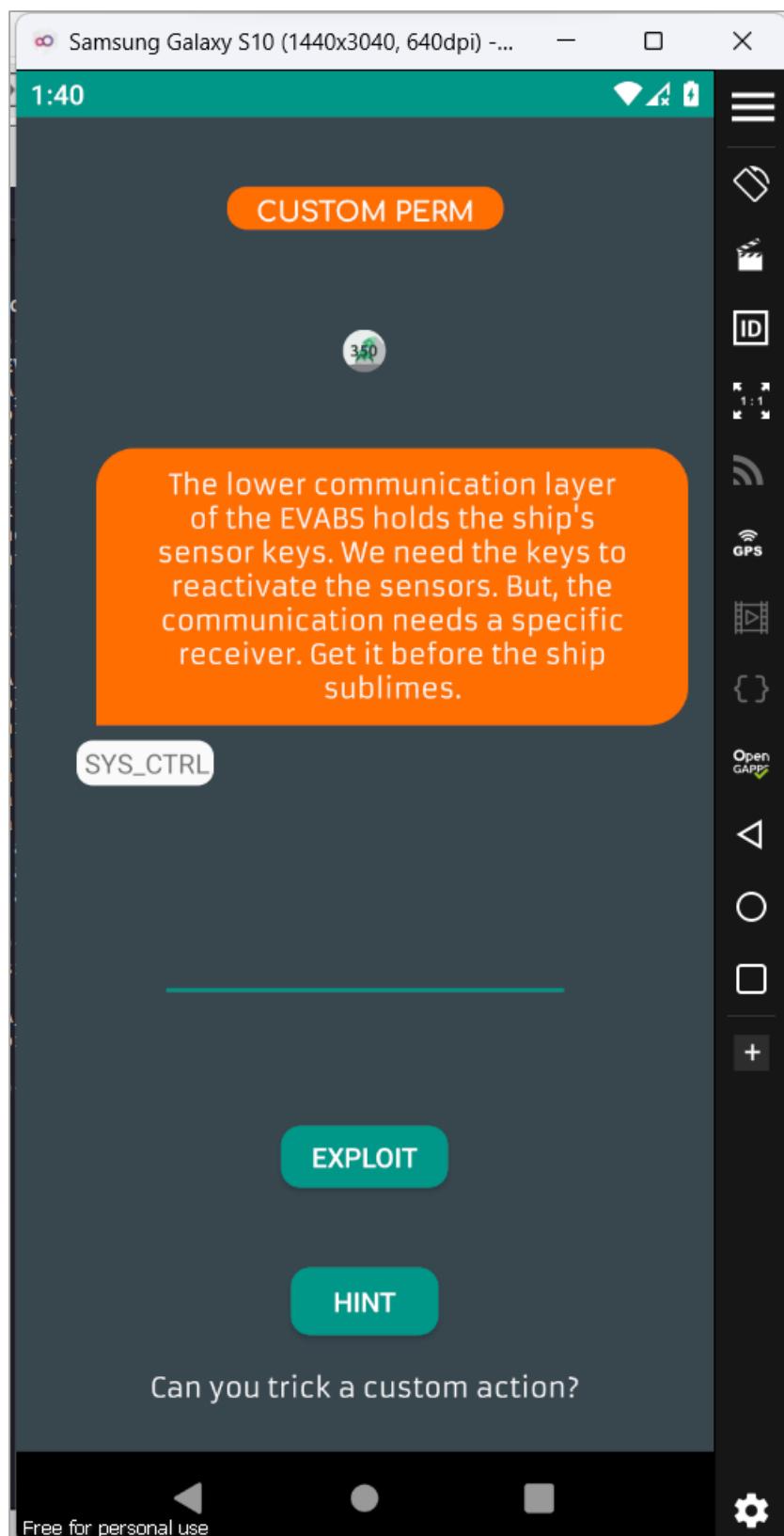
Cài đặt lại ứng dụng và bấm nút Turn on, chúng ta được:



Hình 26: Lấy được flag

**1.10.** Level 10 – Intercept

**1.11.** Level 11 – Custom permission



Hình 27: Level 11

Với level 11, chúng ta được yêu cầu nhập vào chuỗi hợp lệ để chuyển sang action khác, theo dõi nội dung của code, chúng ta thấy có điểm đáng chú ý:

```

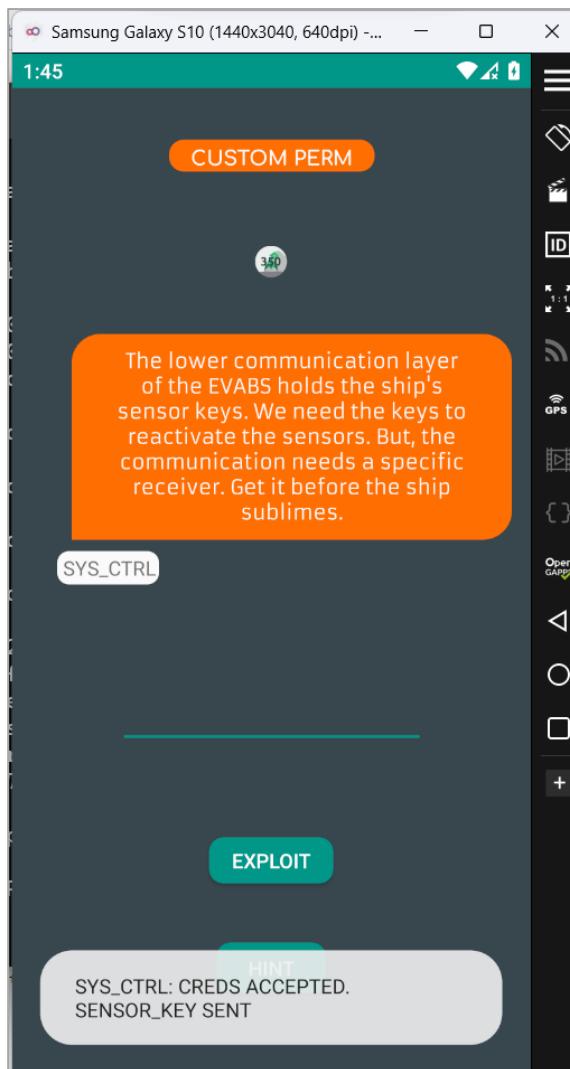
1 import android.view.View;
2 import android.widget.Button;
3 import android.widget.EditText;
4 import android.widget.TextView;
5 import android.widget.Toast;
6
7 public class CustomAccess extends AppCompatActivity {
8     public final String EVABS_SENSOR_KEY = "com.revo.evabs.actionSENSOR_KEY";
9
10    static {
11        System.loadLibrary("native-lib");
12    }
13
14    private void GetSensorKey() {
15        String str = ((EditText)findViewById(2131361891)).getText().toString();
16        if ((new String(new char[] {
17            'c', 'u', 's', 't', '0', 'm', '_', 'p', '3', 'r',
18            'm' }))).equals(str)) {
19            Toast.makeText((Context)this, "SYS_CTRL: CREDS ACCEPTED. SENSOR_KEY SENT", 1).show();
20            Intent intent = new Intent("com.revo.evabs.actionSENSOR_KEY");
21            String Builder stringBuilder = new String Builder();
22            stringBuilder.append("EVABS{");
23            stringBuilder.append(stringFromJNI());
24            stringBuilder.append("}");
25            intent.putExtra("android.intent.extra.TEXT", stringBuilder.toString());
26            intent.setType("text/plain");
27            startActivity(intent);
28        } else {
29            Toast.makeText((Context)this, "SYS_CTRL: WRONG_CREDS. SENSOR_KEY LOCKED", 1).show();
30        }
31    }
32
33    @Override
34    protected void onCreate(Bundle savedInstanceState) {
35        super.onCreate(savedInstanceState);
36        setContentView(R.layout.activity_main);
37        GetSensorKey();
38    }
39}

```

Hình 28: Phần code giúp hiện flag

Code trên đơn giản là nếu chúng ta nhập đúng chuỗi nó sẽ hiển thị ra flag và chuỗi đó chính là “cust0m\_p3rm”.

Tuy nhiên, khi nhập chuỗi trên, nó chuyển về chính trang này kèm thông điệp như dưới:



Hình 29: Hiển thị thông điệp

Vậy nên vấn đề là chúng ta cần tác động vào intent để flag hiện ra, code cụ thể như sau:

```
import frida
import sys

def onMessage(message, data):
    print(message)

package = "EVABS"

jscode = """
```

```
Java.perform(function () {  
    send("[-] Starting hooks android.content.Intent.putExtra");  
  
    var intent = Java.use("android.content.Intent");  
  
    intent.putExtra.overload("java.lang.String", "java.lang.String").implementation = function(var_1,  
var_2) {  
  
        send("[+] Flag: " + var_2);  
  
    };  
  
});  
****  
  
process = frida.get_usb_device().attach(package)  
  
script = process.create_script(jscode)  
  
script.on("message", onMessage)  
  
print("[*] Hooking", package)  
  
script.load()  
  
sys.stdin.read()
```

Dòng package chúng ta sẽ xác định bằng lệnh frida-ps -U và kết quả sẽ cho tên cùng với pid:

```
PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

PS C:\Users\Pengu> frida-ps -U
PID  Name
-----
4830  EVABS
1929  Google Play Store
578   adb
1129  android.ext.services
204   android.hardware.atace@1.0-service
806   android.hardware.audio.service
294   android.hardware.authsecret@1.0-service
296   android.hardware.camera.provider@2.4-service
297   android.hardware.cas@1.2-service
298   android.hardware.configstore@1.1-service
301   android.hardware.drm@1.0-service
302   android.hardware.drm@1.3-service.clearkey
304   android.hardware.gatekeeper@1.0-service.software
305   android.hardware.gnss@1.0-service
306   android.hardware.graphics.allocator@2.0-service
307   android.hardware.graphics.composer@2.3-service
308   android.hardware.health@2.0-service.genymotion
326   android.hardware.identity-service.example
761   android.hardware.input.classifier@1.0-service.default
206   android.hardware.keymaster@3.0-service
309   android.hardware.light@2.0-service
310   android.hardware.memtrack@1.0-service
311   android.hardware.neuralnetworks@1.3-service-sample-all
313   android.hardware.neuralnetworks@1.3-service-sample-float-fast
314   android.hardware.neuralnetworks@1.3-service-sample-float-slow
316   android.hardware.neuralnetworks@1.3-service-sample-minimal
```

Hình 30: Package sẽ là "EVABS"

Thực hiện hooking với code trên và thực hiện lại bước nhập chuỗi “cust0m\_p3rm” để lấy flag:

```

1 import frida
2 import sys
3
4 def onMessage(message, data):
5     print(message)
6
7 package = "EVABS"
8
9 jscode = """
10 Java.perform(function () {
11     send("[+] Starting hooks android.content.Intent.putExtra");
12     var intent = java.use("android.content.Intent");
13     ... intent.putExtra.overload("java.lang.String", "java.lang.String").implementation = function(var_1, var_2) {
14         ... send("[+] Flag: " + var_2);
15     };
16 });
17 """
18
19 process = frida.get_usb_device().attach(package)
20 script = process.create_script(jscode)
21 script.on("message", onMessage)
22 print("[+] Hooking", package)
23 script.load()
24 sys.stdin.read()

```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

[Running] python -u "C:\Users\Pengu\AppData\Local\Temp\tempCodeRunnerFile.py"

[+] Hooking EVABS

{'type': 'send', 'payload': '[+] Starting hooks android.content.Intent.putExtra'}

{'type': 'send', 'payload': '[+] Flag: EVABS{always\_verify\_packag3s}'}

{'type': 'error', 'description': 'Error: Implementation for putExtra expected return value compatible with android.content.Intent', 'stack': 'Error: Implementation for putExtra expected return value compatible with android.content.Intent\n at ne (frida/node\_modules/frida-java-bridge/lib/class-factory.js:674)\n at <anonymous> (frida/node\_modules/frida-java-bridge/lib/class-factory.js)', 'fileName': 'frida/node\_modules/frida-java-bridge/lib/class-factory.js', 'lineNumber': 674, 'columnNumber': 1}'}

Ln 12, Col 53 (159 selected) Spaces: 4 UFT-8 CRLF ⚡ Python Go Live ⚡ Prettier ⚡ 3:46 PM 6/19/2023

Hình 31: Lấy được flag

## EVABS{always\_ver1fy\_packag3s}

### 1.12. Level 12 - Intrusment

```

com/revo/evabs/Fridal.class x
JD-GUI Decomplier
Match case

7 import android.widget.Button;
8 import android.widget.TextView;
9 import java.util.Random;
10
11 public class Fridal extends AppCompatActivity implements View.OnClickListener {
12     int a = 25;
13
14     int b = 2;
15
16     int x;
17
18     static {
19         System.loadLibrary("native-lib");
20     }
21
22     public void onClick(View paramView) {
23         TextView textView3 = (TextView)findViewById(2131361996);
24         TextView textView4 = (TextView)findViewById(2131362132);
25         TextView textView2 = (TextView)findViewById(2131362134);
26         TextView textView1 = (TextView)findViewById(2131362142);
27         textView4.setText(String.valueOf(this.a));
28         textView2.setText(String.valueOf(this.b));
29         this.x = this.a * this.b;
30         int i = (new Random()).nextInt(70);
31         textView1.setText(String.valueOf(this.x));
32         if (this.x > i + 150) {
33             textView3.setText("VIBRAN IS READY TO FLY! YOU ARE GOING HOME!");
34             Log.d("CONGRATZ!", stringFromJNI());
35         } else {
36             textView3.setText("Co-ordinates Not Found!");
37         }
38
39     protected void onCreate(Bundle paramBundle) {
40         super.onCreate(paramBundle);
41         setContentView(2131492901);
42         ((Button)findViewById(2131361902)).setOnClickListener(this);
43         ((Button)findViewById(2131361844)).setOnClickListener((View.OnClickListener)new Object(this, (TextView)findViewById(2131362093)));
44     }
45
46     public native String stringFromJNI();
47 }

```

Hình 32: Nội dung code level 12

Chúng ta có thể thấy nội dung code sẽ có 2 biến a và b với kết quả được gán từ trước và x sẽ là kết quả của a nhân b, sau đó giá trị biến i sẽ được random với Random().nextInt(70) – giá trị I sẽ nằm trong khoảng 0 đến 69.

Sau đó, kết quả sẽ được đưa vào lệnh if để so sánh x có lớn hơn i+150 hay không hay cũng chính so sánh a\*b với i+150. Nếu lớn hơn, thông điện sẽ được hiển thị trong log.

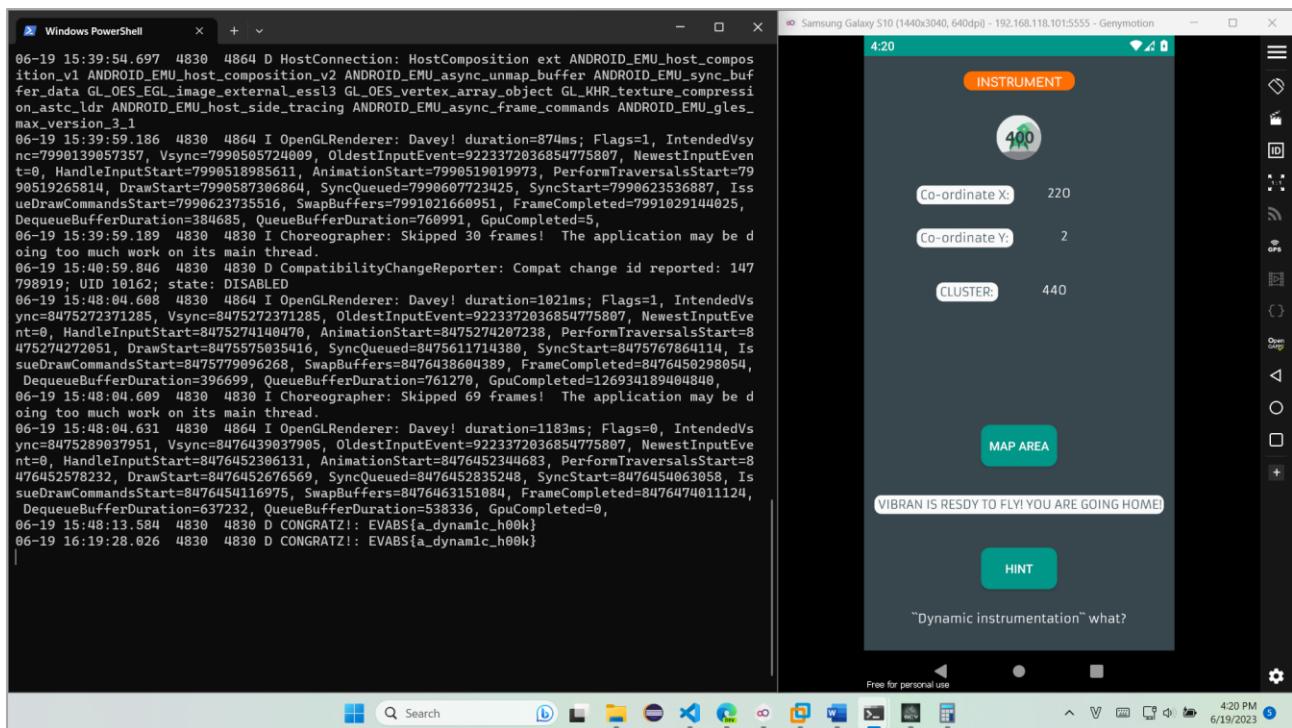
Do a và b trước đó chỉ gán là 25 và 2 nên kết quả chỉ là 50 không thể lớn hơn i+150 được, vậy nên chúng ta sẽ thay đổi giá trị của a hoặc b sao cho kết quả luôn lớn hơn khi kết quả i được random + 150.

```
30 .method public constructor <init>()V
31     .locals 1
32
33     .line 13
34     invoke-direct {p0}, Landroid/support/v7/app/AppCompatActivity;→<init>()V
35
36     .line 15
37     const/16 v0, 0xDC
38
39     istruct v0, p0, Lcom/revo/evabs/Frida1;→a:I
40
41     const/4 v0, 0x2
42
43     istruct v0, p0, Lcom/revo/evabs/Frida1;→b:I
44
45     return-void
46 .end method
```

Hình 33: Sửa giá trị a

Trong hình, chúng ta thay đổi giá trị a thành 0xDC tương ứng với 220 trong thập phân để giá trị sau khi nhân với b chắc chắn sẽ lớn hơn i+150 (max sẽ là 69+150 = 219).

Đóng gói và kí, chúng ta được:

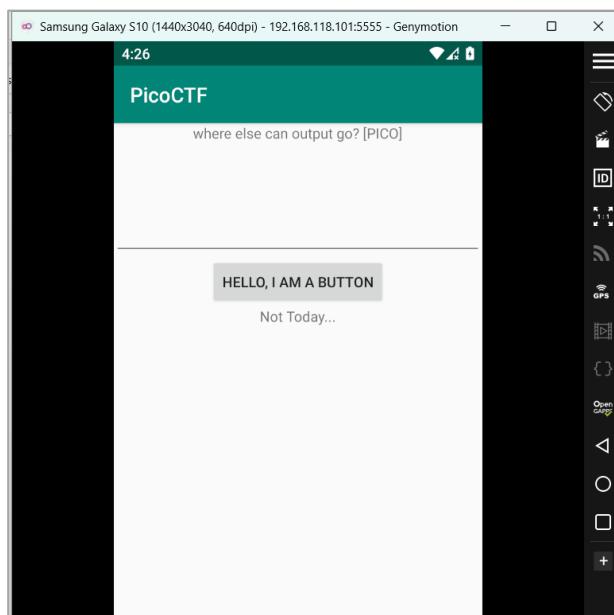


Hình 34: Tìm thấy flag

## EVABS{a\_dynam1c\_h00k}

### 2. Challenges 2

#### 2.1. One.apk



Hình 35: Level 1

Xem log của ứng dụng:

The screenshot shows a Windows PowerShell window on the left and a Genymotion emulator window on the right. The PowerShell window displays log output from an adb shell command, specifically searching for 'picoctf' processes and then using logcat to capture logs. The Genymotion emulator shows the PicoCTF application running on an Android device. The app has a green header bar with the title 'PicoCTF'. Below the header, there is a text input field containing the placeholder 'where else can output go? [PICO]'. A button labeled 'HELLO, I AM A BUTTON' is visible, along with a note 'Not Today...'. The Genymotion interface includes various icons for file operations and device management.

```

PS C:\Program Files\Genymobile\Genymotion\tools> .\adb shell ps | grep 'picoctf'
u0_a177 5432 283 12763220 126568 ep_poll 73722efc90a S com.hellocmu.picocf
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb logcat --pid 5432
----- beginning of main -----
06-19 16:24:21.030 5432 5432 I Zygote : seccomp disabled by setenforce 0
06-19 16:24:21.040 5432 5432 I ellocmu.picocf: Late-enabling -Xcheck:jni
06-19 16:24:21.083 5432 5432 I ellocmu.picocf: Unquickening 12 vdex files!
06-19 16:24:21.140 5432 5432 W ellocmu.picocf: Unexpected CPU variant for X86 using default ts: x86_64
06-19 16:24:21.452 5432 5432 D ApplicationLoaders: Returning zygote-cached class loader: /system/framework/android.test.base.jar
06-19 16:24:21.547 5432 5432 D NetworkSecurityConfig: No Network Security Config specified, using platform default
06-19 16:24:21.550 5432 5432 D NetworkSecurityConfig: No Network Security Config specified, using platform default
06-19 16:24:21.555 5432 5432 I RenderThread: type=1400 audit(0.0:543): avc: denied { read } for name="libandroiddemu.so" dev="sdal4" ino=4484 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:21.655 5432 5432 I RenderThread: type=1400 audit(0.0:544): avc: denied { open } for path="/system/vendor/lib64/libandroiddemu.so" dev="sdal4" ino=4484 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:21.655 5432 5432 I RenderThread: type=1400 audit(0.0:545): avc: denied { getat tr } for path="/system/vendor/lib64/libandroiddemu.so" dev="sdal4" ino=4484 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:21.659 5432 5432 I RenderThread: type=1400 audit(0.0:546): avc: denied { execute } for path="/system/vendor/lib64/libandroiddemu.so" dev="sdal4" ino=4484 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:21.664 5432 5463 D libEGL : loaded /vendor/lib64/egl/LibEGL_emulation.so
06-19 16:24:21.666 5432 5463 D libEGL : loaded /vendor/lib64/egl/libGLESv1_CM_emulation.so
06-19 16:24:21.672 5432 5463 D libEGL : loaded /vendor/lib64/egl/libGLESv2_emulation.so
06-19 16:24:21.803 5432 5432 W ellocmu.picocf: Accessing hidden method Landroid/view/View; >computeFitSystemWindows(Landroid/graphics/Rect;Landroid/graphics/Rect;)Z (greylist, reflection, allowed)
06-19 16:24:21.803 5432 5432 W ellocmu.picocf: Accessing hidden method Landroid/view/ViewG

```

Hình 36: Xem log của app

Sau đó chúng ta bấm nút HELLO, I AM A BUTTON, flag sẽ được hiển thị:

The screenshot shows a Windows PowerShell window on the left and a Genymotion emulator window on the right. The PowerShell window displays log output from an adb shell command, specifically searching for 'picocf' processes and then using logcat to capture logs. The Genymotion emulator shows the PicoCTF application running on an Android device. The app has a green header bar with the title 'PicoCTF'. Below the header, there is a text input field containing the placeholder 'where else can output go? [PICO]'. A button labeled 'HELLO, I AM A BUTTON' is visible, along with a note 'Not Today...'. The Genymotion interface includes various icons for file operations and device management.

```

tr } for path="/system/vendor/lib64/hw/gralloc.ranchu.so" dev="sdal4" ino=4471 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:23.295 5432 5432 I RenderThread: type=1400 audit(0.0:548): avc: denied { read } for name="gralloc.ranchu.so" dev="sdal4" ino=4471 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:23.295 5432 5432 I RenderThread: type=1400 audit(0.0:549): avc: denied { open } for path="/system/vendor/lib64/hw/gralloc.ranchu.so" dev="sdal4" ino=4471 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:23.295 5432 5432 I RenderThread: type=1400 audit(0.0:550): avc: denied { execute } for path="/system/vendor/lib64/hw/gralloc.ranchu.so" dev="sdal4" ino=4471 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:23.295 5432 5432 I RenderThread: type=1400 audit(0.0:551): avc: denied { open } for path="/system/vendor/lib64/hw/gralloc.ranchu.so" dev="sdal4" ino=4471 scontext=u:r:untrusted_app_29:s0:c177,c256,c512,c768 tcontext=u:object_r:vendor_file:s0 tclass=file permissive=1 app=com.hellocmu.picocf
06-19 16:24:23.295 5432 5461 D HostConnection: createUnique: call
06-19 16:24:23.302 5432 5461 D HostConnection: HostConnection::get() New Host Connection established 0x73702928be50, pid 5432, tid 5461
06-19 16:24:23.310 5432 5461 D HostConnection: HostComposition ext ANDROID_EMU_host_composition_v1 ANDROID_EMU_host_composition_v2 ANDROID_EMU_async_unmap_buffer ANDROID_EMU_sync_buffer_data GL_OES_EGL_image_external_ess3 GL_OES_vertex_array_object GL_KHR_texture_compression_astc_ldr ANDROID_EMU_host_side_tracing ANDROID_EMU_async_frame_commands ANDROID_EMU_gles_max_version_3_1
06-19 16:24:23.772 5432 5461 I OpenGLRenderer: Davey! duration=1753ms; Flags=1, IntendedVsync=10653788950811, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=10653802512123, AnimationStart=10653802567158, PerformTraversalsStart=10653802624987, DrawStart=10655158240016, SyncQueued=10655163140080, SyncStart=10655234793321, IssueDrawCommandsStart=10655234970997, SwapBuffers=10655562326289, FrameCompleted=1065514389610, DequeueBufferDuration=1653003, QueueBufferDuration=710426, GpuCompleted=0, 06-19 16:24:23.786 5432 5432 I Choreographer: Skipped 108 frames! The application may be doing too much work on its main thread.
06-19 16:24:23.822 5432 5461 I OpenGLRenderer: Davey! duration=1839ms; Flags=0, IntendedVsSync=10653822284143, Sync=10655622840871, OldestInputEvent=9223372036854775807, NewestInputEvent=0, HandleInputStart=1065538025121231, AnimationStart=10655632691657, SyncStart=1065563535848, IssueDrawCommandsStart=10655635446197, SwapBuffers=10655637481651, FrameCompleted=1065564372220, DequeueBufferDuration=3454349, QueueBufferDuration=8532649, GpuCompleted=0, 06-19 16:24:28.029 5432 5432 I PICO : picoCTF{a.moose.once.bit.my.sister}

```

Hình 37: Lấy được flag

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

## 2.2. two.apk

Thực hiện mở tệp two.apk bằng BCV, chúng ta thấy code nằm trong phần MainActivity dùng để lấy flag như sau:



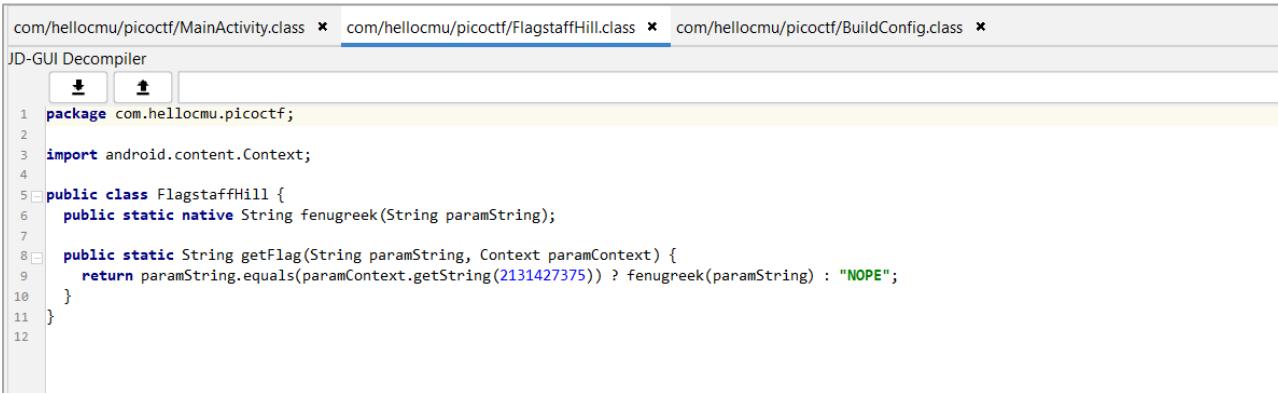
```

com/hellocmu/picoctf/MainActivity.class x com/hellocmu/picoctf/FlagstaffHill.class x com/hellocmu/picoctf/BuildConfig.class x
JD-GUI Decompiler
1 package com.hellocmu.picoctf;
2
3 import android.content.Context;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9 import androidx.appcompat.app.AppCompatActivity;
10
11 public class MainActivity extends AppCompatActivity {
12     Button button;
13
14     Context ctx;
15
16     TextView text_bottom;
17
18     EditText text_input;
19
20     TextView text_top;
21
22     public void buttonClick(View paramView) {
23         String str = this.text_input.getText().toString();
24         this.text_bottom.setText(FlagstaffHill.getFlag(str, this.ctx));
25     }

```

Hình 38: Gọi tới getFlag

Hàm getFlag có nội dung như dưới:



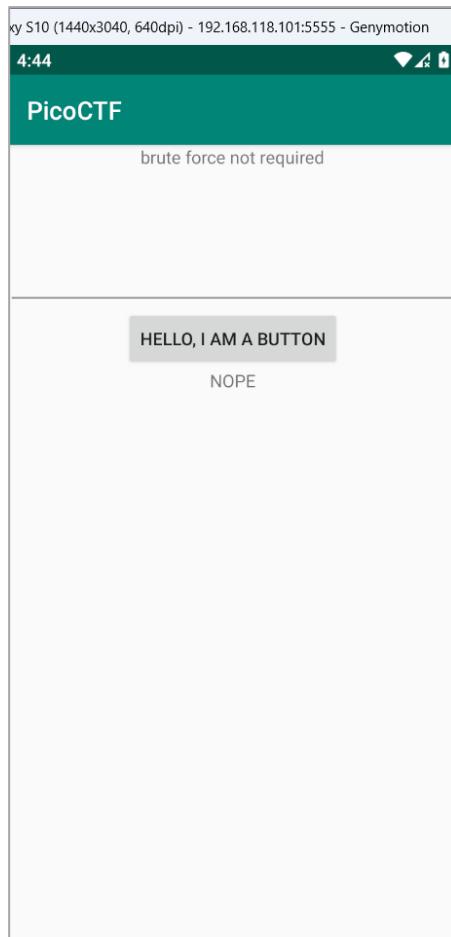
```

com/hellocmu/picoctf/MainActivity.class x com/hellocmu/picoctf/FlagstaffHill.class x com/hellocmu/picoctf/BuildConfig.class x
JD-GUI Decompiler
1 package com.hellocmu.picoctf;
2
3 import android.content.Context;
4
5 public class FlagstaffHill {
6     public static native String fenugreek(String paramString);
7
8     public static String getFlag(String paramString, Context paramContext) {
9         return paramString.equals(paramContext.getString(2131427375)) ? fenugreek(paramString) : "NOPE";
10    }
11 }
12

```

Hình 39: Nội dung hàm getFlag

Theo đó, hàm getFlag sẽ trả về kết quả dựa trên chuỗi nhập vào, nếu chuỗi nhập vào trùng với chuỗi giá trị chuỗi được gán cho “2132427375”, thì flag sẽ được hiển thị, ngược lại nó sẽ hiển thị NOPE:



Hình 40: Chuỗi nhập vào không đúng

Dữ liệu liên quan tới các chuỗi sẽ được tìm thấy trong lớp R\$string, chúng ta thấy nó sẽ là **password**:

```

Workspace
com/hellocmu/picoctf/BuildConfig.class x com/hellocmu/picoctf/R$id.class x com/hellocmu/picoctf/FlagstaffHill x com/hellocmu/picoctf/R$string.class x
com/hellocmu/picoctf>MainActivity.class x com/hellocmu/picoctf/FlagstaffHill.class x

JD-GUI Decomiler
2131427375 Match case
58 public static final int abc_menu_space_shortcut_label = 2131427355;
59
60 public static final int abc_menu_sym_shortcut_label = 2131427356;
61
62 public static final int abc_prepend_shortcut_label = 2131427357;
63
64 public static final int abc_search_hint = 2131427358;
65
66 public static final int abc_searchview_description_clear = 2131427359;
67
68 public static final int abc_searchview_description_query = 2131427360;
69
70 public static final int abc_searchview_description_search = 2131427361;
71
72 public static final int abc_searchview_description_submit = 2131427362;
73
74 public static final int abc_searchview_description_voice = 2131427363;
75
76 public static final int abc_shareactionprovider_share_with = 2131427364;
77
78 public static final int abc_shareactionprovider_share_with_application = 2131427365;
79
80 public static final int abc_toolbar_collapse_description = 2131427366;
81
82 public static final int app_name = 2131427367;
83
84 public static final int bat = 2131427368;
85
86 public static final int bear = 2131427369;
87
88 public static final int cottontail = 2131427370;
89
90 public static final int gopher = 2131427371;
91
92 public static final int hint = 2131427372;
93
94 public static final int manatee = 2131427373;
95
96 public static final int myotis = 2131427374;
97
98 public static final int password = 2131427375;

```

Hình 41: Password được gán với 2131427375

Tiếp đó, chúng ta sẽ tìm giá trị chuỗi được gán cho password trong tệp res/values/string.xml:

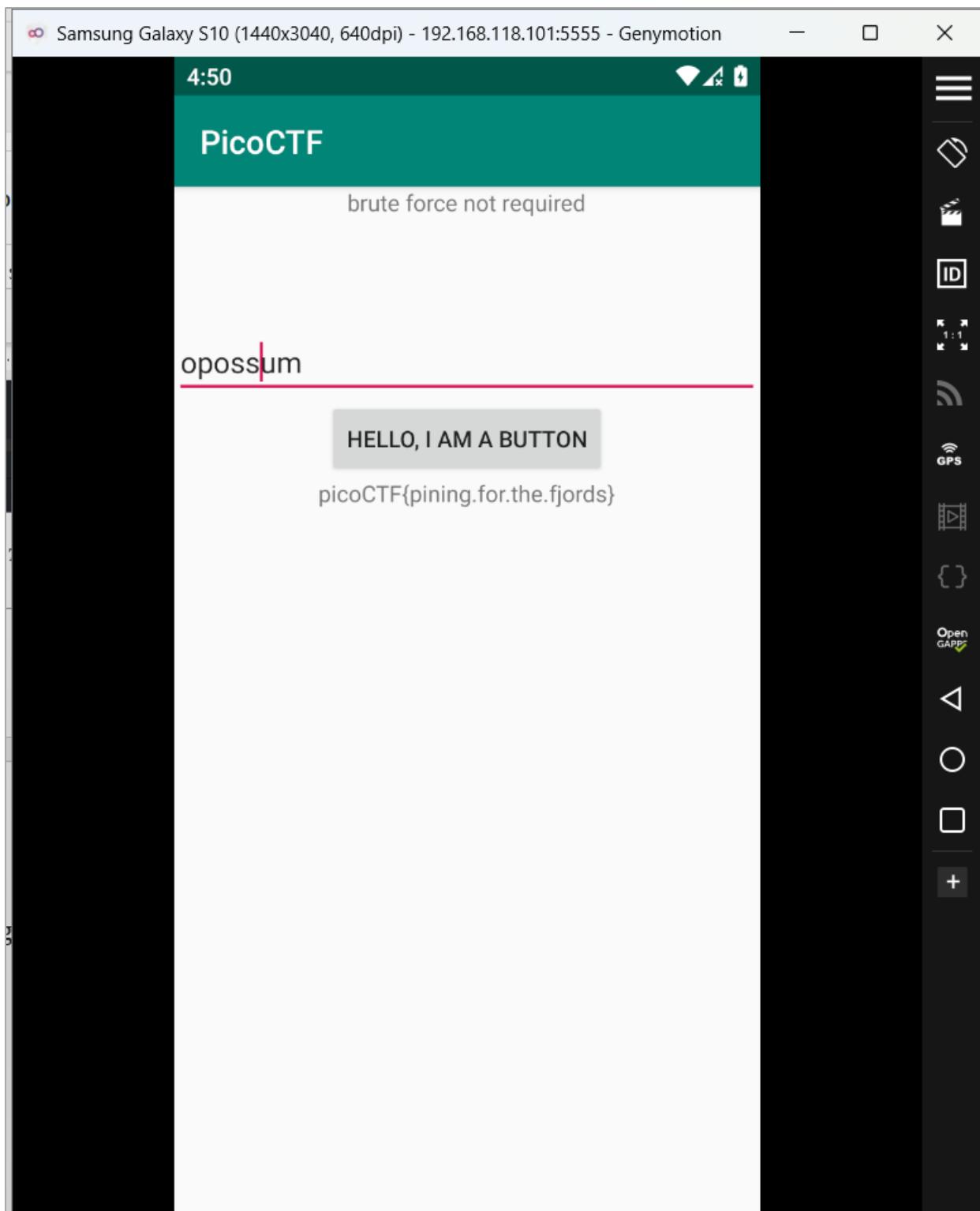
```

File Edit Search View Document Help
strings.xml
strings.xml
17 <string name="abc_font_family_display_3_material">sans-serif</string>
18 <string name="abc_font_family_display_4_material">sans-serif-light</string>
19 <string name="abc_font_family_headline_material">sans-serif</string>
20 <string name="abc_font_family_menu_material">sans-serif</string>
21 <string name="abc_font_family_subhead_material">sans-serif</string>
22 <string name="abc_font_family_title_material">sans-serif-medium</string>
23 <string name="abc_menu_alt_shortcut_label">Alt+</string>
24 <string name="abc_menu_ctrl_shortcut_label">Ctrl+</string>
25 <string name="abc_menu_delete_shortcut_label">Delete</string>
26 <string name="abc_menu_enter_shortcut_label">Enter</string>
27 <string name="abc_menu_function_shortcut_label">Function</string>
28 <string name="abc_menu_meta_shortcut_label">Meta+</string>
29 <string name="abc_menu_shift_shortcut_label">Shift+</string>
30 <string name="abc_menu_space_shortcut_label">Space</string>
31 <string name="abc_menu_sym_shortcut_label">Sym</string>
32 <string name="abc_prepend_shortcut_label">Menu</string>
33 <string name="abc_search_hint">Search...</string>
34 <string name="abc_searchview_description_clear">Clear query</string>
35 <string name="abc_searchview_description_query">Search query</string>
36 <string name="abc_searchview_description_search">Search</string>
37 <string name="abc_searchview_description_submit">Submit query</string>
38 <string name="abc_searchview_description_voice">Voice search</string>
39 <string name="abc_shareactionprovider_share_with">Share with</string>
40 <string name="abc_shareactionprovider_share_with_application">Share with %s</string>
41 <string name="abc_toolbar_collapse_description">Collapse</string>
42 <string name="app_name">PicoCTF</string>
43 <string name="bat">mink</string>
44 <string name="bear">margay</string>
45 <string name="cottontail">shrew</string>
46 <string name="gopher">armadillo</string>
47 <string name="hint">brute force not required</string>
48 <string name="manatee">caribou</string>
49 <string name="myotis">jackrabbit</string>
50 <string name="password">opossum</string>
51 <string name="porcupine">blackbuck</string>
52 <string name="porpoise">mouflon</string>

```

Hình 42: Tìm thấy chuỗi cần nhập vào

Chuỗi chúng ta sẽ nhập vào ứng dụng là opossum:



Hình 43: Tìm thấy flag

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

### 2.3. three.apk

Nội dung code getFlag trong level 3:



```

com/hellocmu/picoctf/FlagstaffHill x com/hellocmu/picoctf/R$string.class x com/hellocmu/picoctf/MainActivity.class x com/hellocmu/picoctf/FlagstaffHill.class x
com/hellocmu/picoctf/MainActivity.class x com/hellocmu/picoctf/FlagstaffHill.class x com/hellocmu/picoctf/BuildConfig.class x com/hellocmu/picoctf/R$id.class x
JD-GUI Decompiler
Match case
1 package com.hellocmu.picoctf;
2
3 import android.content.Context;
4
5 public class FlagstaffHill {
6     public static String getFlag(String paramString, Context paramContext) {
7         String[] arrayOfString = new String[6];
8         arrayOfString[0] = "weatherwax";
9         arrayOfString[1] = "ogg";
10        arrayOfString[2] = "garlick";
11        arrayOfString[3] = "nitt";
12        arrayOfString[4] = "aching";
13        arrayOfString[5] = "dismas";
14        int j = 3 - 3;
15        int m = 3 / 3 + j;
16        int i = m + m - j;
17        int k = 3 + i;
18        return paramString.equals("".concat(arrayOfString[k]).concat(".").concat(arrayOfString[m]).concat(".").concat(arrayOfString[j]).concat(".").concat(arrayOfString[k + 1]));
19    }
20
21    public static native String sesame(String paramString);
22}

```

Hình 44: Hàm getFlag

Chúng ta cần thực hiện tính toán các giá trị j, m, I, k để từ đó đưa vào kết quả lấy các chuỗi với vị trí tương ứng:

```

return paramString.equals("".concat(arrayOfString[k]).concat(".").concat(arrayOfString[m]).concat(".").
concat(arrayOfString[j]).concat(".").concat(arrayOfString[k + j - m]).concat(".").concat(arrayOfString[3]).concat(".").concat(arrayOfString[i])) ? sesame(paramString) : "NOPE";

```

Qua tính toán, chúng ta được như sau:

$$j = 0$$

$$m = 1$$

$$i = 1 + 1 - 0 = 2$$

$$k = 3 + 2 = 5$$

$$k+j-m = 5 + 0 - 1 = 4$$

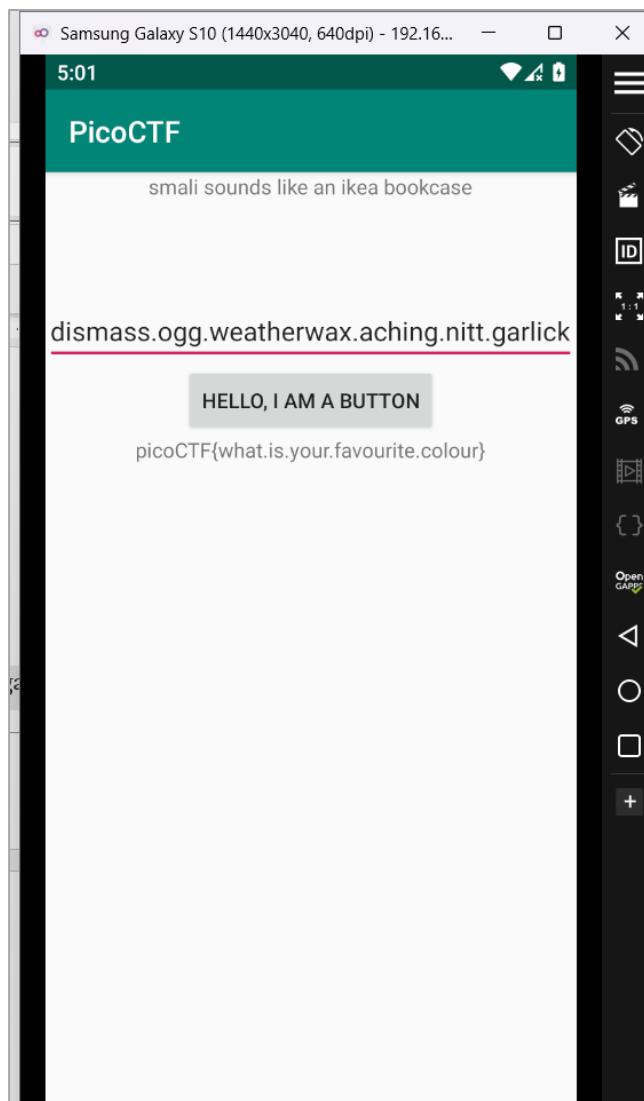
Thứ tự các giá trị được gọi tới trong dòng code trên sẽ là

#### 5.1.0.4.3.2

Tương ứng với:

dismass.ogg.weatherwax.aching.nitt.garlick

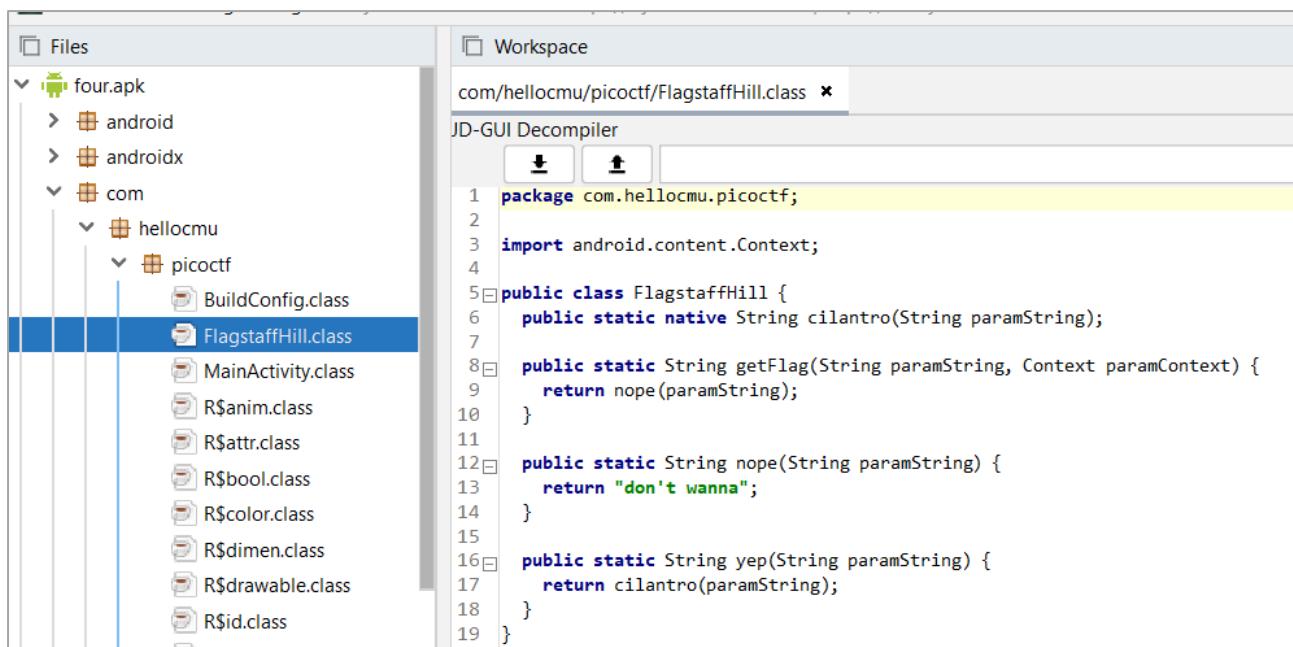
Nhập chuỗi trên vào ứng dụng, chúng ta có flag:



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

#### 2.4. four.apk

Nội dung hàm getFlag của level 4:



Hình 45: Nội dung hàm getFlag

Ta thấy hàm getFlag luôn được return về hàm nope(), vậy nên chúng ta sẽ thử thay đổi để nó gọi tới hàm yep:

```

19 .method public static getFlag(Ljava/lang/String;Landroid/content/Context;)Ljava/lang/String;
20     .locals 1
21     .param p0, "input"    # Ljava/lang/String;
22     .param p1, "ctx"      # Landroid/content/Context;
23
24     .line 19
25     invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;→nope(Ljava/lang/String;)Ljava/lang/String;
26
27     move-result-object v0
28
29     .line 20
30     .local v0, "flag":Ljava/lang/String;
31     return-object v0
32 .end method

```

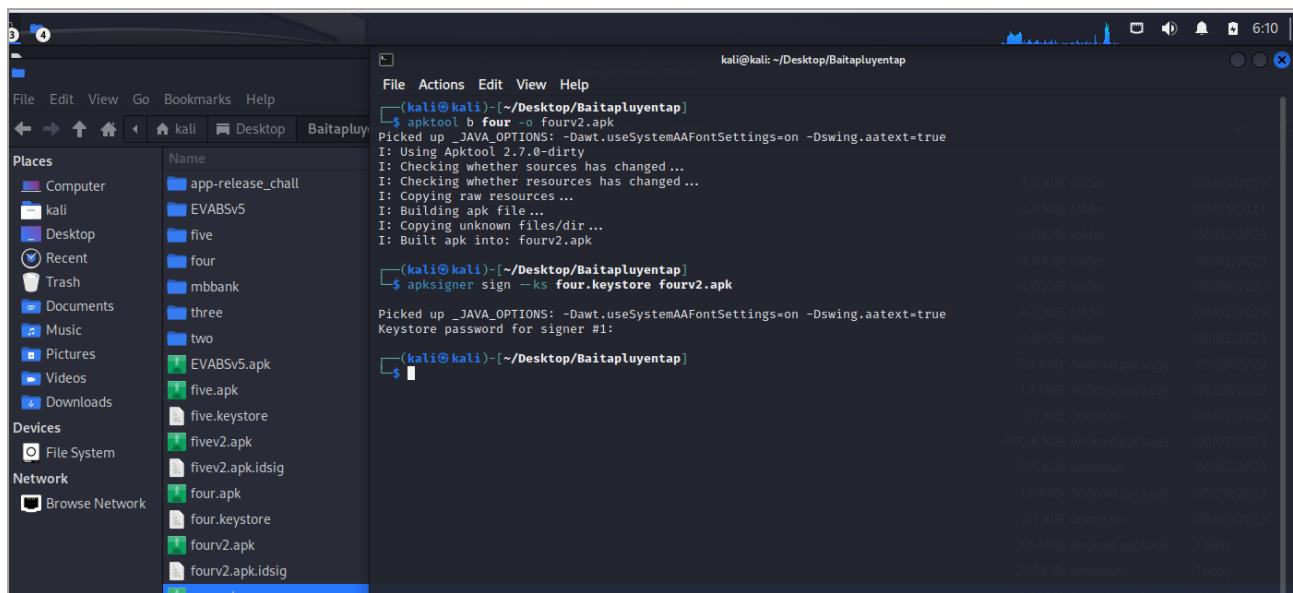
```

19 .method public static getFlag(Ljava/lang/String;Landroid/content/Context;)Ljava/lang/String;
20     .locals 1
21     .param p0, "input"    # Ljava/lang/String;
22     .param p1, "ctx"      # Landroid/content/Context;
23
24     .line 19
25     invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;→yep(Ljava/lang/String;)Ljava/lang/String;
26
27     move-result-object v0
28
29     .line 20
30     .local v0, "flag":Ljava/lang/String;
31     return-object v0
32 .end method

```

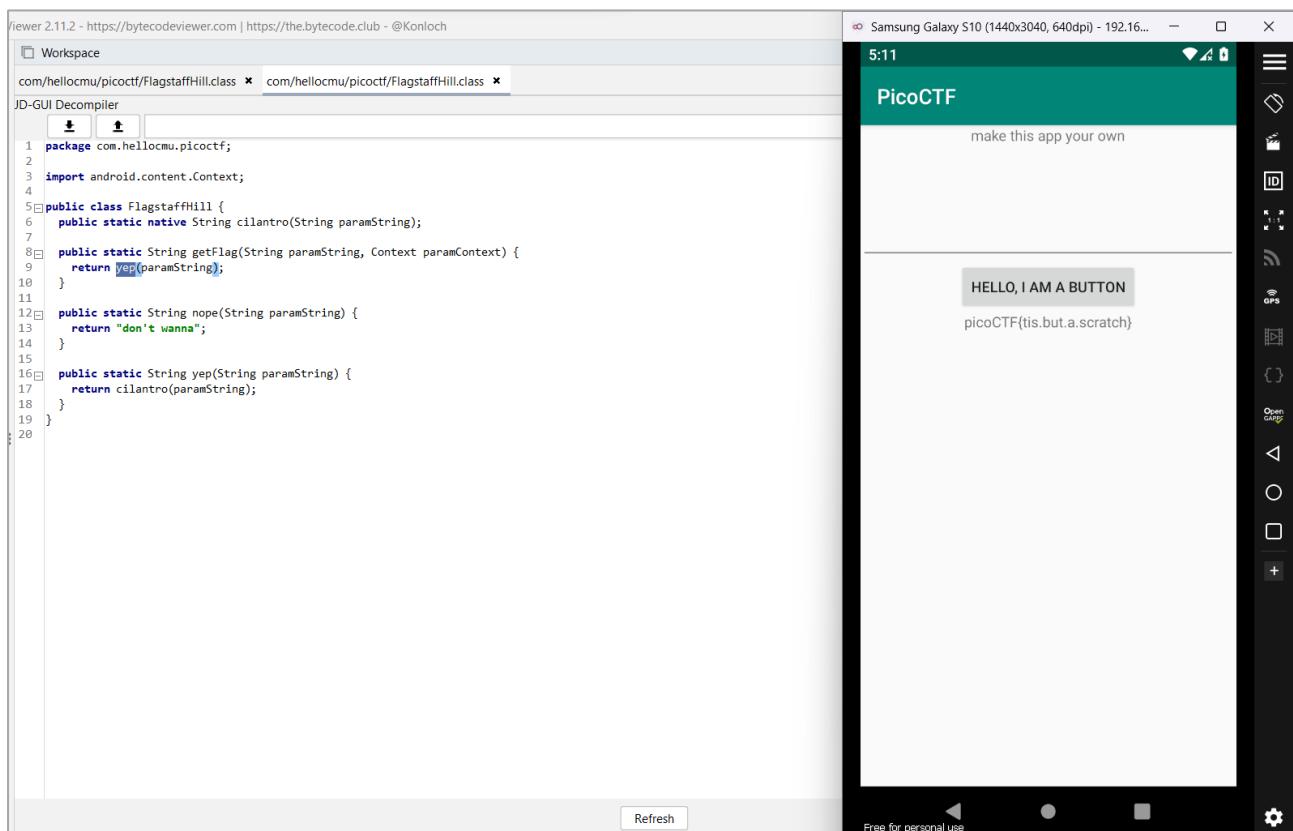
Hình 46: Thay đổi code trong hàm getFlag

Sửa dòng 25 từ “nope” thành “yep”, sau đó rebuild và kí:



Hình 47: Repatch apk

Kết quả chúng ta được:



Hình 48: Lấy được flag

picoCTF{tis.but.a.scratch}

## 2.5. five.apk

```

File View Settings Plugins Bytecode Viewer 2.11.2 - https://bytecodeviewer.com | https://the.bytecode.club - @Konloch
Files
> five.apk
> fourv2.apk
> fourv.apk
> fourv2.apk
> one.apk
> three.apk
> two.apk
Workspace
JD-GUI Decomplier
1 package com.hellocmu/picoctf;
2 import android.content.Context;
3
4 public class FlagstaffHill {
5     public static native String cardamom(String paramString);
6
7     public static String getFlag(String paramString, Context paramContext) {
8         StringBuilder paramStringBuilder = new StringBuilder("aaa");
9         paramStringBuilder.append(paramString);
10        paramStringBuilder.append("bbb");
11        paramStringBuilder.append("ccc");
12        paramStringBuilder.append("ddd");
13        paramStringBuilder.append("eee");
14        paramStringBuilder.append("fff");
15        paramStringBuilder.append("ggg");
16        paramStringBuilder.append("hhh");
17        paramStringBuilder.append("iii");
18        paramStringBuilder.append("jjj");
19        paramStringBuilder.append("kkk");
20        paramStringBuilder.append("lll");
21        paramStringBuilder.append("mmm");
22        paramStringBuilder.append("nnn");
23        paramStringBuilder.append("ooo");
24        paramStringBuilder.append("ppp");
25        return paramString.equals("") .concat(stringBuilder1.toString()) .concat(stringBuilder2.toString()) .concat(stringBuilder3.toString()) ? "call it" : "NOPE";
26    }
}

```

Hình 49: Nội dung hàm getFlag

Hàm getFlag của câu 5 cũng tương tự câu 3 là chúng ta cần tính toán, nhưng câu 5 khi tìm được đúng chuỗi nó chỉ trả về thông điệp là “call it”, vậy nên chúng ta cần thực hiện sửa code sao cho nó gọi tới hàm cardamom():

```

228     invoke-virtual {p0, v4}, Ljava/lang/String;→equals(Ljava/lang/Object;)Z
229
230     move-result v5
231
232     if-eqz v5, :cond_0
233
234     const-string v5, "call it"
235
236     return-object v5
237

```

```

229
230     move-result v5
231
232     if-eqz v5, :cond_0
233
234     invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;→sesame(Ljava/lang/String;)Ljava/lang/String;
235
236     move-result-object v5
237
238     return-object v5
239

```

Hình 50: Thay đổi gọi về cardamom của lệnh if

Thực hiện patch app và kí:

```

keystore password for signer #1:
└─(kali㉿kali)-[~/Desktop/Baitapluyentap]
$ apktool b five -o fivev2.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.7.0-dirty
I: Checking whether sources has changed ...
I: Smaling smali folder into classes.dex ...
I: Checking whether resources has changed ...
I: Copying raw resources ...
I: Building apk file ...
I: Copying unknown files/dir ...
I: Built apk into: fivev2.apk

└─(kali㉿kali)-[~/Desktop/Baitapluyentap]
$ apksigner sign --ks five.keystore fivev2.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:

└─(kali㉿kali)-[~/Desktop/Baitapluyentap]
$ 

```

Hình 51: Repatch apk

Sau khi cài đặt lại ứng dụng, chúng ta tiếp tục với bước tính toán:

```

com/hellocmu/picotf/FlagstaffHill.class x com/hellocmu/picotf/FlagstaffHill.class x com/hellocmu/picotf/FlagstaffHill.class x
JD-GUI Decompiler
Match case
1 package com.hellocmu.picotf;
2
3 import android.content.Context;
4
5 public class FlagstaffHill {
6     public static native String cardamom(String paramString);
7
8     public static String getFlag(String paramString, Context paramContext) {
9         StringBuilder stringBuilder1 = new StringBuilder("aaa");
10        StringBuilder stringBuilder2 = new StringBuilder("aaa");
11        StringBuilder stringBuilder3 = new StringBuilder("aaa");
12        StringBuilder stringBuilder4 = new StringBuilder("aaa");
13        stringBuilder1.setCharAt(0, (char)(stringBuilder1.charAt(0) + 4));
14        stringBuilder1.setCharAt(1, (char)(stringBuilder1.charAt(1) + 19));
15        stringBuilder1.setCharAt(2, (char)(stringBuilder1.charAt(2) + 18));
16        stringBuilder2.setCharAt(0, (char)(stringBuilder2.charAt(0) + 7));
17        stringBuilder2.setCharAt(1, (char)(stringBuilder2.charAt(1) + 0));
18        stringBuilder2.setCharAt(2, (char)(stringBuilder2.charAt(2) + 1));
19        stringBuilder3.setCharAt(0, (char)(stringBuilder3.charAt(0) + 0));
20        stringBuilder3.setCharAt(1, (char)(stringBuilder3.charAt(1) + 11));
21        stringBuilder3.setCharAt(2, (char)(stringBuilder3.charAt(2) + 15));
22        stringBuilder4.setCharAt(0, (char)(stringBuilder4.charAt(0) + 14));
23        stringBuilder4.setCharAt(1, (char)(stringBuilder4.charAt(1) + 20));
24        stringBuilder4.setCharAt(2, (char)(stringBuilder4.charAt(2) + 15));
25        return paramString.equals("").concat(stringBuilder3.toString()).concat(stringBuilder2.toString()).concat(stringBuilder1.toString()).concat(stringBuilder4.toString()) ? cardamom(paramStri
26    }
27 }

```

Hình 52: Code sau khi đổi

stringBuilder1:

- Ký tự đầu tiên (index 0): 'a' + 4 = 'e'
- Ký tự thứ hai (index 1): 'a' + 19 = 't'
- Ký tự thứ ba (index 2): 'a' + 18 = 's'

stringBuilder2:

- Ký tự đầu tiên (index 0): 'a' + 7 = 'h'
- Ký tự thứ hai (index 1): 'a' + 0 = 'a'
- Ký tự thứ ba (index 2): 'a' + 1 = 'b'

stringBuilder3:

- Ký tự đầu tiên (index 0): 'a' + 0 = 'a'
- Ký tự thứ hai (index 1): 'a' + 11 = 'l'
- Ký tự thứ ba (index 2): 'a' + 15 = 'p'

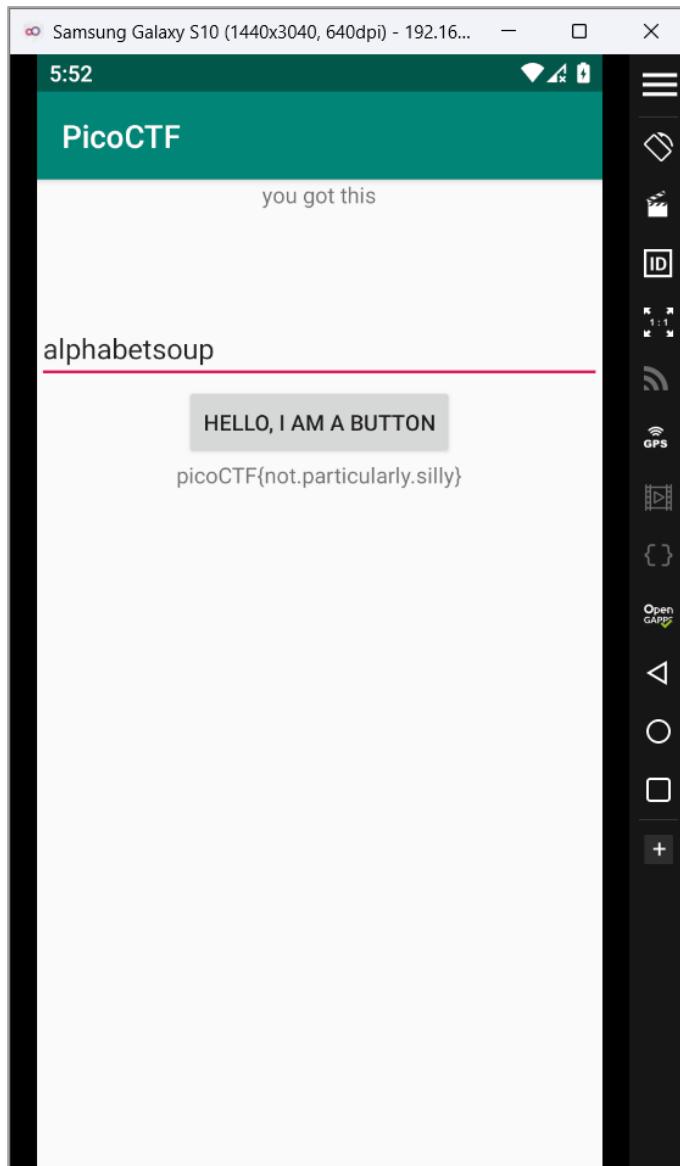
stringBuilder4:

- Ký tự đầu tiên (index 0): 'a' + 14 = 'o'
- Ký tự thứ hai (index 1): 'a' + 20 = 'u'
- Ký tự thứ ba (index 2): 'a' + 15 = 'p'

Nối các chuỗi trên theo thứ tự 3-2-1-4, chúng ta được keyword:

**“alphabetsoup”**

Nhập keyword và lấy flag:



Hình 53: Lấy được flag

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