

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Tên chủ đề: CTF CHALLENGES LAB 04

GVHD: Ngô Đức Hoàng Sơn

1. THÔNG TIN CHUNG:

Lớp: NT213.P12.ANTT

STT	Họ và tên	MSSV	Email
1	Hồ Vĩnh Khánh	22520633	22620633@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng
1	Level 1	100%
2	Level 2	100%
3	Level 3	100%
4	Level 4	100%
5	Level 5	100%
6	Level 6	100%
7	Level 7	100%
8	Level 8	100%
9	Level 9	100%
10	Level 10	100%
11	Level 11	100%
12	Level 12	100%
13	Challenge 1	100%
14	Challenge 2	100%
15	Challenge 3	100%
16	Challenge 4	100%
17	Challenge 5	100%
Điểm tự đánh giá		10/10

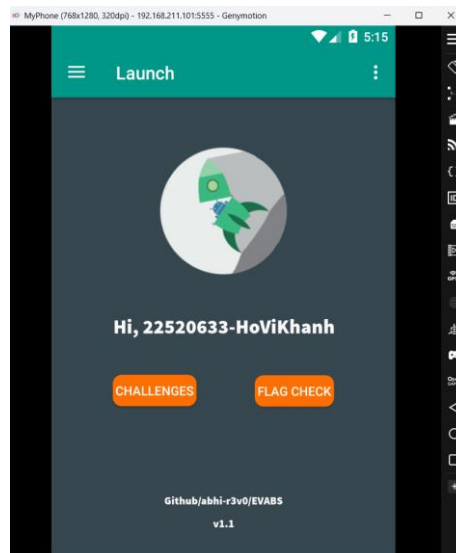
¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

D. CHALLENGES CTF (BÀI TẬP BẢO MẬT ỨNG DỤNG)

D.1 EVABS

Giao diện sau khi tải file EVABsv5.apk về máy ảo android.



Level 1: Debug Me

Bắt đầu với level đầu tiên ta thấy



Đầu tiên chạy lệnh adb shell ps để tìm PID của ứng dụng

```

u0_a3      1565    242 1175724  72364 ep_poll    e882dbb9 S com.android.providers.calenda
u0_a51     1588    242 1184104  78708 ep_poll    e882dbb9 S com.android.email
u0_a64     1632    242 1186512  92536 ep_poll    e882dbb9 S com.android.messaging
system     1702    242 1173948  66660 ep_poll    e882dbb9 S com.genymotion.superuser
u0_a9      1731    242 1173704  65624 ep_poll    e882dbb9 S android.ext.services
u0_a6      1754    242 1173720  67152 ep_poll    e882dbb9 S com.android.defcontainer
u0_a47     1781    242 1178368  68240 ep_poll    e882dbb9 S com.android.gallery3d
u0_a67     1809    242 1244528 111628 ep_poll    e882dbb9 S com.revo.evabs
root       1834      2      0      0 worker_thread 0 S [kworker/6:2]

```

Khi tìm được PID của ứng dụng rồi, chạy lệnh adb logcat --pid=[app's pid]. Trong trường hợp này là **adb logcat --pid=1809**

Khi đọc được log rồi, click vào button "LOG THE KEY" và xem flag ở logcat.

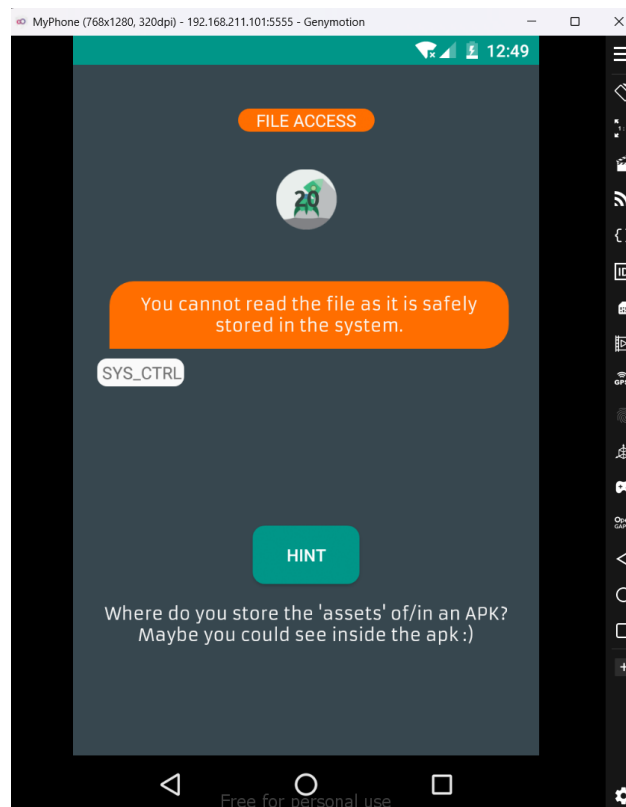
```

11-20 17:11:31.742 1809 1814 I zygote : Increasing code cache capacity to 256KB
11-20 17:11:31.742 1809 1814 I zygote : JIT allocated 71KB for compiled code of void
.AttributeSet, int,
int)
11-20 17:11:31.744 1809 1814 I zygote : Compiler allocated 4MB to compile void android
.buteSet, int, int)
11-20 17:11:34.036 1809 1814 I zygote : Do full code cache collection, code=102KB, da
11-20 17:11:34.043 1809 1814 I zygote : After code cache collection, code=102KB, data
11-20 17:11:42.326 1809 1814 I zygote : Do partial code cache collection, code=115KB, da
11-20 17:11:42.326 1809 1814 I zygote : After code cache collection, code=115KB, data
11-20 17:11:42.326 1809 1814 I zygote : Increasing code cache capacity to 512KB
11-20 17:11:57.166 1809 1814 I zygote : Do full code cache collection, code=249KB, da
11-20 17:11:57.168 1809 1814 I zygote : After code cache collection, code=175KB, data
11-20 17:12:00.627 1809 1814 I zygote : Do partial code cache collection, code=183KB, da
11-20 17:12:00.628 1809 1814 I zygote : After code cache collection, code=183KB, data
11-20 17:12:00.629 1809 1814 I zygote : Increasing code cache capacity to 1024KB
11-20 17:12:00.632 1809 1814 I zygote : JIT allocated 71KB for compiled code of void
.AttributeSet, int,
int)
11-20 17:12:00.632 1809 1814 I zygote : Compiler allocated 4MB to compile void android
.buteSet, int, int)
11-20 17:12:01.206 1809 1830 D OpenGLRenderer: endAllActiveAnimators on 0xc5d68380 (Ap
11-20 17:17:38.051 1809 1814 I zygote : Do full code cache collection, code=469KB, da
11-20 17:17:38.052 1809 1814 I zygote : After code cache collection, code=438KB, data
11-20 17:19:47.408 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:19:48.629 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:19:50.896 1809 1809 I chatty : uid=10067(u0_a67) com.revo.evabs identical 4
11-20 17:19:51.330 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:19:54.706 1809 1814 I zygote : Do partial code cache collection, code=495KB, da
11-20 17:19:54.707 1809 1814 I zygote : After code cache collection, code=495KB, data
11-20 17:19:54.707 1809 1814 I zygote : Increasing code cache capacity to 2MB
11-20 17:29:31.356 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:29:31.351 1809 1809 I com.revo.evabs: type=1400 audit(0.0:1358): avc: denied
0:c512,c768 tcontext=u:r:init:s0 tclass=unix_dgram_socket permissive=1
11-20 17:29:31.356 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:29:38.828 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}
11-20 17:29:43.372 1809 1809 I chatty : uid=10067(u0_a67) com.revo.evabs identical 6
11-20 17:29:43.703 1809 1809 D ** SYS_CTRL **: EVABS{logging_info_never_safel}

```

Flag: EVABS{logging_info_never_safel}

Level 2: File Access



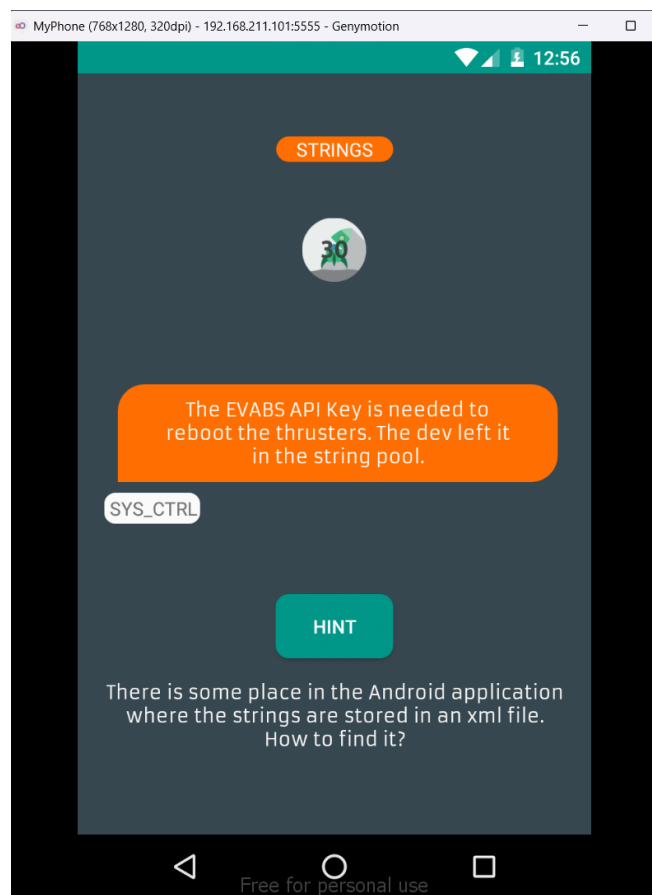
Một trong số những nơi ứng dụng android sử dụng để lưu trữ tài nguyên là **assets** folder. Tất cả những tài nguyên này đều được nén vào file apk cùng với các file code của ứng dụng. Để lấy được tài nguyên thì chỉ cần unzip file apk.

Vào thư mục assets và chúng ta sẽ thấy 1 file chứa flag.

```
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5$ cd assets/  
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5/assets$ ls  
fonts  secrets  
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5/assets$ cat secrets  
EVABS{fil3s !n ass3ts ar3 eas!lv hackabl3}vika2004@Kelvin:/mnt/c/Users/hovik/  
Downloads/EVABSV5/assets$
```

Flag: EVABS{fil3s !n ass3ts ar3 eas!lv hackabl3}

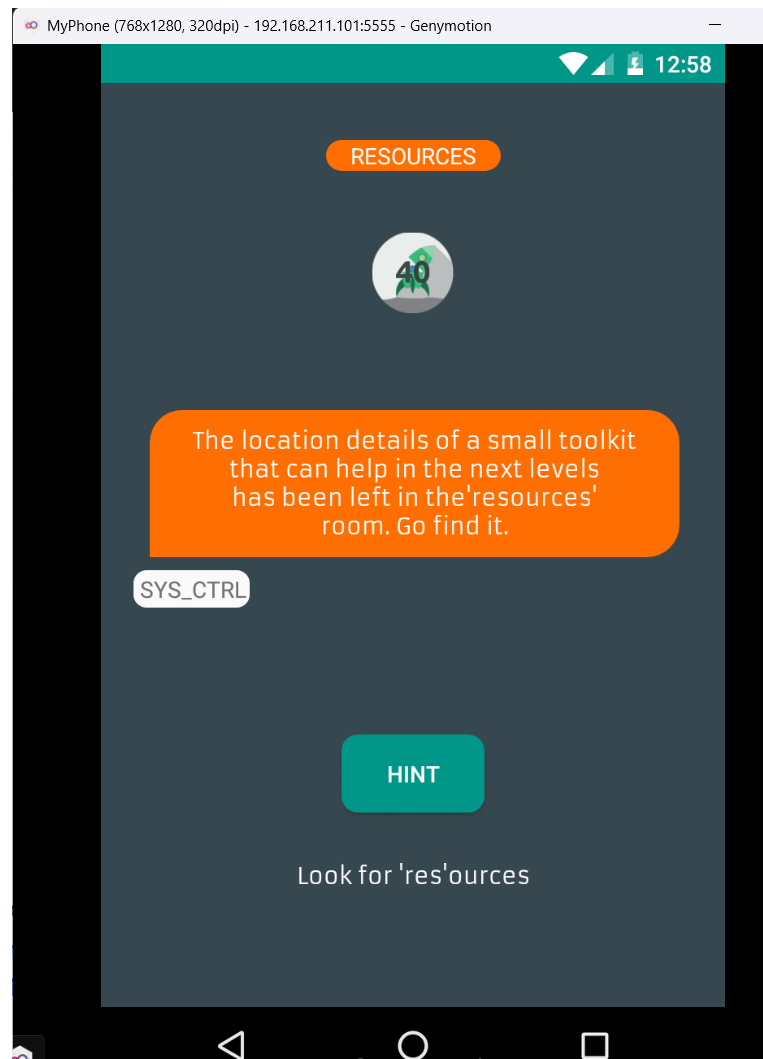
Level 3: Strings



Khi dùng bytecode viewer để truy cập file strings.xml thì tìm thấy flag như gợi ý.

```
90 <string name="permission_rationale">contacts permissions are needed for providing email
91 completions."</string>
92 <string name="project_id">evabs-c0e8b</string>
93 <string name="prompt_email">Email</string>
94 <string name="prompt_password">Password (optional)</string>
95 <string name="search_menu_title">Search</string>
96 <string name="section_format">Hello World from section: %1$d</string>
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
```

Flag: EVABS{saf3ly_st0red_in_Strings?}

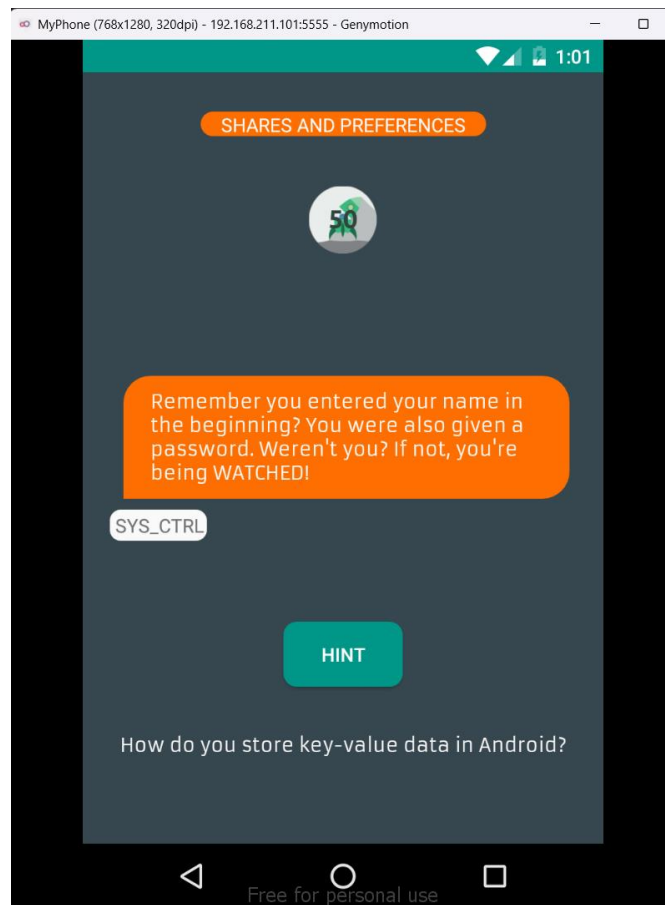
Level 4: Resource

Qua gợi ý, biết rằng flag nằm trong 1 file nào đó có trong thư mục **res**. Vì vậy ta dùng grep để tìm file flag: **grep -r "EVABS{" ***. Và biết được flag nằm tại **res/raw/link.txt**

```
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5$ cd res/  
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5/res$ grep -r "EVABS{" *  
Binary file layout/activity_flagcheck.xml matches  
Binary file layout-v17/activity_flagcheck.xml matches  
raw/link.txt:EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}  
vika2004@Kelvin:/mnt/c/Users/hovik/Downloads/EVABSV5/res$
```

Flag: EVABS{th!s_plac3_is_n0t_as_s3cur3_as_it_l00ks}

Level 5: Shares and Preferences



SharedPreferences là một API lưu trữ dữ liệu vĩnh viễn trong các file XML. Dữ liệu được lưu trữ bởi SharedPreferences object có cấu trúc dạng key - value.

SharedPreferences object có thể được khai báo cho tất cả ứng dụng sử dụng, hoặc khai báo private. Dữ liệu được lưu trong các file XML tại `/data/data/<package-name>/shared_prefs/*.xml`.

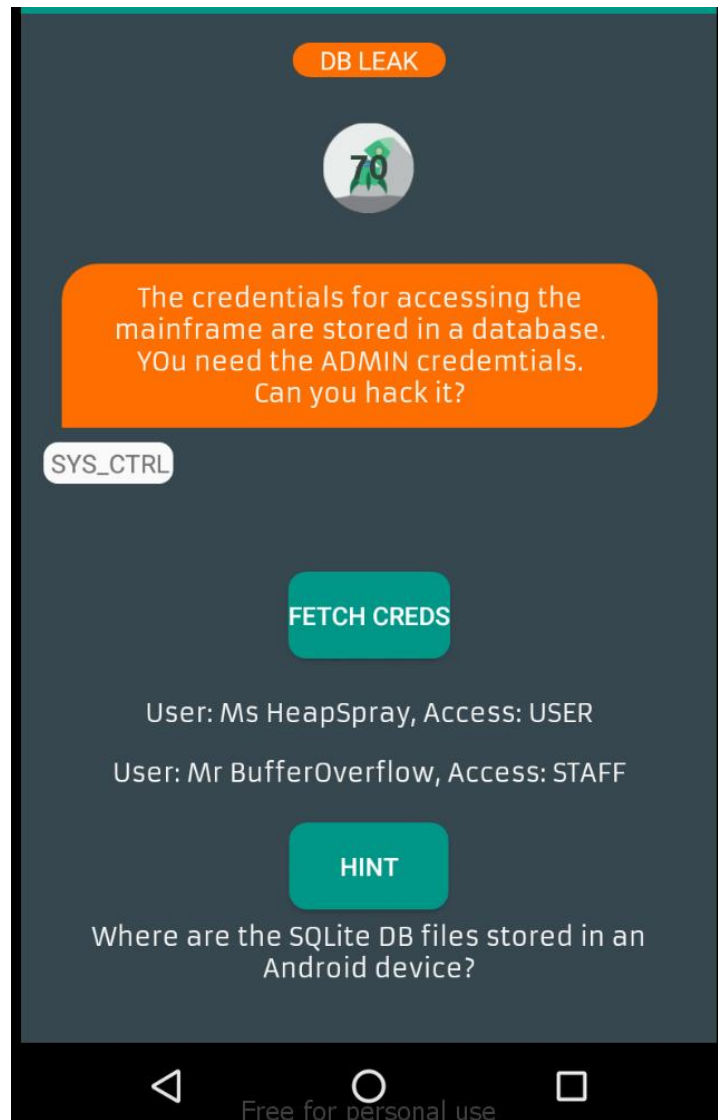
Sử dụng adb shell để truy cập vào hệ thống android bằng lệnh `cd /data/data/com.revo.evabs/shared_prefs` và chạy lệnh `grep -r "EVABS" *` để tìm flag giấu trong các file xml.

```
C:\platform-tools>adb shell
genymotion:/ # cd /data/data/com.revo.evabs/shared_prefs
genymotion:/data/data/com.revo.evabs/shared_prefs # grep -r "EVABS" *
DETAILS.xml: <string name="password">EVABS{shar3d_pr3fs_c0uld_be_c0mpromiz3ds}</string>
genymotion:/data/data/com.revo.evabs/shared_prefs # |
```

Flag: EVABS{shar3d_pr3fs_c0uld_be_c0mpromiz3ds}

Level 6: DB Leak

Các ứng dụng Android cũng cần lưu trữ dữ liệu local. Một trong những nơi ứng dụng sử dụng để lưu trữ dữ liệu là SQLite DB. Các database này luôn nằm tại `/data/data/<package-name>/databases`



Sau khi click button "FETCH Creds" thì sẽ có DB được sinh ra trong Local Storage.

Giờ thì chạy adb shell "**ls /data/data/com.revo.evabs/databases**" để kiểm tra xem có những db nào. Ở đây chỉ có 1 db là MAINFRAME_ACCESS.

```
C:\platform-tools>adb shell
genymotion:/ # ls /data/data/com.revo.evabs/databases
MAINFRAME_ACCESS  MAINFRAME_ACCESS-journal
genymotion:/ #
```

Pull DB đó về máy thật để mở bằng SQLite browser bằng lệnh adb pull "**/data/data/com.revo.evabs/databases/MAINFRAME_ACCESS**". Xem các bảng thấy flag là password của user Dr.l33t có role admin.

```
C:\platform-tools>adb pull "/data/data/com.revo.evabs/databases/MAINFRAME_ACCESS"
/data/data/com.revo.evabs/databases/MAINFRAME_AC...led, 0 skipped. 1.8 MB/s (16384 bytes in 0.009s)
C:\platform-tools>
```


DB Browser for SQLite - C:\platform-tools\MAINFRAME_ACCESS

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo

Database Structure Browse Data Edit Pragmas Execute SQL

Table: CREDs

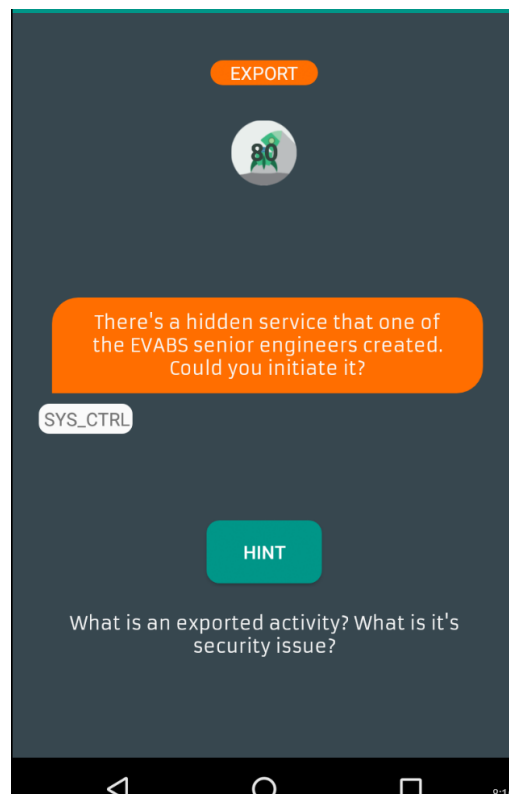
	admin	pass	access
	Filter	Filter	Filter
1	Dr.133t	<u>EVABS{sqlite_is_not_safe}E</u>	ADMIN
2	Mr BufferOverflow	OxNotSecureSQLite_	STAFF
3	Ms HeapSpray	SQLite_exploit	USER

Flag: EVABS{sqlite_is_not_safe}

Level 7: Export

Trong ứng dụng Android có khái niệm **activity**. Activity trong Android là nơi diễn ra mọi hoạt động tương tác với người dùng, bởi vì tất cả các màn hình ứng dụng đều phải được “đính” trên một Activity.

Thông tin về các activity được lưu trong file **AndroidManifest.xml**, xuất hiện trong các thẻ **<activity** **>**. Trong thẻ này có 1 thuộc tính quan trọng là **android:exported**. Nếu thuộc tính này có giá trị **true** thì activity đó có thể bị kích hoạt bởi các ứng dụng khác.

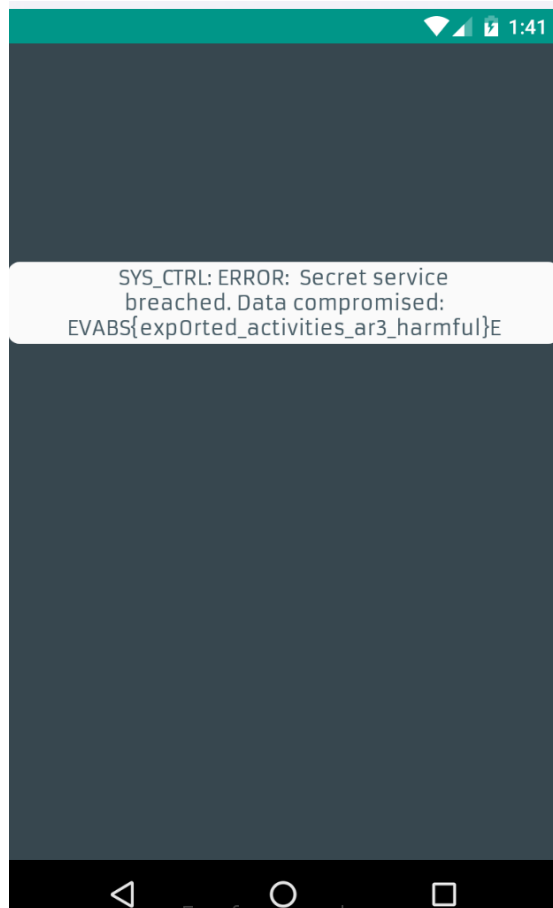


Khi kiểm tra manifest, ở ngay dòng thứ 4 trong đã có thông tin về một Activity bị exported:

```
<activity android:exported="true"
android:name="com.revo.evabs.ExportedActivity"/>
```

Khi một activity bị exported, chúng ta có thể khởi động nó bằng adb. Sử dụng adb để trigger các exported activity bằng lệnh: **adb shell am start -n [package name]/[package name].[exported activity]**.

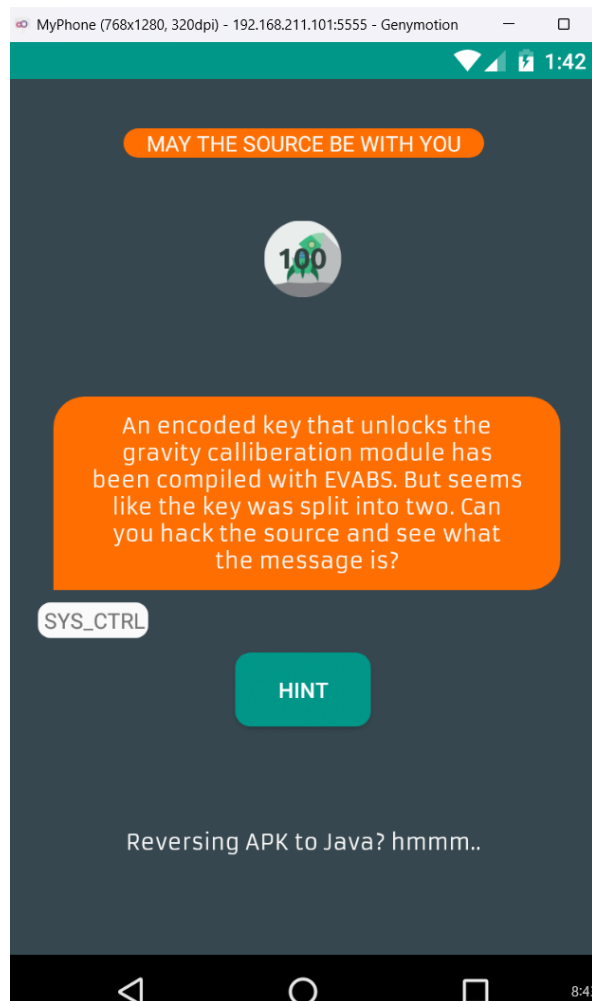
```
C:\platform-tools>adb shell am start -n com.revo.evabs/com.revo.evabs.ExportedActivity
Starting: Intent { cmp=com.revo.evabs/.ExportedActivity }
C:\platform-tools>
```



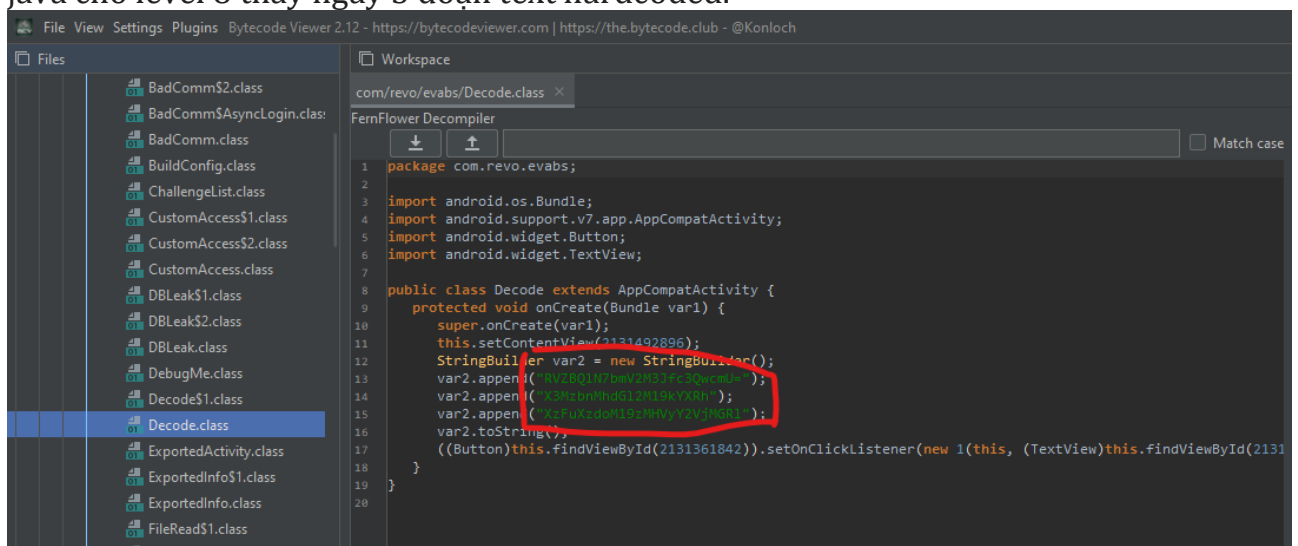
Flag: EVABS{exported_activities_ar3_harmful}

Level 8: Decode

Hardcoded các string quan trọng là không an toàn, khi hacker có thể dễ dàng reverse file apk và tìm được các đoạn string này.



Reverse sang code java bằng Bytecode viewer, sau đó mở file **Decode.class** - file code java cho level 8 thấy ngay 3 đoạn text hardcoded.



Decode B64 với 3 string này ta có được flag.

Theo gợi ý của level này thì chúng ta đi phân tích file SmaliInject.java:

SmaliInject.class:

```

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 SmaliInject extends AppCompatActivity {
9     String SIGNAL = "LAB_OFF";
10
11     static {
12         System.loadLibrary("native-lib");
13     }
14
15     protected void onCreate(Bundle var1) {
16         super.onCreate(var1);
17         this setContentView(2131492906);
18         Button var6 = (Button) this.findViewById(2131361846);
19         Button var2 = (Button) this.findViewById(2131361845);
20         TextView var4 = (TextView) this.findViewById(2131362096);
21         TextView var5 = (TextView) this.findViewById(2131362104);
22         TextView var7 = (TextView) this.findViewById(2131362095);
23         TextView var3 = (TextView) this.findViewById(2131362087);
24         var2.setOnClickListener(new 1(this, var5));
25         var6.setOnClickListener(new 2(this, var7, var4, var3));
26     }
27
28     public native String stringFromSmali();
29 }

```

SmaliInject\$2.class:

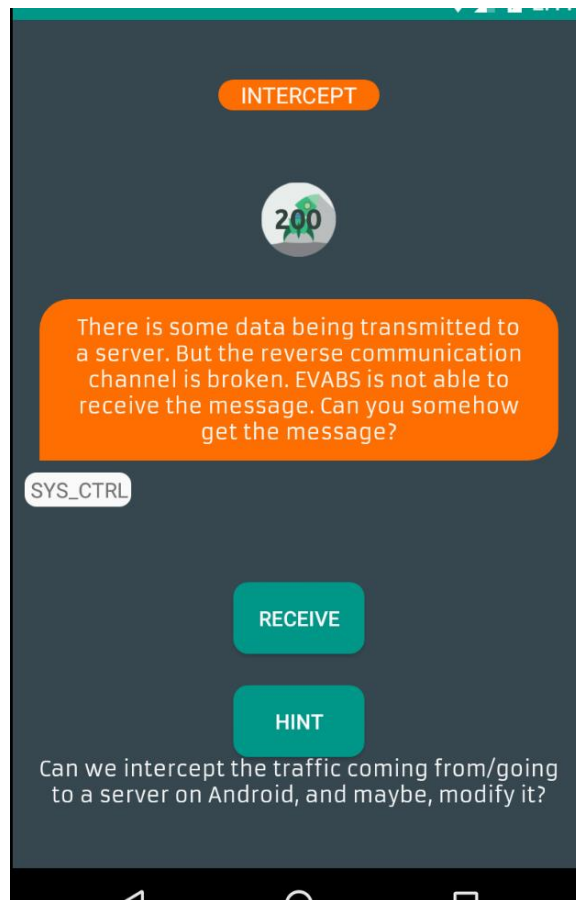
```

1 package com.revo.evabs;
2
3 import android.view.View;
4 import android.widget.TextView;
5
6 class SmaliInject$2 implements View.OnClickListener {
7     final SmaliInject this$0;
8     final TextView val$labstat;
9     final TextView val$stvflag;
10    final TextView val$stvlaboff;
11
12    SmaliInject$2(SmaliInject var1, TextView var2, TextView var3, TextView var4) {
13        this.this$0 = var1;
14        this.val$stvlaboff = var2;
15        this.val$labstat = var3;
16        this.val$stvflag = var4;
17    }
18
19    public void onClick(View var1) {
20        String var3 = this.this$0.stringFromSmali();
21        if (this.this$0.SIGNAL.equals("LAB_ON")) {
22            this.val$stvlaboff.setText("SYS_CTRL_CODE: LAB_ON");
23            this.val$labstat.setText("SYS_CTRL: ACCESS_GRANTED: LAB_UNLOCKED");
24            TextView var4 = this.val$stvflag;
25            StringBuilder var2 = new StringBuilder();
26            var2.append("SYS_CTRL_CODE:");
27            var2.append(var3);
28            var2.append("\n");
29            var4.setText(var2.toString());
30        } else {
31            this.val$stvlaboff.setText("SYS_CTRL_CODE: LAB_OFF");
32            this.val$labstat.setText("SYS_CTRL: ACCESS_DENIED");
33        }
34    }
35 }

```

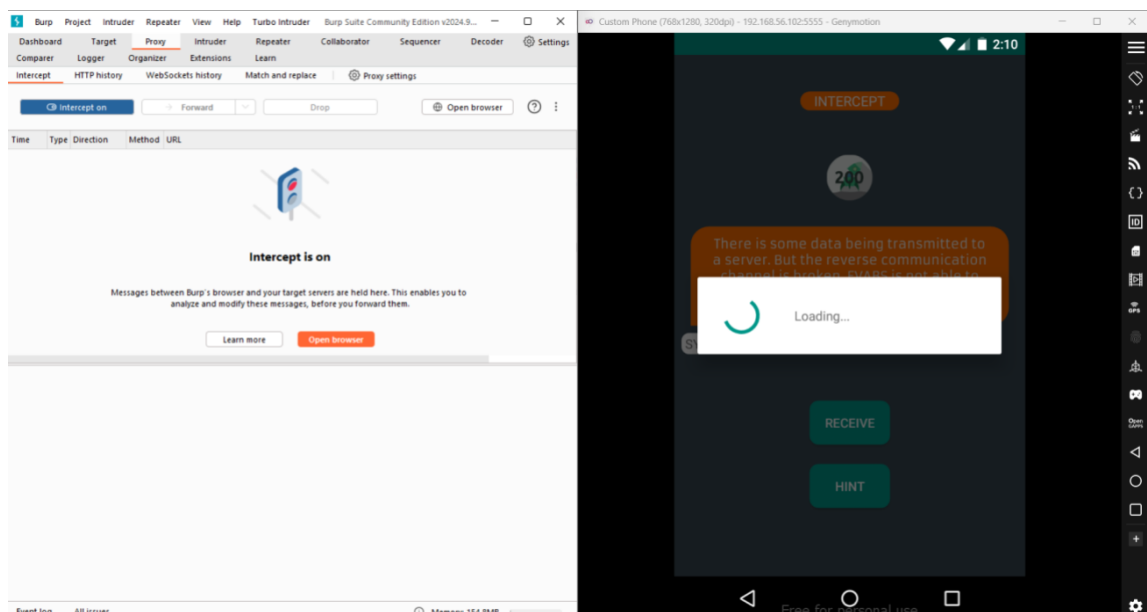
Thấy nếu biến SIGNAL = "LAB_ON" thì chương trình sẽ in ra flag. Nhưng biến SIGNAL đang được đặt mặc định là "LAB_OFF"

Chúng ta có thể sửa code và build lại ứng dụng mới mà vẫn dùng được các chức năng bình thường do Dev đã không có cơ chế kiểm tra độ toàn vẹn code.

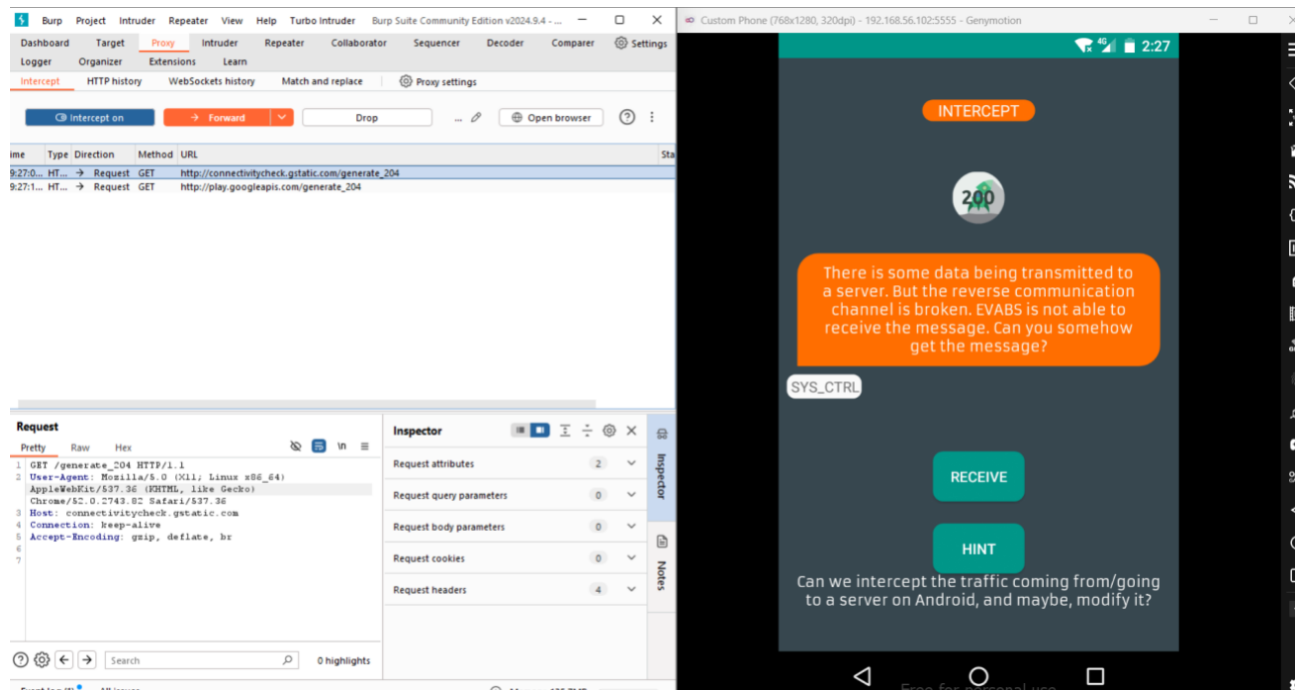
Level 10: Intercept

Theo như gợi ý thì cần phải intercept request bằng Burpsuite để bắt lại gói tin khi nhấn RECEIVE

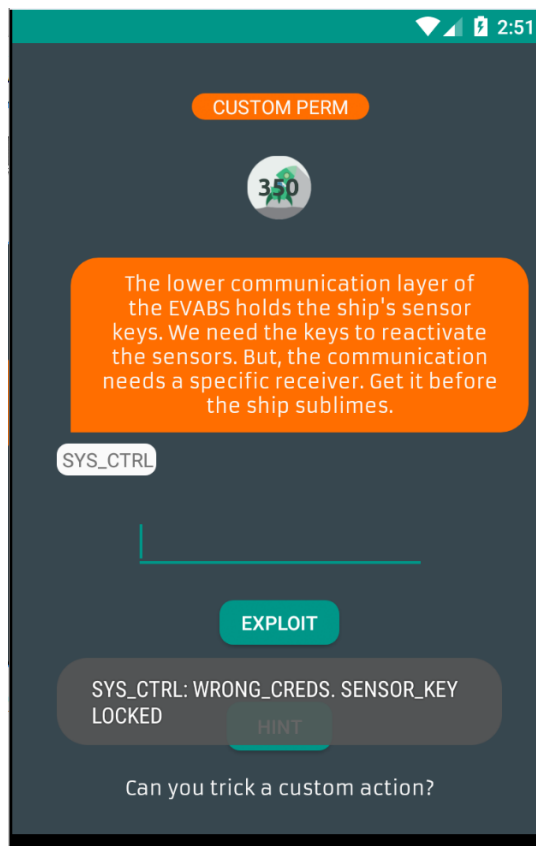
Cấu hình Proxy như sau để bắt được các request từ thiết bị android ra ngoài như ở lab4:



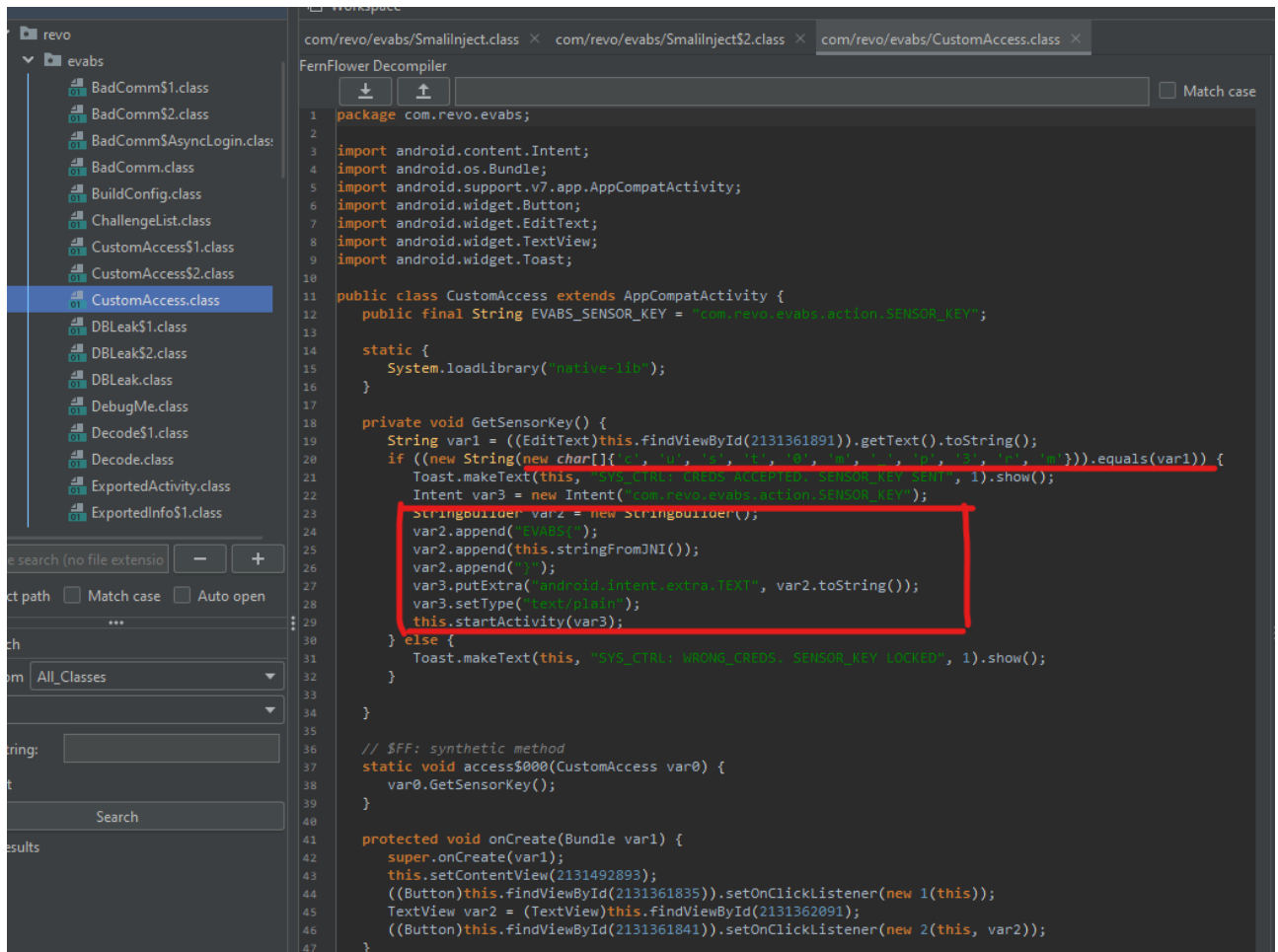
Sau khi config để intercept được request rồi thì chỉ cần Send to repeater và gửi request lên là được. Trong respond có chứa flag.



Level 11: Custom PERM



Ở Level này ta phải tìm đúng input mới được, xem source trên bytecode viewer để dàng thấy ngay input đúng là **cust0m_p3rm**.



Sau khi nhập đúng input thì flag sẽ được truyền vào intent **com.revo.evabs.action.SENSOR_KEY** bằng hàm **putExtra()**.

Viết chương trình thay đổi hàm putExtra để in ra flag.

```

1  import frida
2  import sys
3
4  Tabnine | Edit | Test | Explain | Document | Ask
5  def on_message(message, data):
6      print(message)
7
8  package = "com.revo.evabs"
9
10 jscode = """
11 Java.perform(function () {
12     send("[-] Starting hooks on android.content.Intent.putExtra");
13     var intent = Java.use("android.content.Intent");
14     intent.putExtra.overload("java.lang.String", "java.lang.String").implementation = function(var_1, var_2) {
15         send("[+] Flag: " + var_2);
16         return this.putExtra(var_1, var_2);
17     };
18 });
19 """
20 device = frida.get_device_manager().add_remote_device("127.0.0.1:27042")
21 pid = device.spawn([package])
22 process = device.attach(pid)
23
24
25
26 script = process.create_script(jscode)
27 script.on("message", on_message)
28 print("[*] Hooking into", package)
29 script.load()
30
31
32 device.resume(pid)
33
34 sys.stdin.read()

```

Chạy tiến trình frida-server bên phía điện thoại

```

C:\platform-tools>adb shell
genymotion:/ # .frida-server &
[1] 2960

```

Chạy code Frida đồng thời mở mobile nhập đúng input cust0m_p3rm tìm được thì sẽ ra flag.

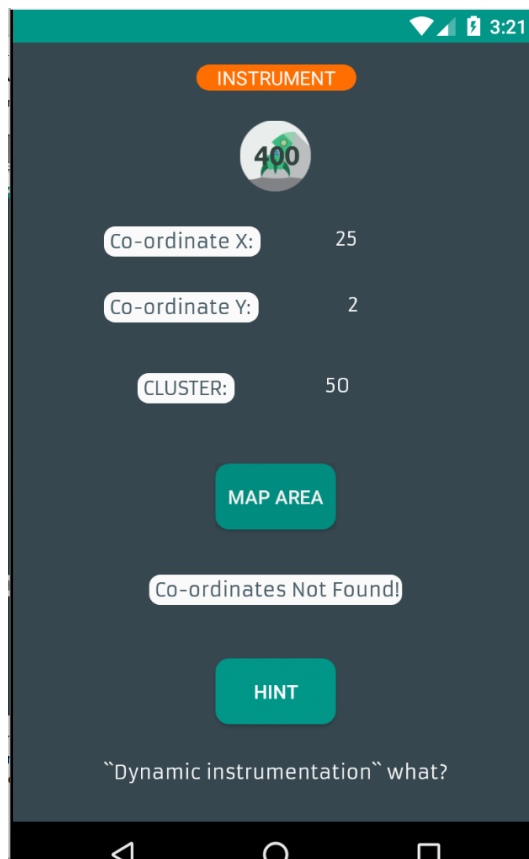
Thực thi ra kết quả

```

D:\1-2025\BaoMatWeb_UngDung\ThucHanh\Lab4>python lv.py
[*] Hooking into com.revo.evabs
{'type': 'send', 'payload': '[-] Starting hooks on android.content.Intent.putExtra'}
{'type': 'send', 'payload': '[+] Flag: EVABS{always_verify_packag3sa}'}

```

Flag: EVABS{always_verify_packag3sa}

Level 12 – Instrument

Khi click "MAP AREA" thì sẽ xuất hiện 2 tọa độ x và y cùng với 1 giá trị bằng $x * y$. Bài này cũng cần sử dụng Frida.

Khi mở file code Frida1.java phân tích, flag in ra nếu $(x=a*b) >$ giá trị ngẫu nhiên từ 0 đến 70 cộng 150. Vì cho sẵn $a = 25$, $b = 2$ nên dù giá trị ngẫu nhiên có là mấy x cũng không thể lớn hơn về bên phải được. Nên là hook vào hàm random, để điều kiện luôn đúng cho giá trị random là -150. (Khi đó $50 > 0$).

```

com/revo/evabs/frida1.class
FernFlow Decompiler
1 package com.revo.evabs;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.util.Log;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.TextView;
9 import java.util.Random;
10
11 public class Frida1 extends AppCompatActivity implements View.OnClickListener {
12     int a = 25;
13     int b = 2;
14     int x;
15
16     static {
17         System.loadLibrary("native-lib");
18     }
19
20     public void onClick(View var1) {
21         TextView var5 = (TextView) this.findViewById(2131361996);
22         TextView var3 = (TextView) this.findViewById(2131362132);
23         TextView var6 = (TextView) this.findViewById(2131362134);
24         TextView var4 = (TextView) this.findViewById(2131362142);
25         var3.setText(String.valueOf(this.a));
26         var6.setText(String.valueOf(this.b));
27         this.x = this.a * this.b;
28         int var2 = (new Random()).nextInt(70);
29         var4.setText(String.valueOf(this.x));
30         if (this.x > var2 + 150) {
31             var5.setText("WARNING: READY TO FLY! YOU ARE GOING HOME!");
32             Log.d("COMBRATZ!", this.stringFromJNI());
33         } else {
34             var5.setText("Co-ordinates Not Found!");
35         }
36     }
37 }

```

```

1 import frida
2 import sys
3
4 Tabnine | Edit | Test | Explain | Document | Ask
5 def on_message(message, data):
6     print(message)
7
8 package = "com.revo.evabs"
9
10 jscode = """
11 Java.perform(function () {
12     send("[-] Starting hooks on java.util.Random.nextInt");
13     var random = Java.use("java.util.Random");
14     random.nextInt.overload("int").implementation = function(var_1) {
15         send("[+] Hooked nextInt, returning: -150");
16         return -150; // Hook the nextInt method to always return -150
17     });
18 """"
19
20 # Connect to the USB device
21 device = frida.get_usb_device()
22
23 # Spawn the application if it's not already running
24 pid = device.spawn([package])
25 process = device.attach(pid)
26
27 # Create and load the script
28 script = process.create_script(jscode)
29 script.on("message", on_message)
30 print("[*] Hooking", package)
31 script.load()
32
33 # Resume the application to start execution
34 device.resume(pid)
35
36 # Keep the script running to listen for messages
37 sys.stdin.read()

```

```

D:\1-2023\BaoHatWeb UngDung\ThucHanh\Lab4>python lv12.py
[*] Hooking com.revo.evabs
{'type': 'send', 'payload': '[-] Starting hooks on java.util.Random.nextInt'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}
{'type': 'send', 'payload': '[+] Hooked nextInt, returning: -150'}

```

adb logcat ra thì thấy flag in ra ở log.

```
11-08 17:41:31.083 500 500 W batteryd: type=1400 audit(0.0:17338): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:31.402 3653 3653 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
11-08 17:41:32.064 3653 3653 I chatty : uid=10075(u0_a75) com.revo.evabs identical 1 line
11-08 17:41:32.807 3653 3653 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
11-08 17:41:33.083 500 500 W batteryd: type=1400 audit(0.0:17341): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:33.091 500 500 I chatty : uid=0(root) /system/bin/batteryd identical 18 lines
11-08 17:41:33.091 500 500 W batteryd: type=1400 audit(0.0:17360): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:33.437 3653 3653 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
11-08 17:41:35.082 500 500 W batteryd: type=1400 audit(0.0:17363): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:35.086 500 500 I chatty : uid=0(root) /system/bin/batteryd identical 18 lines
11-08 17:41:35.086 500 500 W batteryd: type=1400 audit(0.0:17382): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:35.136 3653 3653 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
11-08 17:41:36.376 3653 3653 I chatty : uid=10075(u0_a75) com.revo.evabs identical 3 lines
11-08 17:41:36.673 3653 3653 D CONGRATZ!: EVABS{a_dynam1c_h00k}E
11-08 17:41:37.078 500 500 W batteryd: type=1400 audit(0.0:17385): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
11-08 17:41:39.086 500 500 I chatty : uid=0(root) /system/bin/batteryd identical 38 lines
11-08 17:41:39.086 500 500 W batteryd: type=1400 audit(0.0:17426): avc: granted { read } for path="/dev/fuse" dev="tmpfs" ino=9364 scontext=u:r:init:s0
tcontext=u:object_r:fuse_device:s0 tclass=chr_file
```

Flag: EVABS{a_dynam1c_h00k}

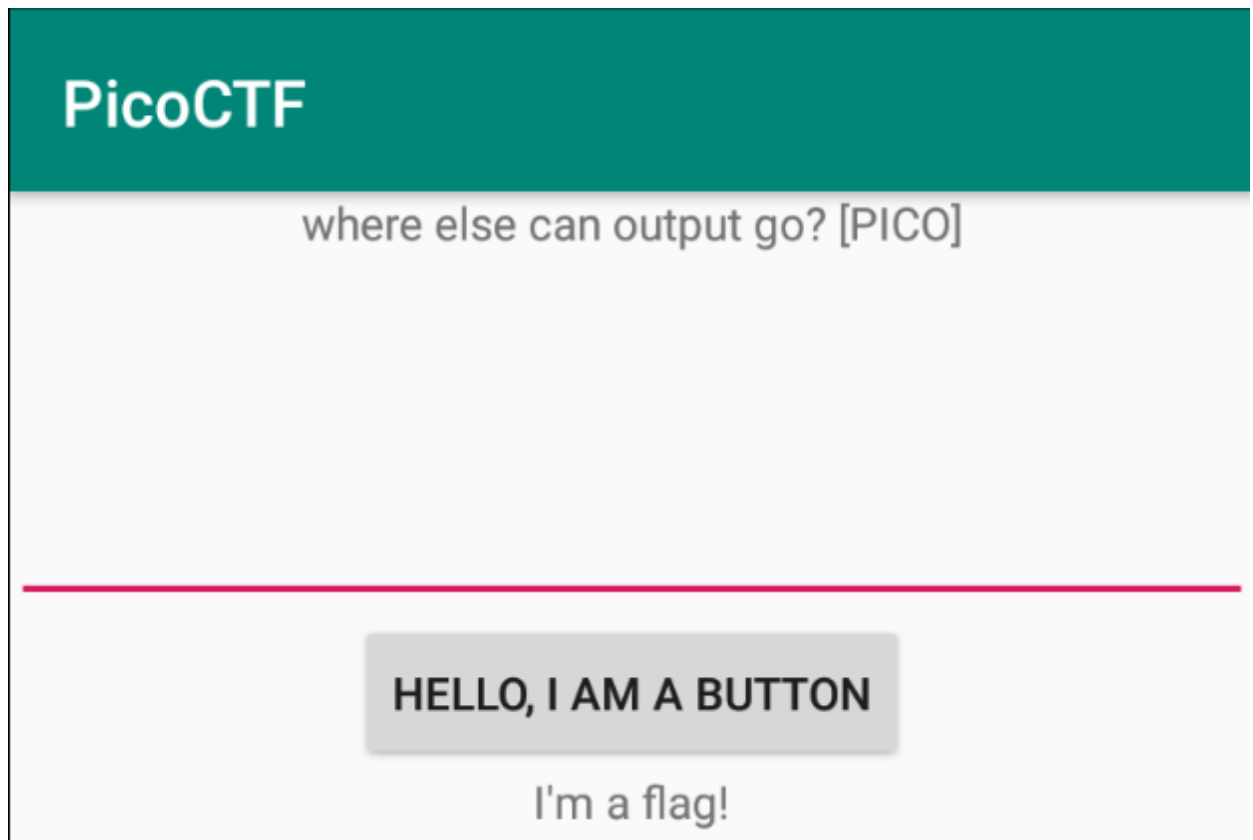
D.2 Droid

1. One.apk

Cài đặt one.apk

```
c:\platform-tools>adb install one.apk
Performing Streamed Install
Success

c:\platform-tools>|
```



Gợi ý của challenge là output có thể nằm ở đâu ngoài app. Dùng findstr format của flag là "picoCTF" từ logcat để tìm flag.

```
c:\platform-tools>adb logcat | findstr "picoCTF"
11-21 15:42:45.721 3432 3432 I PICO : picoCTF{a.moose.once.bit.my.sister}
```

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

2. Two.apk

Xóa apk cũ và cài đặt two.apk

```
c:\platform-tools>adb uninstall com.hellocmu.picoctf
Success

c:\platform-tools>adb install two.apk
Performing Streamed Install
Success

c:\platform-tools>|
```

PicoCTF

brute force not required

HELLO, I AM A BUTTON

I'm a flag!

Decompile apk với apktool

```
c:\platform-tools>apktool d two.apk
I: Using Apktool 2.10.0 on two.apk with 8 thread(s).
I: Loading resource table...
I: Baksmaling classes.dex...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\hovik\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Kiểm tra trong folder two/res/values/strings.xml

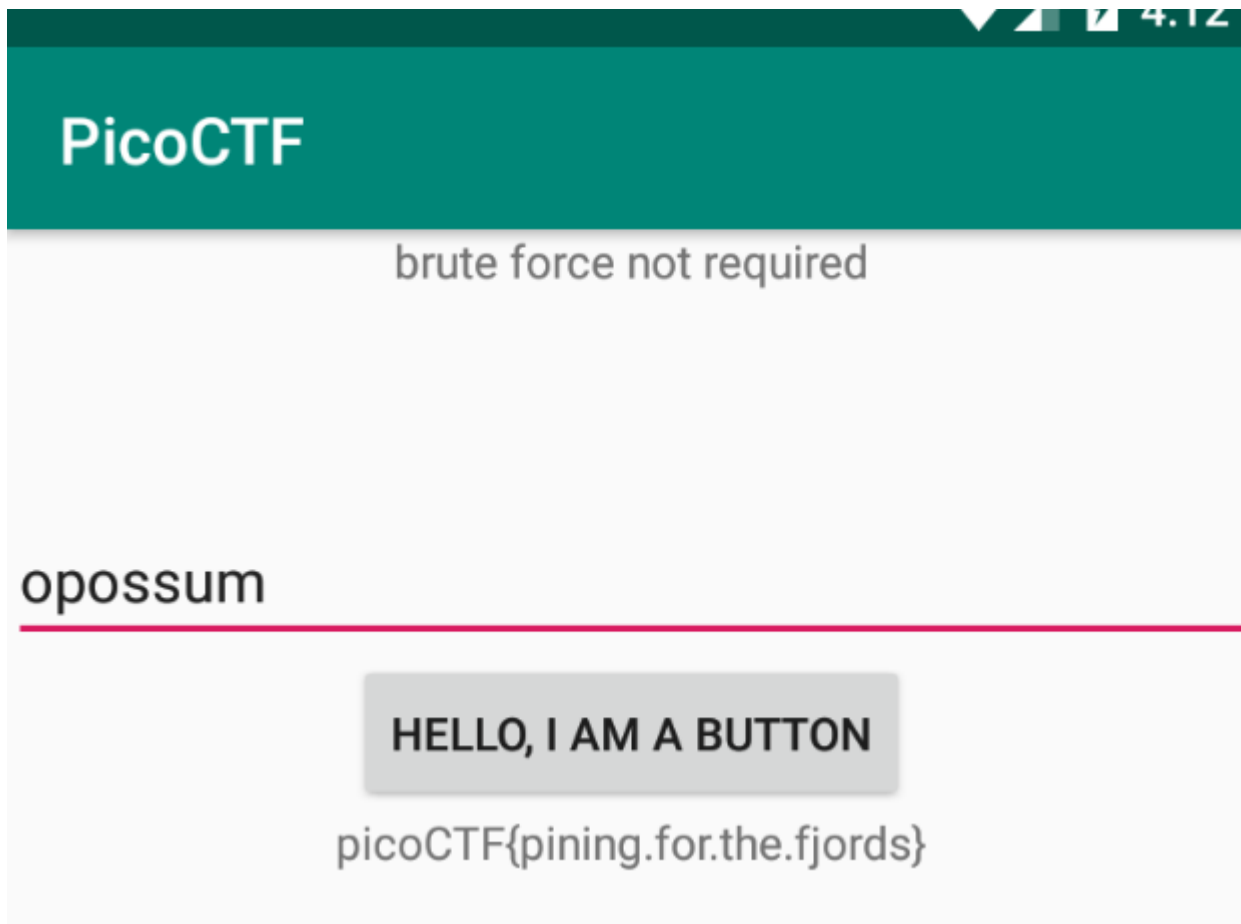
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<resources>
  <string name="abc_action_bar_home_description">Navigate home</string>
  <string name="abc_action_bar_up_description">Navigate up</string>
  <string name="abc_action_menu_overflow_description">More options</string>
  <string name="abc_action_mode_done">Done</string>
  <string name="abc_activity_chooser_view_see_all">See all</string>
  <string name="abc_activitychooserview_choose_application">Choose an app</string>
  <string name="abc_capital_off">OFF</string>
  <string name="abc_capital_on">ON</string>
  <string name="abc_font_family_body_1_material">sans-serif</string>
  <string name="abc_font_family_body_2_material">sans-serif-medium</string>
  <string name="abc_font_family_button_material">sans-serif-medium</string>
  <string name="abc_font_family_caption_material">sans-serif</string>
  <string name="abc_font_family_display_1_material">sans-serif</string>
  <string name="abc_font_family_display_2_material">sans-serif</string>
  <string name="abc_font_family_display_3_material">sans-serif</string>
  <string name="abc_font_family_display_4_material">sans-serif-light</string>
  <string name="abc_font_family_headline_material">sans-serif</string>
  <string name="abc_font_family_menu_material">sans-serif</string>
  <string name="abc_font_family_subhead_material">sans-serif</string>
  <string name="abc_font_family_title_material">sans-serif-medium</string>
  <string name="abc_menu_alt_shortcut_label">Alt+</string>
  <string name="abc_menu_ctrl_shortcut_label">Ctrl+</string>
  <string name="abc_menu_delete_shortcut_label">delete</string>
  <string name="abc_menu_enter_shortcut_label">enter</string>
  <string name="abc_menu_function_shortcut_label">Function+</string>
  <string name="abc_menu_meta_shortcut_label">Meta+</string>
  <string name="abc_menu_shift_shortcut_label">Shift+</string>
  <string name="abc_menu_space_shortcut_label">space</string>
  <string name="abc_menu_sym_shortcut_label">Sym+</string>
  <string name="abc_prepend_shortcut_label">Menu+</string>
  <string name="abc_search_hint">Search...</string>
  <string name="abc_searchview_description_clear">Clear query</string>
  <string name="abc_searchview_description_query">Search query</string>
  <string name="abc_searchview_description_search">Search</string>
  <string name="abc_searchview_description_submit">Submit query</string>
  <string name="abc_searchview_description_voice">Voice search</string>
  <string name="abc_shareactionprovider_share_with">Share with</string>
  <string name="abc_shareactionprovider_share_with_application">Share with %s</string>
  <string name="abc_toolbar_collapse_description">Collapse</string>
  <string name="app_name">PicoCTF</string>
  <string name="bat">mink</string>
  <string name="bear">margay</string>
  <string name="cottontail">shrew</string>
  <string name="gopher">armadillo</string>
  <string name="hint">brute force not required</string>
  <string name="manatee">caribou</string>
  <string name="myotis">jackrabbit</string>
  <string name="password">opossum</string>
  <string name="porcupine">blackbuck</string>
  <string name="porpoise">mouflon</string>
  <string name="search_menu_title">Search</string>
  <string name="skunk">elk</string>
  <string name="status_bar_notification_info_overflow">999+</string>
  <string name="vole">beaver</string>
</resources>

```

Password là opossum. Nhập vào ứng dụng để tìm flag.

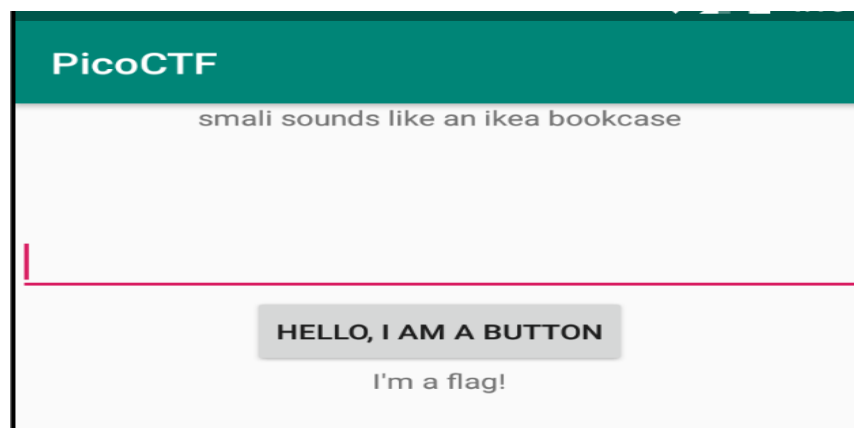


Flag: picoCTF{pining.for.the.fjords}

3. Three.apk

Cài đặt three.apk

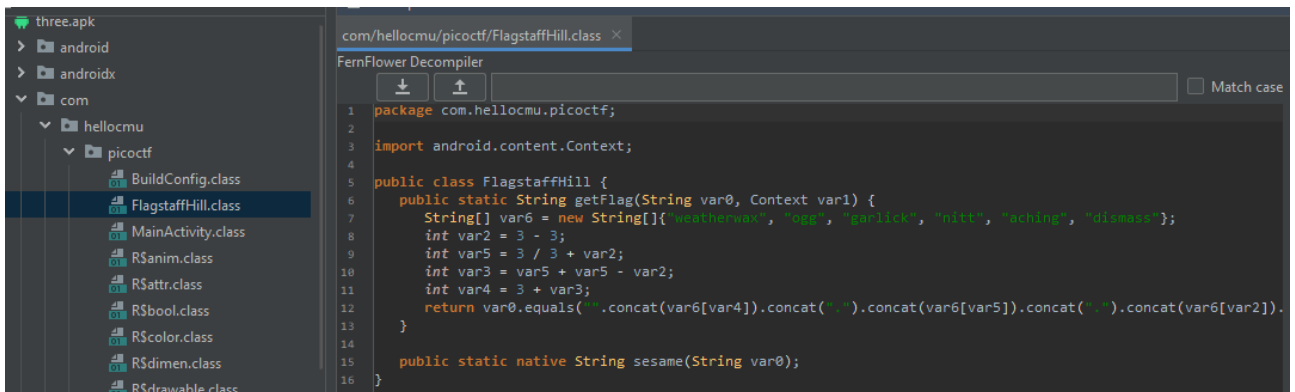
```
C:\platform-tools>adb install three.apk
Performing Streamed Install
Success
C:\platform-tools>
```



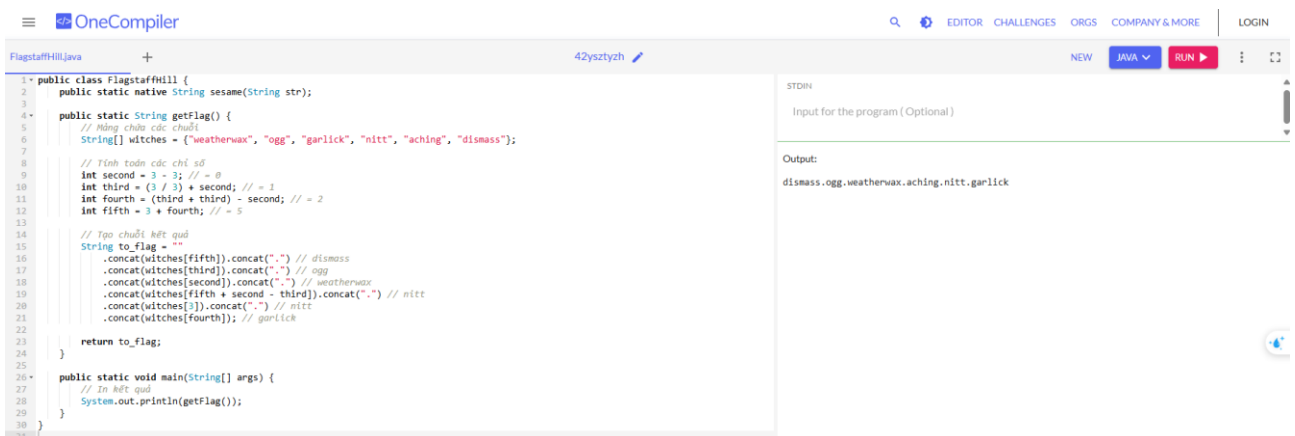
Decompile apk bằng apktool

```
C:\platform-tools>apktool d three.apk
I: Using Apktool 2.10.0 on three.apk with 8 thread(s).
I: Baksmaling classes.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\hovik\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

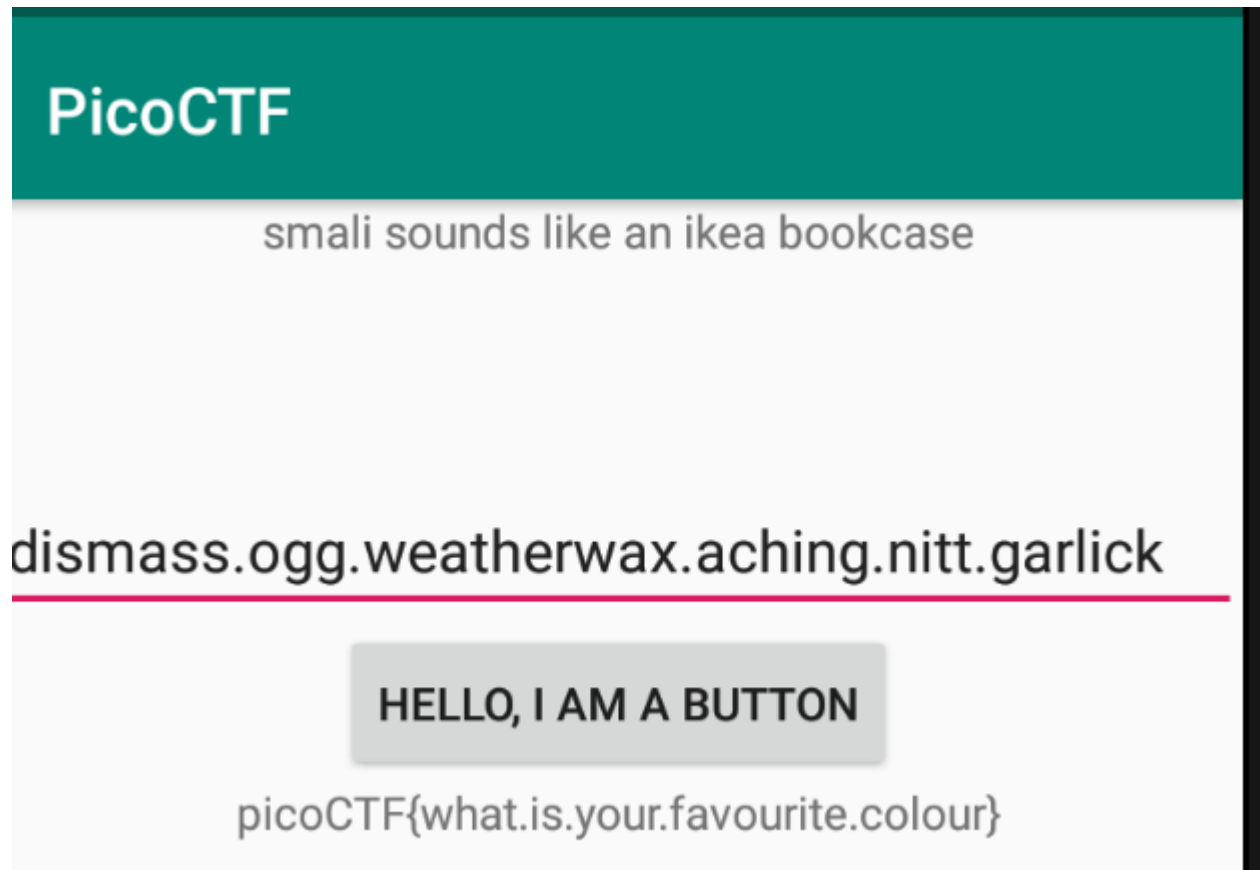
Xem code của FlagstaffHill.class



Viết lại hàm và chạy với compiler của Java



Nhận được password và nhập vào ứng dụng để lấy flag

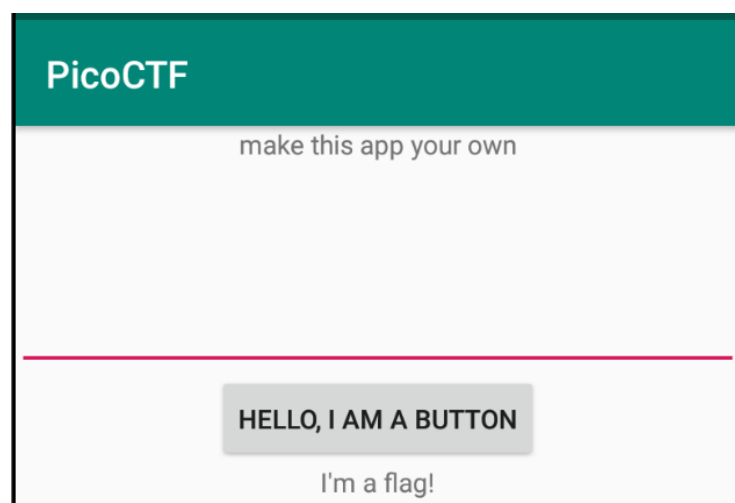


Flag: picoCTF{what.is.your.favourite.colour}

4. Four.apk

Cài đặt four.apk

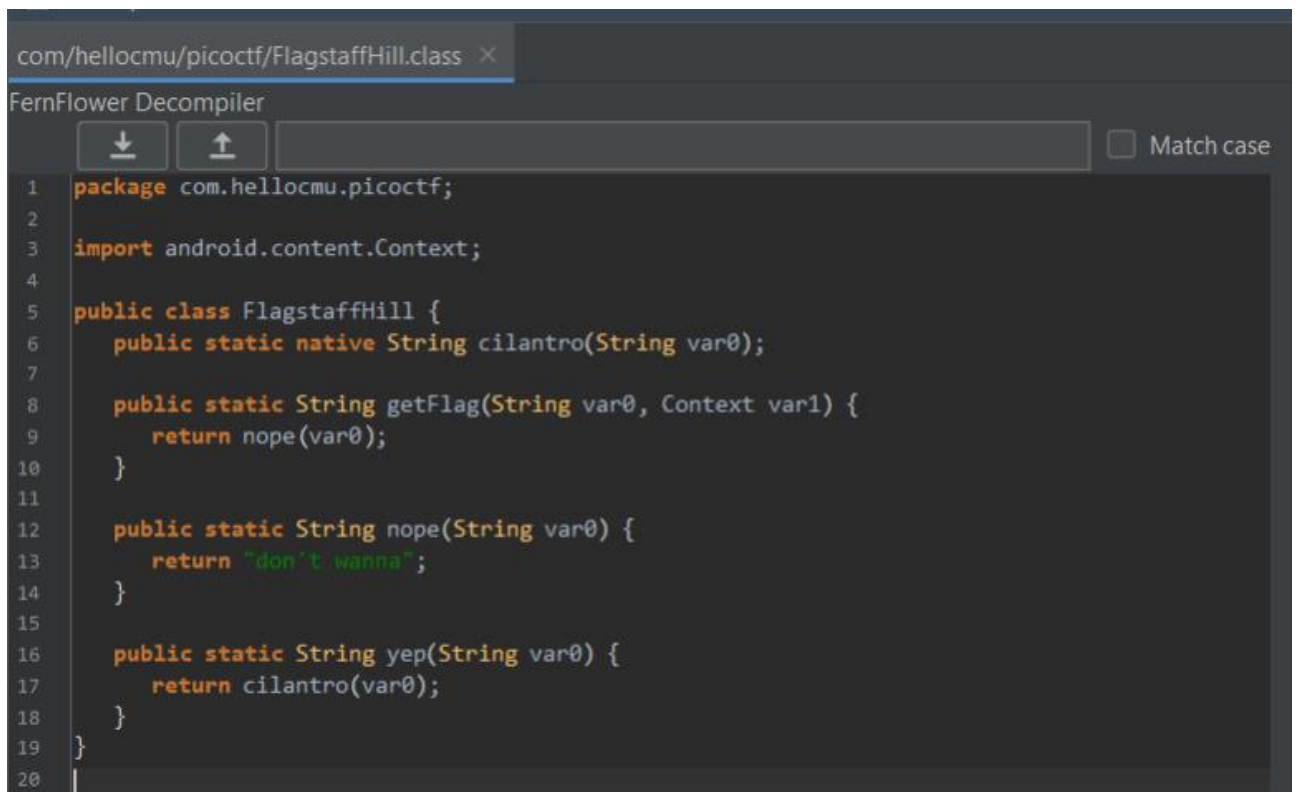
```
C:\platform-tools>adb install four.apk
Performing Streamed Install
Success
C:\platform-tools>
```



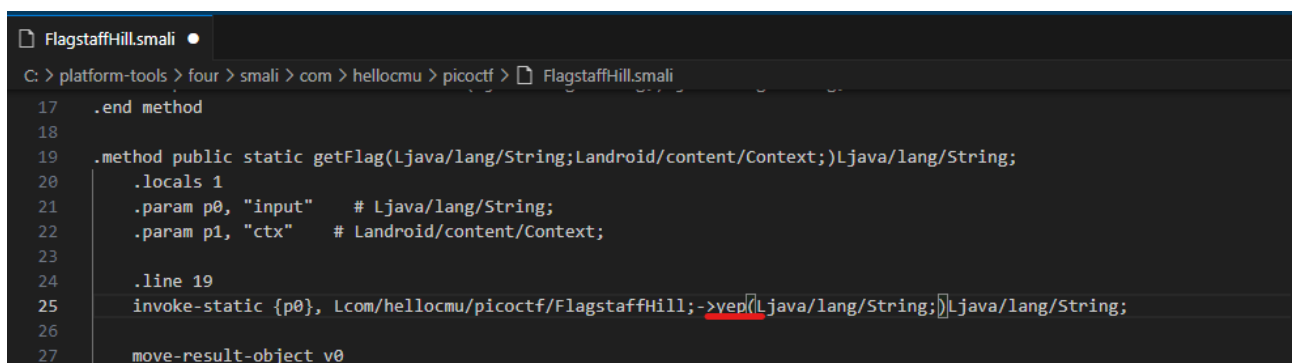
Decompile file apk

```
C:\platform-tools>apktool d four.apk
I: Using Apktool 2.10.0 on four.apk with 8 thread(s).
I: Baksmaling classes.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\hovik\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
C:\platform-tools>
```

Đọc code ta có thể thấy hàm nope chỉ trả về “don’t wanna” nên ta có thể đổi nó thành về hàm yep và compile



```
com/helloemu/picoctf/FlagstaffHill.class
FernFlower Decompiler
package com.helloemu.picoctf;
import android.content.Context;
public class FlagstaffHill {
    public static native String cilantro(String var0);
    public static String getFlag(String var0, Context var1) {
        return nope(var0);
    }
    public static String nope(String var0) {
        return "don't wanna";
    }
    public static String yep(String var0) {
        return cilantro(var0);
    }
}
```



```
FlagstaffHill.smali
C:\> platform-tools > four > smali > com > helloemu > picoctf > FlagstaffHill.smali
17 .end method
18
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/helloemu/picoctf/FlagstaffHill;.>yep(Ljava/lang/String;)Ljava/lang/String;
26
27     move-result-object v0
```

Thực hiện compile lại file

```
C:\platform-tools>apktool b four -o four2.apk
I: Using Apktool 2.10.0 with 8 thread(s).
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: four2.apk

C:\platform-tools>
```

Tạo chữ ký

```
C:\platform-tools>keytool -genkey -v -keystore four2.keystore -alias four2 -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces.
What is your first and last name?
What is the name of your organizational unit?
What is the name of your organization?
What is the name of your City or Locality?
What is the name of your State or Province?
```

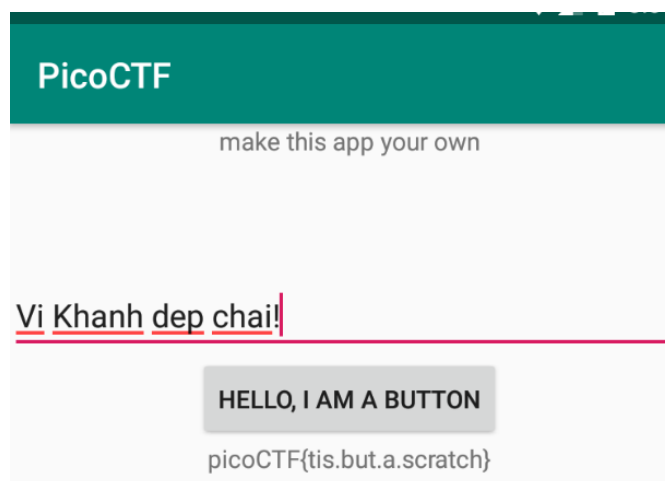
Thực hiện ký file apk để install vào ứng dụng

```
C:\platform-tools>C:\Users\hovik\AppData\Local\Android\Sdk\build-tools\35.0.0\apksigner sign --ks four2.keystore four2.apk
Keystore password for signer #1:
C:\platform-tools>
```

Install apk mới được sửa

```
C:\platform-tools>adb install four2.apk
Performing Streamed Install
Success

C:\platform-tools>
```

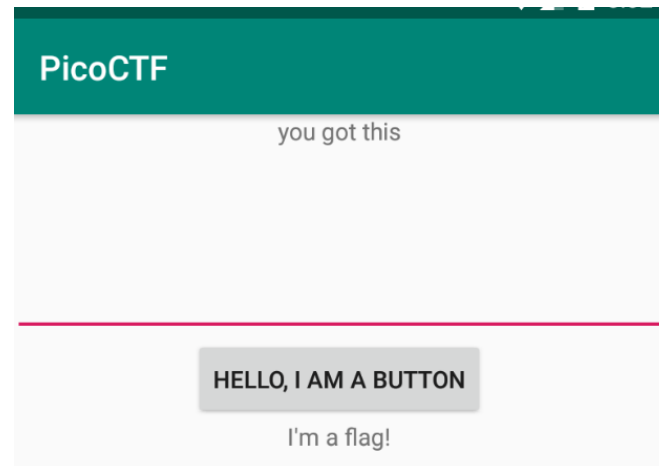


Flag: picoCTF{tis.but.a.scratch}

5. Five.apk

Cài đặt five.apk

```
C:\platform-tools>adb install five.apk
Performing Streamed Install
Success
```



Sau khi decompile và mở file FlagstaffHill thấy chuỗi password được tạo ra bằng cách nối các giá trị.

```
FernFlowDecompiler
1 package com.hellocmu.picoctf;
2
3 import android.content.Context;
4
5 public class FlagstaffHill {
6     public static native String cardamom(String var0);
7
8     public static String getFlag(String var0, Context var1) {
9         StringBuilder var4 = new StringBuilder( "aa" );
10        StringBuilder var5 = new StringBuilder( "aa" );
11        StringBuilder var2 = new StringBuilder( "aa" );
12        StringBuilder var3 = new StringBuilder( "aa" );
13        var4.setCharAt(0, (char)(var4.charAt(0) + 4));
14        var4.setCharAt(1, (char)(var4.charAt(1) + 19));
15        var4.setCharAt(2, (char)(var4.charAt(2) + 18));
16        var5.setCharAt(0, (char)(var5.charAt(0) + 7));
17        var5.setCharAt(1, (char)(var5.charAt(1) + 0));
18        var5.setCharAt(2, (char)(var5.charAt(2) + 1));
19        var2.setCharAt(0, (char)(var2.charAt(0) + 0));
20        var2.setCharAt(1, (char)(var2.charAt(1) + 11));
21        var2.setCharAt(2, (char)(var2.charAt(2) + 15));
22        var3.setCharAt(0, (char)(var3.charAt(0) + 14));
23        var3.setCharAt(1, (char)(var3.charAt(1) + 20));
24        var3.setCharAt(2, (char)(var3.charAt(2) + 15));
25        return var0.equals( ".concat(var2.toString()).concat(var5.toString()).concat(var4.toString()).concat(var3.toString())" ) ? "call it" : "NOPE";
26    }
27 }
28
```

Viết lại hàm và run với compiler của Java

```

1 class Main {
2     public static void main(String[] args) {
3         StringBuilder ace = new StringBuilder("aaa");
4         StringBuilder jack = new StringBuilder("aaa");
5         StringBuilder queen = new StringBuilder("aaa");
6         StringBuilder king = new StringBuilder("aaa");
7
8         ace.setCharAt(0, (char) (ace.charAt(0) + 4));
9         ace.setCharAt(1, (char) (ace.charAt(1) + 19));
10        ace.setCharAt(2, (char) (ace.charAt(2) + 18));
11
12        jack.setCharAt(0, (char) (jack.charAt(0) + 7));
13        jack.setCharAt(1, (char) (jack.charAt(1) + 0));
14        jack.setCharAt(2, (char) (jack.charAt(2) + 1));
15
16        queen.setCharAt(0, (char) (queen.charAt(0) + 0));
17        queen.setCharAt(1, (char) (queen.charAt(1) + 11));
18        queen.setCharAt(2, (char) (queen.charAt(2) + 15));
19
20        king.setCharAt(0, (char) (king.charAt(0) + 14));
21        king.setCharAt(1, (char) (king.charAt(1) + 20));
22        king.setCharAt(2, (char) (king.charAt(2) + 15));
23
24        String password = queen.toString()
25            .concat(jack.toString())
26            .concat(ace.toString())
27            .concat(king.toString());
28
29        System.out.println(password);
30    }
31 }
32

```

Output: alphabetsoup

Decompile file apk

```

C:\platform-tools>apktool d five.apk
I: Using Apktool 2.10.0 on five.apk with 8 thread(s).
I: Baksmaling classes.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\hovik\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

```

Invoke hàm cardamom để cho nó gán kết quả và v0, trả về v0 để trả về flag thay vì “call it”.

```

move-result-object v4

.line 36
.local v4, "password":Ljava/lang/String;
invoke-virtual {p0, v4}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

move-result v5

if-eqz v5, :cond_0

const-string v5, "call it"

return-object v5

.line 37
cond_0
const-string v5, "NOPE"

return-object v5
end method

```

```
.line 36
.local v4, "password":Ljava/lang/String;
invoke-virtual {p0, v4}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

move-result v5

invoke-static {p0}, Lcom/hellocmu/picoctf/FlagstaffHill;->cardmom (Ljava/lang/String;) Ljava/lang/String;
move-result-object v0
return-object v0

.line 37
:cond_0
const-string v5, "NOPE"
```

Thực hiện build ký và cài đặt lại ứng dụng

```
C:\platform-tools>apktool b five -o five2.apk
I: Using Apktool 2.10.0 with 8 thread(s).
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: five2.apk
```

Tạo và ký file apk

```
C:\platform-tools>keytool -genkey -v -keystore five2.keystore -alias five2 -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces
.
What is your first and last name?
What is the name of your organizational unit?
What is the name of your organization?
What is the name of your City or Locality?
What is the name of your State or Province?
What is the two-letter country code for this unit?
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
```

```
C:\platform-tools>C:\Users\hovik\AppData\Local\Android\Sdk\build-tools\35.0.0\apksigner sign --ks five2.keystore five2.apk
Keystore password for signer #1:
C:\platform-tools>
```

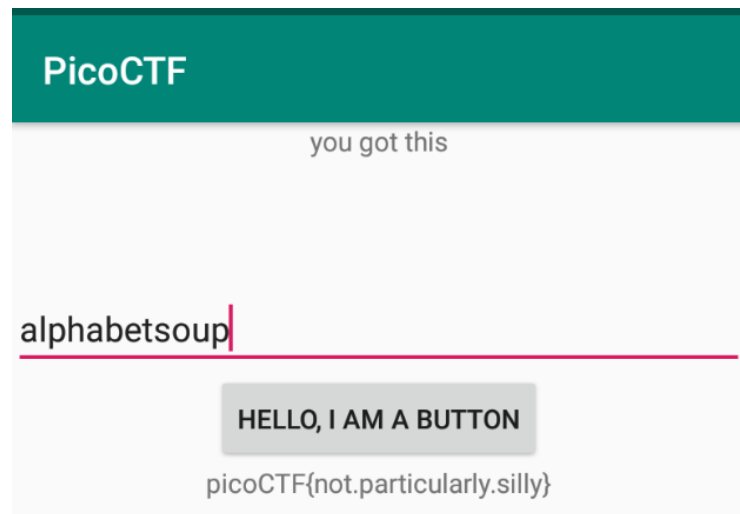
Cài lại

```
C:\platform-tools>adb uninstall com.hellocmu.picoctf
Success

C:\platform-tools>adb install five2.apk
Performing Streamed Install
Success

C:\platform-tools>
```

Nhập lại “alphabetsoup” vào ứng dụng và nhận được flag



Flag: picoCTF{not.particularly.silly}