

BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Lab 5: Mobile Forensic

GVHD: Đoàn Minh Trung

Nhóm 13

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.021.ATCL

STT	Họ và tên	MSSV	Email
1	Ngô Minh Quân	21522492	21522492@gm.uit.edu.vn
2	Phùng Đức Lương	21522312	21522312@gm.uit.edu.vn
3	Hồ Ngọc Thiện	21522620	21522620@gm.uit.edu.vn
4	Chu Nguyễn Hoàng Phương	21522483	21522483@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 1	100%
2	Kịch bản 2	100%
3	Kịch bản 3	100%
4	Kịch bản 4	50%

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

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

Mục Lục

A. KỊCH BẢN 01	3
B. KỊCH BẢN 02	8
C. KỊCH BẢN 03	11
D. KỊCH BẢN 04	16
YÊU CẦU CHUNG	27

BÁO CÁO CHI TIẾT

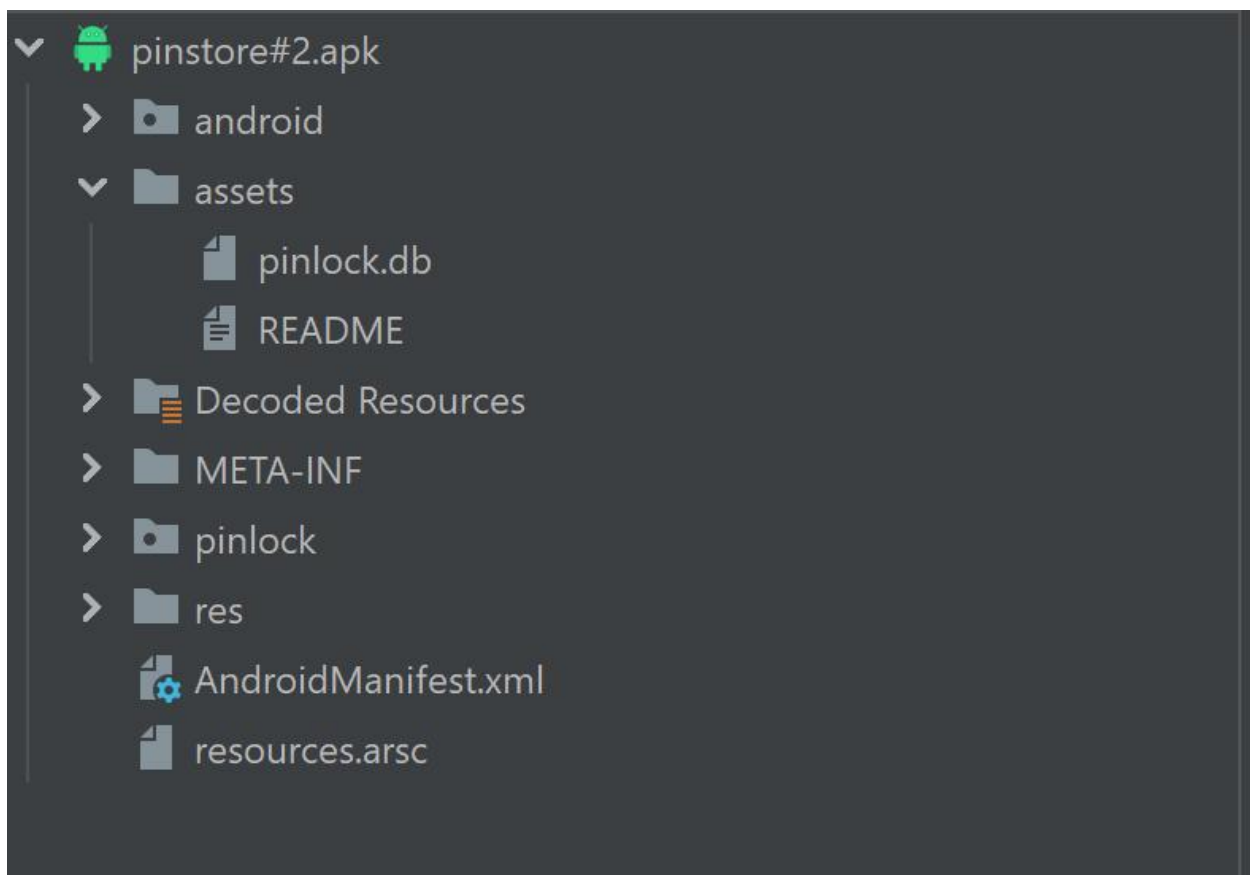
A. KỊCH BẢN 01

1. Kịch bản 01. Thực hiện phân tích ứng dụng Android

- Mô tả: Phân tích ứng dụng Android, tìm mã PIN trong ứng dụng để tìm flag.
- Tài nguyên thực hiện: pinstore.zip
- Yêu cầu – Gợi ý: Sử dụng các công cụ dịch ngược (decompile) trên mã nguồn Android để phân tích.

Đáp án:

-Sử dụng công cụ BytecodeViewer ta thấy được file pinlock.db



Truy cập file Db để tìm mã pin

```
kali@kali: ~/Downloads/resources-session05/pinstore/assets
File Actions Edit View Help
(kali@kali)-[~/Downloads/resources-session05/pinstore/assets]
$ sqlite3 pinlock.db
SQLite version 3.44.2 2023-11-24 11:41:44
Enter ".help" for usage hints.
sqlite> .databases
main: /home/kali/Downloads/resources-session05/pinstore/assets/pinlock.db r/w
sqlite> .table
android_metadata  pinDB          secretsDBv1        secretsDBv2
sqlite> select * from android_metadata
...>
en_US
sqlite> select * from pinDB
...>
1d8531a519b3d4dfebece0259f90b466a23efc57b
sqlite> select * from secretsDBv1
Parse error: no such table: secretsDBv1
sqlite> select * from secretsDBv1
...>
1hcsvUnln5jMdw3GeI4o/txB5vaEf1PFAnKQ3kPsRW2o5rR0a1JE54d0BLkzXPtqB
sqlite> select * from secretsDBv2
...>
1Bi528nDlNBcX9BcCC+ZqGQo10z01+GOWSmvxRj7jg1g=
sqlite>
```

Ta được những mã hash tuy nhiên chỉ có hash 2 có thể crack

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

d8531a519b3d4dfebece0259f90b466a23efc57b

Tôi không phải là người máy

reCAPTCHA

Bảo mật - Giữ Khẩn

Crack Hashes

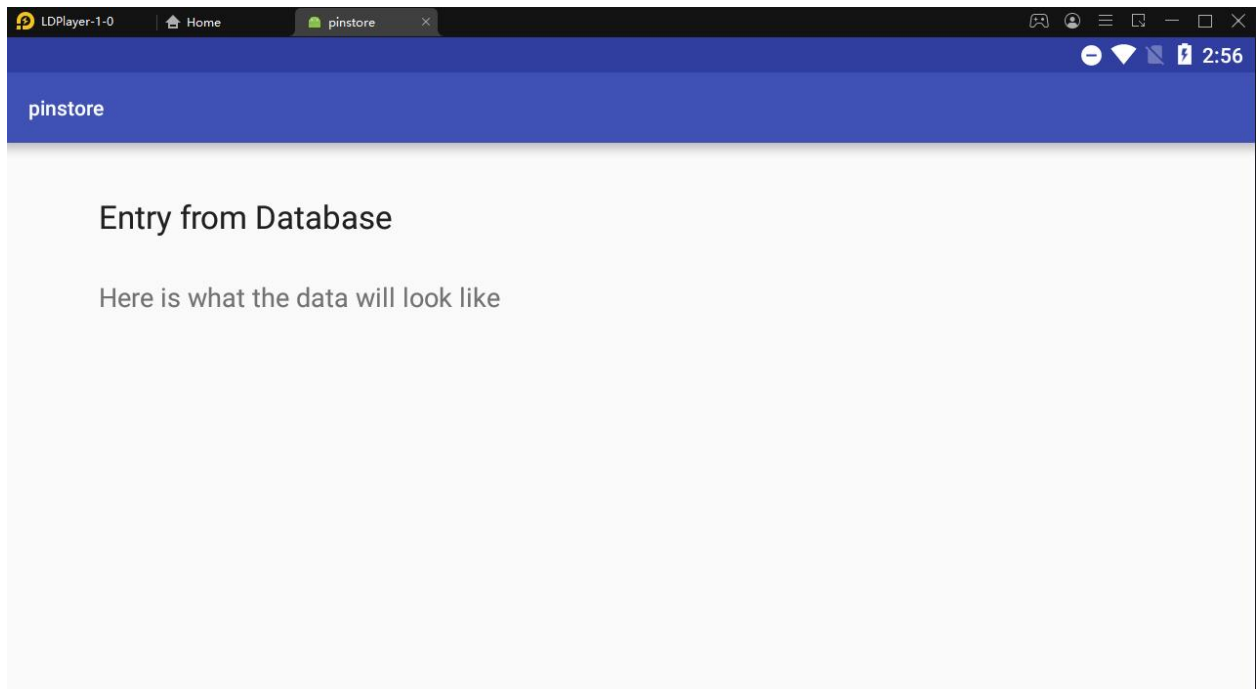
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, sha512_224, sha512_224_160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
d8531a519b3d4dfebece0259f90b466a23efc57b	sha1	7498

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

Tiến hành nhập thử ta có



Đây là kết quả hiển thị v1

Để có thể hiển thị v2 ta cần sửa 1 số đoạn code

Vào SecretDisplay.smali

Ta sửa const-string v7, "v1" thành const-string v7, "v2"

Đồng thời vào DataUtilities.smali sửa
const-string v1, "SELECT entry FROM secretsDBv1" thành
const-string v1, "SELECT entry FROM secretsDBv2"

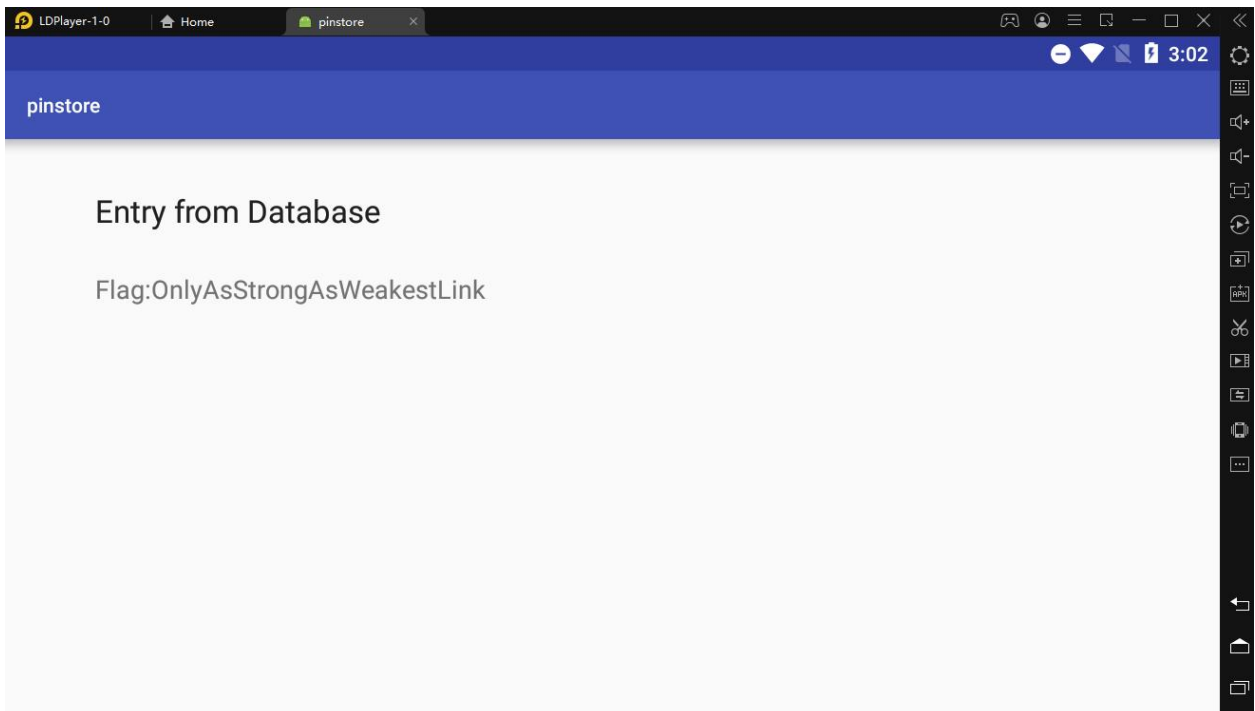
Tiến hành lưu file chỉnh sửa thành file apk

```
(kali@kali)-[~/Downloads/resources-session05]
$ apktool b pin pin1.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: Building resources...
W: aapt: brut.common.BrutException: brut.common.BrutException: Could not extract resource: /prebuilt/linux/aapt_64 (defaulting to $PATH bin
ary)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: pin/dist/pinstore.apk
```

Tạo keystore và kí để có thể sử dụng ứng dụng

```
kali@kali: ~/Downloads/resources-session05/pin/dist
File Actions Edit View Help
(kali@kali)-[~/Downloads/resources-session05/pin/dist]
$ keytool -genkey -v -keystore my.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias app
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: 1
What is the name of your organizational unit?
[Unknown]: 1
What is the name of your organization?
[Unknown]: 1
What is the name of your City or Locality?
[Unknown]: 1
What is the name of your State or Province?
[Unknown]: 1
What is the two-letter country code for this unit?
[Unknown]: 1
Is CN=1, OU=1, O=1, L=1, ST=1, C=1 correct?
[no]: yes
Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=1, OU=1, O=1, L=1, ST=1, C=1
[Storing my.keystore]
(kali@kali)-[~/Downloads/resources-session05/pin/dist]
$ apksigner sign --ks my.keystore pinstore.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:
(kali@kali)-[~/Downloads/resources-session05/pin/dist]
$
```

Ta thu được kết quả

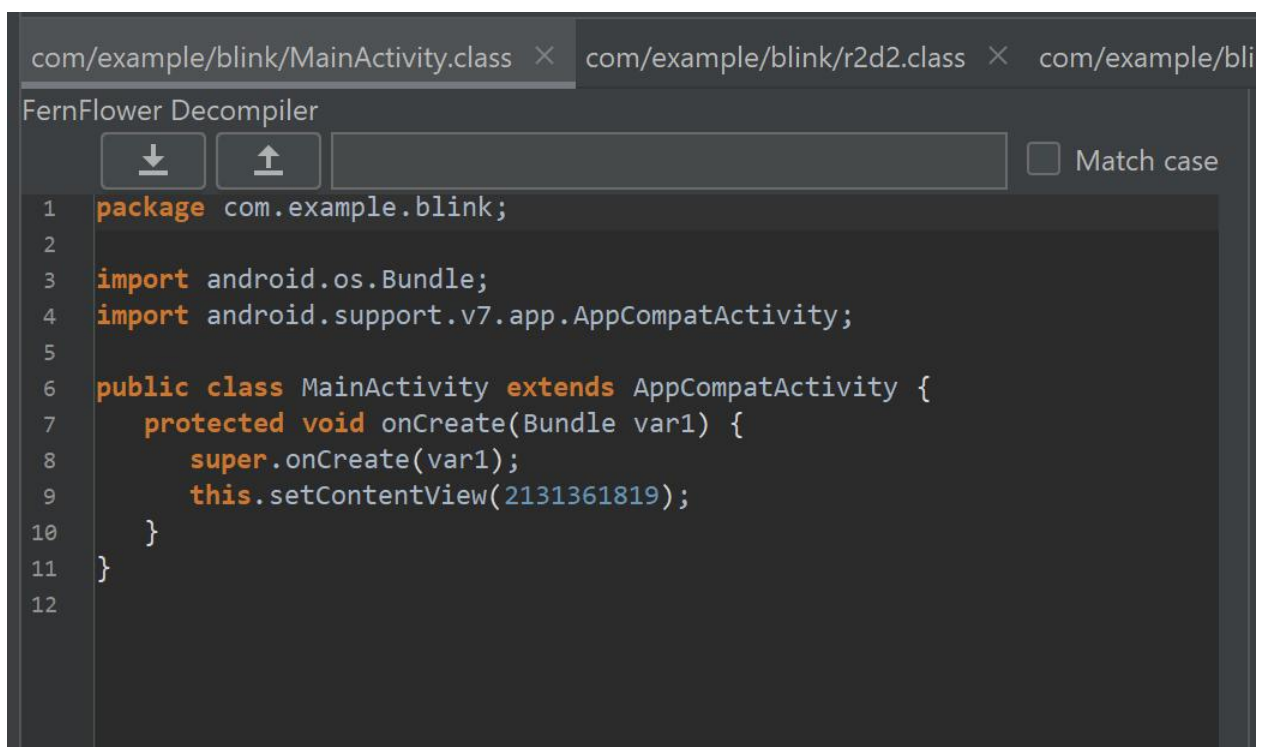


B. KỊCH BẢN 02

2. Kịch bản 02. Thực hiện phân tích tập tin ứng dụng thu được.

- Mô tả: Ứng dụng kb02 cần được phân tích thành mã smali để tìm flag.
- Tài nguyên thực hiện: kb02_zha.apk
- Yêu cầu – Gợi ý: sử dụng công cụ APKTool/ JADX/ dex2jar/ jdgui/ Android Studio, flag có dạng CTF{....}

Đáp án:



```
com/example/blink/MainActivity.class × com/example/blink/r2d2.class × com/example/blink/...
FernFlower Decompiler
[Download] [Upload] [Search] [Match case]
1 package com.example.blink;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class MainActivity extends AppCompatActivity {
7     protected void onCreate(Bundle var1) {
8         super.onCreate(var1);
9         this setContentView(2131361819);
10    }
11 }
12
```

Đọc file MainActivity.java, ta thấy không có gì để hiển thị, ta tìm kiếm các Activity khác để tìm kiếm flag. Cụ thể, ta tìm thấy điểm đáng nghi ở file r2d2.java


```
1 package com.example.blink;
2
3 import android.graphics.BitmapFactory;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.util.Base64;
7 import android.widget.ImageView;
8
9 public class r2d2 extends AppCompatActivity {
10     protected void onCreate(Bundle var1) {
11         super.onCreate(var1);
12         this setContentView(2131361820);
13         ImageView var3 = (ImageView) this.findViewById(2131230804);
14         byte[] var2 = Base64.decode("data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAQ/2wBDAAMCAgICAgMCAgIDAwMDBAYEBAQEBAgGBg");
15         var3.setImageBitmap(BitmapFactory.decodeByteArray(var2, 0, var2.length));
16     }
17 }
18
```

Một hình ảnh được tạo ra bởi 1 đoạn mã base64 được decode về mảng byte, sau đó sử dụng phương thức `setImageBitmap` để hiển thị hình ảnh.

Ta có thể sử dụng lệnh:

`adb shell am start -n com.example.blink/.r2d2 com.example.blink/.r2d2`

để hiển thị Activity này.

```
PS C:\Users\Ngoc Thien\Desktop\resources-session05> adb shell am start -n com.example.blink/.r2d2
Starting: Intent { cmp=com.example.blink/.r2d2 }
```



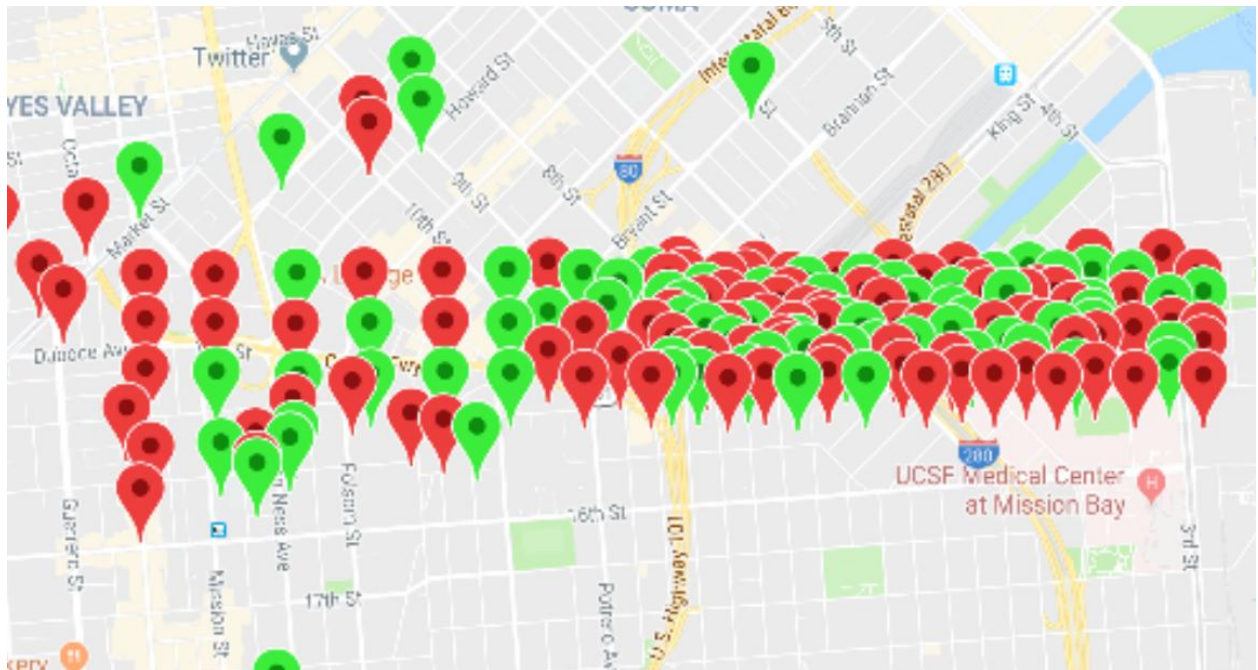
C. KỊCH BẢN 03

3. Kịch bản 03. Thực hiện phân tích tập tin ứng dụng thu được.

- Mô tả: Một ứng dụng có tính năng ghi nhớ các địa điểm mà người dùng muốn hay không muốn tham quan chỉ bằng dấu tick đơn giản trên bản đồ. Tìm flag.
- Tài nguyên: kb03_yon.apk
- Yêu cầu – Gợi ý: Decompile, chú ý CSDL của ứng dụng.

Gợi ý:

Khi mở app ta có thể thấy rất nhiều điểm lộn xộn



Tiến hành phân tích file bằng apktool

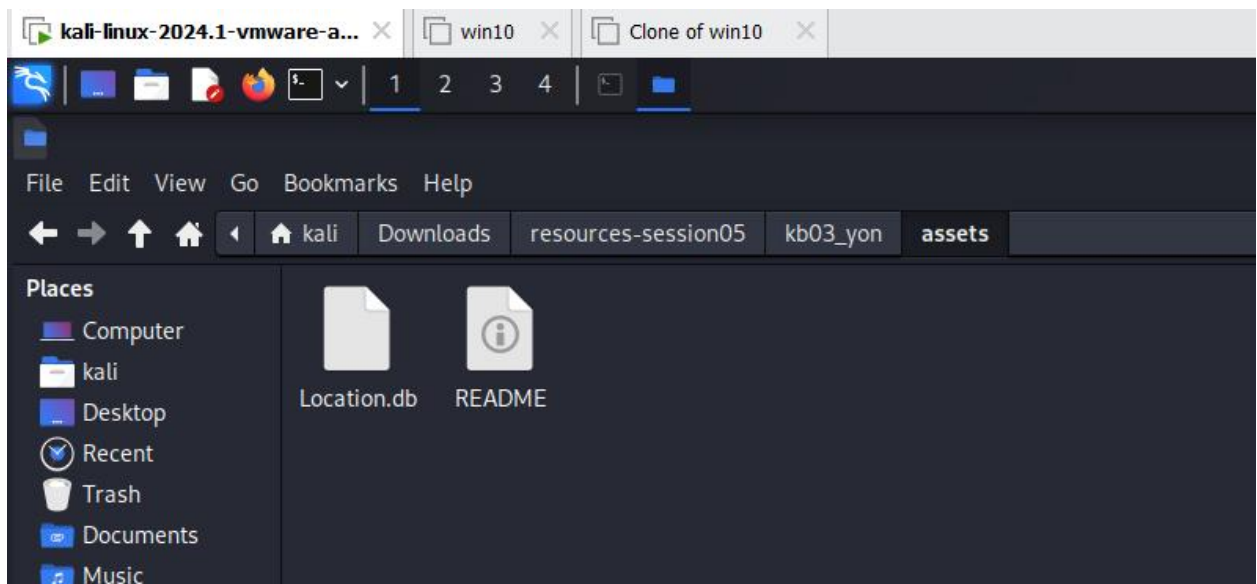
```

(kali@kali)-[~/Downloads/resources-session05]
$ apktool d kb03_yon.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.7.0-dirty on kb03_yon.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /home/kali/.local/share/apktool/framework/1.apk
I: Regular manifest package ...
I: Decoding file-resources ...
I: Decoding values */* XMLs ...
I: Baksmaling classes.dex ...
I: Baksmaling classes2.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...

(kali@kali)-[~/Downloads/resources-session05]
$

```

Ta tìm thấy 1 file Location.db



Ta truy cập vào cơ sở dữ liệu Location.db

Xem dữ liệu ngày 02/08/2019

```

sqlite> select * from locations where date = '02/08/2019';
02/08/2019|37.7703669470161|-122.421883132294|0.0

```

02/08/2019|37.7692135433168|-122.421840216949|0.0
02/08/2019|37.7679922727318|-122.421840216949|0.0
02/08/2019|37.7703330236347|-122.419651534393|0.0
02/08/2019|37.7691456954801|-122.419651534393|0.0
02/08/2019|37.7679244237746|-122.419608619049|120.0
02/08/2019|37.7703669470161|-122.417076613739|120.0
02/08/2019|37.770400870382|-122.414973761871|0.0
02/08/2019|37.7691117715384|-122.417119529083|0.0
02/08/2019|37.7691456954801|-122.414802100494|120.0
02/08/2019|37.7678226502221|-122.417033698395|120.0
02/08/2019|37.7678565747551|-122.41475918515|120.0
02/08/2019|37.7704347937323|-122.41252758725|0.0
02/08/2019|37.7691796194062|-122.412441756561|0.0
02/08/2019|37.7679244237746|-122.412441756561|120.0
02/08/2019|37.7704347937323|-122.410467650726|120.0
02/08/2019|37.7691456954801|-122.410424735382|120.0
02/08/2019|37.7678904992727|-122.410381820038|120.0
02/08/2019|37.7703669470161|-122.408107306793|120.0
02/08/2019|37.770400870382|-122.406004454926|120.0
02/08/2019|37.7690778475811|-122.408064391449|0.0
02/08/2019|37.7691117715384|-122.405961539581|0.0
02/08/2019|37.7678226502221|-122.408064391449|0.0
02/08/2019|37.7678226502221|-122.405961539581|0.0
02/08/2019|37.7702991002377|-122.403687026337|0.0
02/08/2019|37.7690439236083|-122.403601195648|0.0
02/08/2019|37.7677548011092|-122.403558280304|0.0
02/08/2019|37.7702991002377|-122.401584174469|0.0
02/08/2019|37.7690439236083|-122.401498343781|0.0
02/08/2019|37.7677548011092|-122.401369597748|120.0
02/08/2019|37.7702651768251|-122.399309661224|120.0
02/08/2019|37.7690439236083|-122.399352576569|120.0
02/08/2019|37.7678226502221|-122.399223830536|120.0


```

02/08/2019|37.7702651768251|-122.397378470734|0.0
02/08/2019|37.76900999962|-122.397378470734|0.0
02/08/2019|37.7678226502221|-122.397335555389|0.0
02/08/2019|37.7702991002377|-122.395189788178|120.0
02/08/2019|37.7703330236347|-122.393215682343|120.0
02/08/2019|37.7690439236083|-122.393172766998|120.0
02/08/2019|37.7690778475811|-122.395189788178|0.0
02/08/2019|37.7678565747551|-122.395189788178|0.0
02/08/2019|37.7677887256734|-122.393215682343|0.0
02/08/2019|37.7703330236347|-122.390898253754|120.0
02/08/2019|37.7703669470161|-122.388795401886|0.0
02/08/2019|37.76900999962|-122.390812423065|0.0
02/08/2019|37.7690439236083|-122.388623740509|0.0
02/08/2019|37.7678226502221|-122.390769507721|0.0
02/08/2019|37.7678226502221|-122.388623740509|0.0

```

Ta chỉ lấy dữ liệu ngày 02/08/2019 bởi vì đây là ngày chứa thông điệp và xóa tất cả dữ liệu khác

Tiến hành tạo key

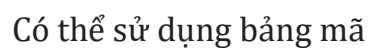
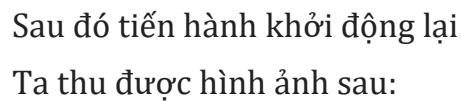
```

(kali@kali)-[~/Downloads/resources-session05/kb03_yon/assets]
$ keytool -genkey -v -keystore my.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias app
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: 0
What is the name of your organizational unit?
[Unknown]: 0
What is the name of your organization?
[Unknown]: 0
What is the name of your City or Locality?
[Unknown]: 0
What is the name of your State or Province?
[Unknown]: 0
What is the two-letter country code for this unit?
[Unknown]: 0
Is CN=0, OU=0, O=0, L=0, ST=0, C=0 correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=0, OU=0, O=0, L=0, ST=0, C=0
[Storing my.keystore]

```

Tạo file apk



BRAILLE Alphabet

A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	
.	,	?	!	'	-	CAPITAL	#	0
1	2	3	4	5	6	7	8	9

Ta có thể giải được FLAG là: **Z3LDA**

D. KỊCH BẢN 04

4. Kịch bản 04. Điều tra trên tập tin ứng dụng thu được.

- Mô tả: Một ứng dụng thời tiết đơn giản có tính năng thu thập và hiển thị thông tin thời tiết.
- Tài nguyên: kb04_tianqi.apk
- Yêu cầu – Gợi ý: Xác định phiên bản Android đang chạy của ứng dụng. Sử dụng một số công cụ decompile apk như Jadx để phân tích code ứng dụng. Flag có định dạng CTF{...}

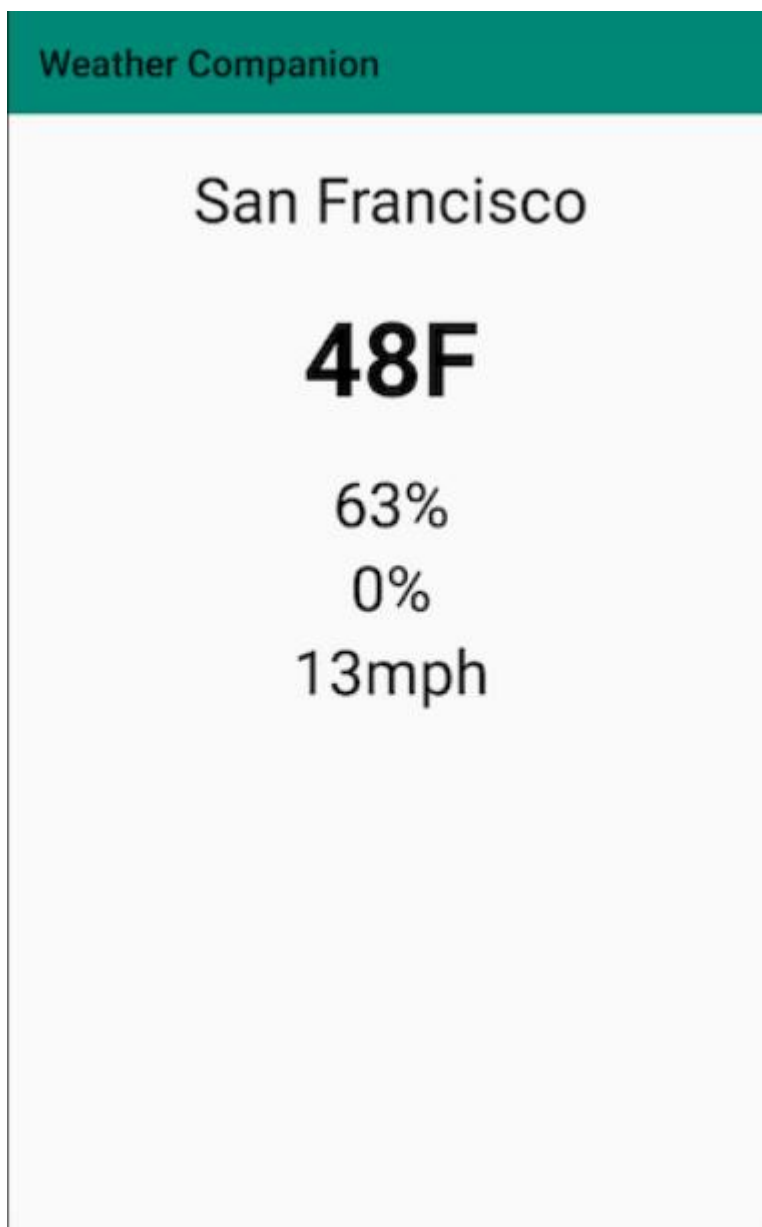
Đáp án:

Đây là kịch bản em tham khảo được từ

<https://aadityapurani.com/2019/03/07/bsidessf-ctf-2019-mobile-track/#weather>.

Hình ảnh trích xuất ra được flag và các bước làm chỉ là minh họa cho ý tưởng, bởi vì không có tài khoản truy cập Google Cloud Storage của challenge CTF này.

Khi mở file lên thì ta chỉ thấy như thế này



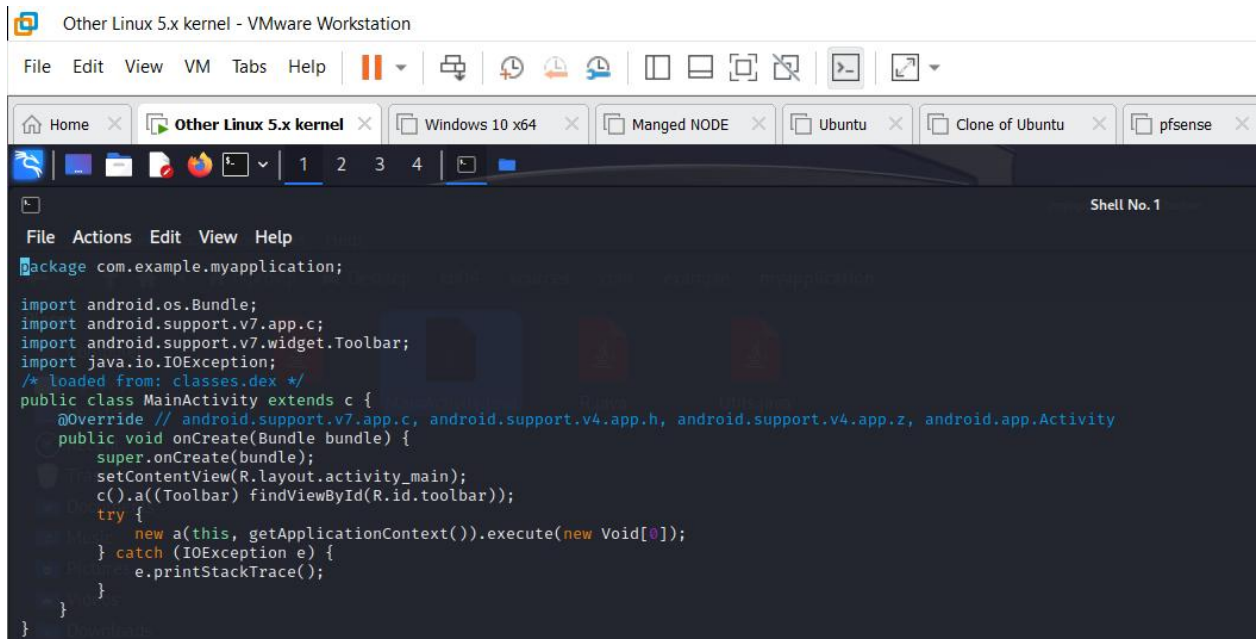
Sau khi decompile thì ta thấy phiên bản sdk mà hệ thống tương thích là 26 hoặc 27 nghĩa là android 8 và android 8.1

A screenshot of a web browser window displaying the content of an AndroidManifest.xml file. The browser's address bar shows the file path: file:///home/niprovip/Desktop/kb04/resources/AndroidManifest.xml. The page content shows the XML code for the application manifest, including the package name, SDK versions, permissions, and the main activity. The code is as follows:

```
<?xml version="1.0" encoding="utf-8" android:splitScreen="true" android:allowBackup="true" android:usesCleartextTraffic="true">
<manifest android:versionCode="1" android:versionName="1.0" package="com.example.myapplication" platformBuildVersionCode="1" platformBuildVersionName="1">
  <uses-sdk android:minSdkVersion="26" android:targetSdkVersion="27"/>
  <uses-permission android:name="android.permission.INTERNET"/>
  <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:allowBackup="true" android:supportRtl="true" android:roundIcon="@mipmap/ic_launcher_round">
    <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/app_name" android:name="com.example.myapplication.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

- Chương trình có activity là com.example.myapplication.MainActivity, đọc sơ qua thì đoạn mã cố gắng thực thi 1 tác vụ bằng cách sử dụng lớp "a", truyền hoạt động "this"

và ngữ cảnh “getApplicationContext()” làm đối số. Lớp này có thể là 1 lớp thực hiện tùy chỉnh của 1 tác vụ bất đồng bộ AsyncTask để thực hiện các hoạt động nền. Sau đó dòng execute để khởi động thực thi tác vụ.



- Từ đó ta xem code của a.java

Đoạn mã của code a.java có chức năng lấy dữ liệu thời tiết và hiển thị nó trên màn hình. Bao gồm một số thành phần như sau:

- **Lớp** a được kế thừa từ AsyncTask, cho phép thực hiện các tác vụ nền mà không làm gián đoạn giao diện người dùng. Với :
- **Phương thức** doInBackground được thực thi trong nền và truy xuất dữ liệu thời tiết. Nó sử dụng lớp Utils để thực hiện các chức năng hỗ trợ khác nhau như thao tác chuỗi và giải mã. Tiếp đó xây dựng URL bằng thông tin từ đối tượng gVar và lớp c rồi thực hiện yêu cầu HTTP GET đến URL được xây dựng. Cuối cùng là đọc phản hồi từ máy chủ và chuyển đổi nó thành chuỗi.
- **Phương thức** onPostExecute sẽ được gọi sau khi doInBackground kết thúc và nhận dữ liệu thời tiết dưới dạng chuỗi. Sau đó, nó phân tích cú pháp phản hồi JSON bằng JSONObject rồi trích xuất chi tiết thời tiết như thành phố, nhiệt độ, lượng mưa, độ ẩm và gió. Cuối cùng, nó cập nhật các phần tử TextView tương ứng trong bố cục MainActivity (ví dụ: cityName, temperatureValue) bằng cách đặt văn bản của chúng thành các giá trị được trích xuất.


```

22 public final class a extends AsyncTaskVoid, Void, String {
23
24     /* renamed from: a; reason: collision with root package name */
25     private final MainActivity f867a;
26     private final Context b;
27
28     public a(MainActivity mainActivity, Context context) {
29         this.f867a = mainActivity;
30         this.b = context;
31     }
32
33     /* JADX WARN: Type inference failed for: r2v7, types: [ServiceT extends com.b.c.f.OptionsT, com.b.c.f] */
34     private String a() {
35         String str = "";
36         k j = k.j();
37         if (j.h == 0) {
38             j.h = j.g.a(j);
39         }
40         g gvar = (g) j.h;
41         String str2 = null;
42         try {
43             Utils utils = new Utils(this.b);
44             String str3 =
45             Utils.a(("L5rL1S1C8W6T1IR0u1QVFIETfF50tK1J50Z0U1C0UR8T3Ma3Fo21H0Kcw6FRUzBQVND0tZ2ddnU21B20VBQ9QKfRQ2J0YUy3FhL1yI2QKVCbXIReV1Y5J0d2pa0G236KcWfPb2L1Y19p5NqFY0tR12-ba12MVRy5kHCN1x520Z2MvovYdE3N0dTBQpuzD1E1GcWnsd3
46             // + Utils.a(utils.dct()) + new String(utils.ssf("Cm0h3716Tz/bjC3kx4v/NH/FH0310h0af1j5377XKTCkAcchV1b0y10p11-6d0VhK1Gee83Ine5u8RtL85001Ygk0u0510P3J71Y0P7YcADKVL8hK4W1-PF1D2VQ11h0u0h4BP5p0f-AD0K1D717KcF2E1V1j75cz/
47             // 00h0x1IMang3Mh1+00E2Kw/325p0w1q1mT3d3V3yU1/fH01h3J0k5810dyW699eh/0f60U7PcTq1Y8aF0d47h0af7B0Q/Mj/F0u18q0tHTH0aPHe1nV7na1X7W1d2FYur1uZ0wH1294zT282B)X015K5fJpTEtUz/3kXt1Nt0W3gTJ6awj+eNpEh5MhNnc556V5u0AF01, 390)) +
48             Utils.a(("MSVXSROXNBVCZ2N0BUCW3Y052fAK3101YU2VS0RSV0HLR1NE2Y5H8ZRE06KLJ35WNLPEX0Y5K1N0R9GOLWfTU0531LFGVChAKJ3CV0T7J5TKNC3MREU45P1NEG4Q2H225QhM50N3K0M2KPFfWQhK6J0TANKD3FHUERRK3ZU5R2VXB5FEQ5U0VAV6R2B13CK0AK1ZHEUT1R5D5RQDT
49             ));
50             URL a2 = gvar.a(c.a("weather-companion", "weather.json").a(), TimeUnit.DAYS, g.a.a(i.a(new ByteArrayInputStream(((((((((((typeV: \\" + utils.f866a.getResources().getString(R.string.type) + "\\",) + "\project_id\":
51             // + Utils.a(utils.b) + "\",) + "\private_key_id\": \\" + Utils.a() + "\",) + "\private_key\": \\" + str3.replace("\\", "\\\") + "\",) + "\client_email\": \\" + "weather-companion-service-acco@b5ides-sf-
52             // ctf-2019-lam.gserviceaccount.com\",) + "\client_id\": \\" + BigInteger.valueOf(utils.gcl()).multiply(BigInteger.valueOf(19L)).multiply(BigInteger.valueOf(105363817965794407L)).toString() + "\\",) + "\auth_url\": \\" + new
53             // String(utils.ssf("uggcfr//bmhg02.t0blymcfv.p0z/g0xra", 35)) + "\\",) + "\token_url\": \\" + new String(utils.ssf("uggcfr//bmhg02.t0blymcfv.p0z/g0xra", 35)) + "\\",) + "\auth_provider_x509_cert_url\": \\" +
54             // Utils.a("ah0c8M0L9Yd3dcu29vZ2x1Y8Pcy5Jb20vc91b3QdJ5vBwVYWRhdG0veD0593ZWf0a0yV1MwYx8hbm1vb11ZC32aWNL1WfJY281NDB1c21kZX0tC2Y1Y3RmLTlWmTKu0WF1LmdzZXJ2aWNL1WmJ3b3VudC5jb20=\", 0) + "\\",).getBytes()))));
55             str2 = a2.toString();
56             HttpURLConnection httpURLConnection = (HttpURLConnection) a2.openConnection();
57             httpURLConnection.connect();
58             httpURLConnection.getContentLength();
59             BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(httpURLConnection.getInputStream()));
60             StringReader stringReader = new StringReader();
61             while (true) {
62                 String readLine = bufferedReader.readLine();
63
64                 if (readLine == null) {
65                     break;
66                 }
67                 stringBuffer.append(readLine + "\n");
68             }
69             str = stringBuffer.toString();
70         } catch (IOException e) {
71             e.printStackTrace();
72         }
73         if (str2 != null) {
74             Log.d("Background url", "Background URL in progress");
75         }
76         return str;
77     }
78
79     @Override // android.os.AsyncTask
80     protected final /* synthetic */ String doInBackground(Void[] voidArr) {
81         return a();
82     }
83
84     @Override // android.os.AsyncTask
85     protected final /* synthetic */ void onPostExecute(String str) {
86         String str2 = str;
87         Log.d("Complete repsonse", str2);
88         try {
89             JSONObject jsonObject = new JSONObject(str2);
90             String string = ((JSONObject) jsonObject.get("display_location")).getString("city");
91             JSONObject jsonObject2 = (JSONObject) jsonObject.get("current_weather");
92             String string2 = jsonObject2.getString("temperature");
93             String string3 = jsonObject2.getString("precipitation");
94             String string4 = jsonObject2.getString("humidity");
95             String string5 = jsonObject2.getString("wind");
96             ((TextView) this.f867a.findViewById(R.id.cityName)).setText(string);
97             ((TextView) this.f867a.findViewById(R.id.precipitationValue)).setText(string3);
98             ((TextView) this.f867a.findViewById(R.id.humidityValue)).setText(string4);
99             ((TextView) this.f867a.findViewById(R.id.windValue)).setText(string5);
100             ((TextView) this.f867a.findViewById(R.id.temperatureValue)).setText(string2);
101         } catch (JSONException e) {
102             e.printStackTrace();
103         }
104     }
105 }

```

- Vậy Json ở đây là gì, ta xem trong file Utils.java, có thể thấy rằng chúng đang giải mã Base64, int thành char, BigInteger thành Hex và các chức năng của ba lớp thư viện native.

```

public static String a() {
    return new BigInteger("627096631258101466300448072738386213700396112265").toString(16);
}

public static String a(String str, int i) {
    byte[] bytes;
    if (i == 0) {
        bytes = Base64.getDecoder().decode(str);
    } else {
        org.apache.a.a.a aVar = new org.apache.a.a.a.a();
        bytes = str.getBytes();
        if (bytes != null && bytes.length != 0) {
            b.a aVar2 = new b.a();
            aVar.a(bytes, 0, bytes.length, aVar2);
            aVar.a(bytes, 0, -1, aVar2);
            bytes = new byte[aVar2.d];
            int length = bytes.length;
            if (aVar2.c != null) {
                int min = Math.min(aVar2.c != null ? aVar2.d - aVar2.e : 0, length);
                System.arraycopy(aVar2.c, aVar2.e, bytes, 0, min);
                aVar2.e += min;
                if (aVar2.e >= aVar2.d) {
                    aVar2.c = null;
                }
            }
        }
    }
    return new String(bytes);
}

public static String a(int[] iArr) {
    String str = "";
    for (int i = 0; i < iArr.length; i++) {
        str = str + ((char) iArr[i]);
    }
    return str;
}

```

```

/* JADX INFO: Access modifiers changed from: package-private */
public native byte[] dks();

/* JADX INFO: Access modifiers changed from: package-private */
public native long gci();

/* JADX INFO: Access modifiers changed from: package-private */
public native byte[] ss(String str, int i);

```

- Để rõ hơn chúng ta sẽ hook “toString” và kết xuất giải mã bằng Frida. Quá trình tấn công sẽ bao gồm 3 bước

- 1.) Bypass SSL Unpinning
- 2.) Hook toString
- 3.) Monitor toString

Đoạn mã dưới đây sẽ thực hiện quá trình trên:

```
Java.perform(function() {

    // Step - 1

    var array_list = Java.use("java.util.ArrayList");

    var ApiClient =
Java.use('com.android.org.conscrypt.TrustManagerImpl');

    ApiClient.checkTrustedRecursive.implementation = function(a1,
a2, a3, a4, a5, a6) {

        var k = array_list.$new();

        return k;

    }

    // Step - 2

    console.log("Hooking Java");

    const StringBuilder = Java.use('java.lang.StringBuilder');

    StringBuilder.$init.overload('java.lang.String').implementation =
function (arg) {
```



```
        var partial = "";

        var result = this.$init(arg);

        console.log('new StringBuilder("' + result + '");')

        return result;

    }

    console.log("Hooking new StringBuilder(java.lang.String)");

    // Step - 3

    StringBuilder.toString.implementation = function () {

        var result = this.toString();

        console.log('StringBuilder.toString(); => ' + result)

        return result;

    }

    console.log("Hooking StringBuilder.toString() hooked");

}, 0);
```

Lưu nó dưới dạng urlconn-hook.js và chạy câu lệnh sau

```
frida.exe -U -f com.example.myapplication -l urlconn-hook.js --no-pause
```

kết quả là chuỗi có định dạng JSON, được sử dụng để ủy quyền trong google cloud storage

```
new StringBuilder("undefined");
StringBuilder.toString(); => {"type": "service_account",
StringBuilder.toString(); => b
StringBuilder.toString(); => bs
StringBuilder.toString(); => bsid
StringBuilder.toString(); => bsides
StringBuilder.toString(); => bsides-
StringBuilder.toString(); => bsides-sf
StringBuilder.toString(); => bsides-sf-
StringBuilder.toString(); => bsides-sf-c
StringBuilder.toString(); => bsides-sf-ct
StringBuilder.toString(); => bsides-sf-ctf
StringBuilder.toString(); => bsides-sf-ctf-
StringBuilder.toString(); => bsides-sf-ctf-2
StringBuilder.toString(); => bsides-sf-ctf-20
StringBuilder.toString(); => bsides-sf-ctf-201
StringBuilder.toString(); => bsides-sf-ctf-2019
StringBuilder.toString(); => {"type": "service_account", "project_id": "bsides-sf-ctf-2019",
StringBuilder.toString(); => {"type": "service_account", "project_id": "bsides-sf-ctf-2019", "private_key_id": "6dd7fc48a8b1d49edf7f03f74bc47713bec7d989",
StringBuilder.toString(); => ----BEGIN PRIVATE KEY----
MIIeVAIBADANBgkqhkiG9w0BAQEFAASCBKYYggSIAgEAAoIBAQCbnA3X7qZ2Sec4
5W41r+YXJ31wJ28fWyyw0P2SoiTB/i3qTCKk/ltjP61TrJHB5MqKm6Vz/WGw7GSm
nd21xMFqcLwG8N7f+zhIK0XuvRBRs+cMEHw0RbHJUbC03ZagHKfFaLThzPY4x2r
Hh/N81UYi4TB8MGAaZCJH3pU9rTG4+ucx06pWnz3/Enzy0SmihR9Xb50kMY/Aq
7Nj7BAwH8oSMRl11laC8chl8wDdun6fKwYmncBwXSjropu8f6MR7vWM3f0PEua
YBz1ZpxVvYc11iMS0wINTaA1ZBuWhRenA1FwOCT4rISfBgYSqCTSoK5tOWS52tM
tutz/OVJAgBAAECggEAC9T230Uu125W1huHXdWt7n/vU1w7S5yTvhfDsWvkb+5
1+19od5SESrVh79sDR/n4NEkt8bIH5u1wJ3gP0Bw34qARHORX2TNjgxbpad231rY
ekuj+hwZatFA6AE00k+Bw+7L7twrs6j8pgLR4d11Z2ebYz2HhWu106s2pCmNlyM
S5n593YgfmzNotaoJ3dHwK68PuhRHH0dDqLnu0574dXkTFkEePcedyza007Iy
CoNTUYJx0D7aJbTQPVyqm7m0ewh1SYUdHSofcFAL1C9eT2nn6+c9qRgG03AwBCV
s9TmWR9+DwfhHrUmP8AKB4YKov/QGUv29sVId8YwQKbQ0TobsD5196xm2s+NnE
PabZ4w76sg/1o1dPU4w4+/8u/RUT+vJoumsvuf5n7KUXVAP8npu6LYouN1H+zv
ThTInxyTrrA5VamFhoEpeY80fboN1txk39fVbS2jw2GLZQLapN0XIYE0axRnJs
CSyc2LDVWVj0abjzx0CCF+c0s+QKbGQCV7kpsM3Im1w17FpJm/T9oakdvyZNzt3ZU
+DT8mPXh/nM5c6j9vzdGkq31ImH/VSSzVUfwQaxF8VzQ1A169frU/DSt8h7CpGd
LEzS0qq7ubbs7gODK/ZrwS0qhv8doegZGu0ntURfAVL7AnyKGJvQ2SLheVhMJeZ
m04mGME20wKBGXF5SwcGRGHM/WxKGvAG0JWtGwZtnjw+rAcRZFZAAPFqI3JfHwNe
gkyDwhjadvpiaY87T+ar1MXtEs1qCImbGfbGyKhw+5o0/1uH1Xs9vQgWghYMQX
JES6s0Q54IdImrOCHnCVfzk0rsTvkJyKh1M2G0SCIOBF7N1YG5PdRBT5AoGABEWt
```

Bây giờ chúng ta có thể quan sát tất cả các hàm gọi toString(), cuối cùng ta thấy được JSON hoàn chỉnh từ str4

```
[Gnemytom Google Pixel 2::com.example.myapplication]-> > ----BEGIN PRIVATE KEY---- MIIeVAIBADANBgkqhkiG9w0BAQEFAASCBKYYggSIAgEAAoIBAQCbnA3X7qZ2Sec4
5W41r+YXJ31wJ28fWyyw0P2SoiTB/i3qTCKk/ltjP61TrJHB5MqKm6Vz/WGw7GSm
nd21xMFqcLwG8N7f+zhIK0XuvRBRs+cMEHw0RbHJUbC03ZagHKfFaLThzPY4x2r
Hh/N81UYi4TB8MGAaZCJH3pU9rTG4+ucx06pWnz3/Enzy0SmihR9Xb50kMY/Aq
7Nj7BAwH8oSMRl11laC8chl8wDdun6fKwYmncBwXSjropu8f6MR7vWM3f0PEua
YBz1ZpxVvYc11iMS0wINTaA1ZBuWhRenA1FwOCT4rISfBgYSqCTSoK5tOWS52tM
tutz/OVJAgBAAECggEAC9T230Uu125W1huHXdWt7n/vU1w7S5yTvhfDsWvkb+5
1+19od5SESrVh79sDR/n4NEkt8bIH5u1wJ3gP0Bw34qARHORX2TNjgxbpad231rY
ekuj+hwZatFA6AE00k+Bw+7L7twrs6j8pgLR4d11Z2ebYz2HhWu106s2pCmNlyM
S5n593YgfmzNotaoJ3dHwK68PuhRHH0dDqLnu0574dXkTFkEePcedyza007Iy
CoNTUYJx0D7aJbTQPVyqm7m0ewh1SYUdHSofcFAL1C9eT2nn6+c9qRgG03AwBCV
s9TmWR9+DwfhHrUmP8AKB4YKov/QGUv29sVId8YwQKbQ0TobsD5196xm2s+NnE
PabZ4w76sg/1o1dPU4w4+/8u/RUT+vJoumsvuf5n7KUXVAP8npu6LYouN1H+zv
ThTInxyTrrA5VamFhoEpeY80fboN1txk39fVbS2jw2GLZQLapN0XIYE0axRnJs
CSyc2LDVWVj0abjzx0CCF+c0s+QKbGQCV7kpsM3Im1w17FpJm/T9oakdvyZNzt3ZU
+DT8mPXh/nM5c6j9vzdGkq31ImH/VSSzVUfwQaxF8VzQ1A169frU/DSt8h7CpGd
LEzS0qq7ubbs7gODK/ZrwS0qhv8doegZGu0ntURfAVL7AnyKGJvQ2SLheVhMJeZ
m04mGME20wKBGXF5SwcGRGHM/WxKGvAG0JWtGwZtnjw+rAcRZFZAAPFqI3JfHwNe
gkyDwhjadvpiaY87T+ar1MXtEs1qCImbGfbGyKhw+5o0/1uH1Xs9vQgWghYMQX
JES6s0Q54IdImrOCHnCVfzk0rsTvkJyKh1M2G0SCIOBF7N1YG5PdRBT5AoGABEWt
-----END PRIVATE KEY-----
{"type": "service_account", "project_id": "bsides-sf-ctf-2019", "private_key_id": "6dd7fc48a8b1d49edf7f03f74bc47713bec7d989", "pri
vate_key": "-----BEGIN PRIVATE KEY----- MIIeVAIBADANBgkqhkiG9w0BAQEFAASCBKYYggSIAgEAAoIBAQCbnA3X7qZ2Sec4
5W41r+YXJ31wJ28fWyyw0P2SoiTB/i3qTCKk/ltjP61TrJHB5MqKm6Vz/WGw7GSm
nd21xMFqcLwG8N7f+zhIK0XuvRBRs+cMEHw0RbHJUbC03ZagHKfFaLThzPY4x2r
Hh/N81UYi4TB8MGAaZCJH3pU9rTG4+ucx06pWnz3/Enzy0SmihR9Xb50kMY/Aq
7Nj7BAwH8oSMRl11laC8chl8wDdun6fKwYmncBwXSjropu8f6MR7vWM3f0PEua
YBz1ZpxVvYc11iMS0wINTaA1ZBuWhRenA1FwOCT4rISfBgYSqCTSoK5tOWS52tM
tutz/OVJAgBAAECggEAC9T230Uu125W1huHXdWt7n/vU1w7S5yTvhfDsWvkb+5
1+19od5SESrVh79sDR/n4NEkt8bIH5u1wJ3gP0Bw34qARHORX2TNjgxbpad231rY
ekuj+hwZatFA6AE00k+Bw+7L7twrs6j8pgLR4d11Z2ebYz2HhWu106s2pCmNlyM
S5n593YgfmzNotaoJ3dHwK68PuhRHH0dDqLnu0574dXkTFkEePcedyza007Iy
CoNTUYJx0D7aJbTQPVyqm7m0ewh1SYUdHSofcFAL1C9eT2nn6+c9qRgG03AwBCV
s9TmWR9+DwfhHrUmP8AKB4YKov/QGUv29sVId8YwQKbQ0TobsD5196xm2s+NnE
PabZ4w76sg/1o1dPU4w4+/8u/RUT+vJoumsvuf5n7KUXVAP8npu6LYouN1H+zv
ThTInxyTrrA5VamFhoEpeY80fboN1txk39fVbS2jw2GLZQLapN0XIYE0axRnJs
CSyc2LDVWVj0abjzx0CCF+c0s+QKbGQCV7kpsM3Im1w17FpJm/T9oakdvyZNzt3ZU
+DT8mPXh/nM5c6j9vzdGkq31ImH/VSSzVUfwQaxF8VzQ1A169frU/DSt8h7CpGd
LEzS0qq7ubbs7gODK/ZrwS0qhv8doegZGu0ntURfAVL7AnyKGJvQ2SLheVhMJeZ
m04mGME20wKBGXF5SwcGRGHM/WxKGvAG0JWtGwZtnjw+rAcRZFZAAPFqI3JfHwNe
gkyDwhjadvpiaY87T+ar1MXtEs1qCImbGfbGyKhw+5o0/1uH1Xs9vQgWghYMQX
JES6s0Q54IdImrOCHnCVfzk0rsTvkJyKh1M2G0SCIOBF7N1YG5PdRBT5AoGABEWt
-----END PRIVATE KEY-----", "client_email": "weather-companion-service-acco@bsides-sf-ctf-2019.iam.gserviceaccount.com", "client_id": "116037946827001874660", "auth_uri": "https://accounts.google.com/o
/auth2/auth", "token_uri": "https://oauth2.googleapis.com/token", "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs", "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata
/x509/weather-companion-service-acco@bsides-sf-ctf-2019.iam.gserviceaccount.com"}
[Gnemytom Google Pixel 2::com.example.myapplication]->
[Gnemytom Google Pixel 2::com.example.myapplication]-> exit
```

Vậy là quá trình ghi để toString() đã thành công, bây giờ ta có thể sử dụng gsutil để truy cập Google Cloud Storage từ giao diện dòng lệnh

```
$ gsutil auth activate-service-account --key-file=key.json
```

Activated service account credentials **for:** [weather-companion-service-acco@bsides-sf-ctf-2019.iam.gserviceaccount.com]

```
$ gsutil ls -p bsides-sf-ctf-2019 gs://weather-companion  
  
gs://weather-companion/flag.txt  
  
gs://weather-companion/weather.json
```

Chúng ta có thể thấy được flag.txt trong storage , ta sẽ dùng lệnh sau để copy mọi thứ vào để đọc flag

```
$ gsutil -m cp -r -p bsides-sf-ctf-2019 gs://weather-companion ./
```

Flag: CTF {buck3t_s3at5}

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc **(Report)** bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.DOCX và .PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).
Ví dụ: [NT101.K11.ANTT]-Exe01_Group03.
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài nộp.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT