



BÁO CÁO THỰC HÀNH LAB 6

Môn học: Pháp chứng kỹ thuật số

Nhóm: Pha Pha

THÀNH VIÊN THỰC HIỆN:

STT	Họ và tên	MSSV
1	Nguyễn Đoàn Xuân Bình	19521265
2	Trần Hoàng Khang	19521671
3	Nguyễn Mỹ Quỳnh	19520241

BÁO CÁO CHI TIẾT

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

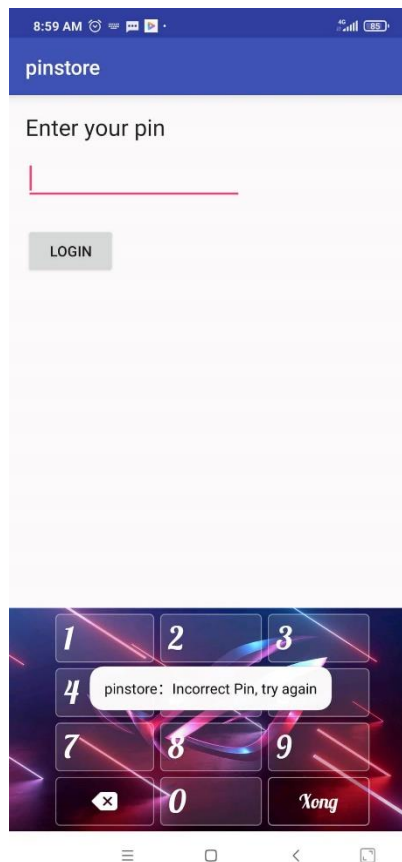
- Mô tả: Phân tích ứng dụng Android, tìm mã PIN trong ứng dụng để tìm flag.

- 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: (không có flag)

Cài ứng dụng lên máy, ta có giao diện app:



Với tính năng nhập mã PIN, nếu PIN sai sẽ pop up lên message “*pinstore: Incorrect Pin, try again*”. Nhiệm vụ của ta là tìm PIN đúng.

Phân tích file apk bằng công cụ **Jadx**. Trong file **AndroidManifest.xml** khai báo **2 activity** và activity để chạy đầu vào chương trình là **MainActivity**.

```
<activity android:name="pinlock.ctf.pinlock.com.pinstore.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<activity android:name="pinlock.ctf.pinlock.com.pinstore.SecretDisplay"/>
```

Đoạn code chính tại class **MainActivity**:

```
public void onClick(View view) {  
    String enteredPin = MainActivity.this.pinEditText.getText().toString();  
    String pinFromDB = null;  
    String hashOfEnteredPin = null;  
    try {  
        DatabaseUtilities dbUtil = new  
DatabaseUtilities(MainActivity.this.getApplicationContext());  
        pinFromDB = dbUtil.fetchPin();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    try {  
        hashOfEnteredPin = CryptoUtilities.getHash(enteredPin);  
    } catch (UnsupportedEncodingException e2) {  
        e2.printStackTrace();  
    } catch (NoSuchAlgorithmException e3) {  
        e3.printStackTrace();  
    }  
    if (pinFromDB.equalsIgnoreCase(hashOfEnteredPin)) {  
        Intent intent = new Intent(MainActivity.this, SecretDisplay.class);  
        intent.putExtra("pin", enteredPin);  
        MainActivity.this.startActivity(intent);  
        return;  
    }  
    MainActivity.this.pinEditText.setText("");  
    Toast.makeText(MainActivity.this, "Incorrect Pin, try again", 1).show();  
}
```

- Biến enteredPin sẽ lấy mã PIN từ ô input chúng ta nhập vào. hashOfEnteredPin sẽ được tạo từ việc lấy hash của chuỗi input. Vào xem hàm hash của CryptoUtilities.getHash()

```
public static String getHash(String input) throws NoSuchAlgorithmException,  
UnsupportedEncodingException {
```

```
byte[] input_bytes = input.getBytes();

MessageDigest md = null;

try {
    md = MessageDigest.getInstance("SHA-1");
} catch (NoSuchAlgorithmException e) {
}

md.update(input_bytes, 0, input_bytes.length);

byte[] hash_bytes = md.digest();

String output = getHex(hash_bytes);

return output;
}
```

Hàm thực hiện những thủ tục cơ bản với loại hash **SHA1**. Vậy chuỗi chúng ta nhập vào sẽ được hash bằng hàm này (SHA1).

- Tiếp theo có một biến là `pinFromDB` là mã PIN được lấy từ database.

- Sau đó, chuỗi được hash của chúng ta sẽ được so sánh với biến `pinFromDB` với cú pháp kiểm tra điều kiện `pinFromDB.equalsIgnoreCase(hashOfEnteredPin)` Nếu như thành công sẽ nhảy đến `SecretDisplay.class`, còn không thì hiện ra message báo lỗi như trên.

Vậy ta chỉ cần vào database và lấy hashed PIN được lưu trong database đó, vậy database này tên là gì? Vào xem phương thức `dbUtil.fetchPin()`:

```
public String fetchPin() throws IOException {
    openDB();

    Cursor cursor = this.db.rawQuery("SELECT pin FROM pinDB", null);

    String pin = "";

    if (cursor.moveToFirst()) {
        pin = cursor.getString(0);
    }

    cursor.close();

    return pin;
}
```

Vậy database được truy vấn ở đây là **pinDB**. File nằm trong đường dẫn *Resources* → *assets* → *pinlock.db*. Lưu file này lại:

Dùng command `sqlite3` trên Linux để đọc dữ liệu database:

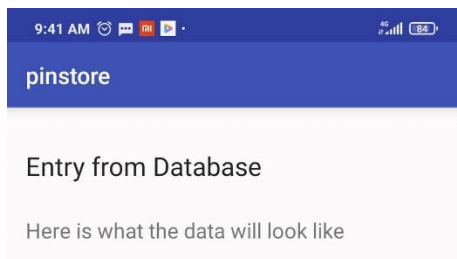
```
(virus@kali)-[~/Desktop]
$ sqlite3 pinlock.db
SQLite version 3.38.5 2022-05-06 15:25:27
Enter ".help" for usage hints.
sqlite> select * from pinDB;
1|d8531a519b3d4dfebece0259f90b466a23efc57b
sqlite>
```

Ta thấy có 1 dòng duy nhất và đó là hashed password. Thử crack nó bằng [crackstation](https://crackstation.net/)

Hash	Type	Result
d8531a519b3d4dfebece0259f90b466a23efc57b	sha1	7498

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

Mã PIN là **7498**. Thử nghiệm lại kết quả:



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:

Nhìn sơ qua thì chúng ta có một ứng dụng chỉ có mỗi giao diện, không có tính năng:



```
<activity android:theme="@style/AppTheme_NoActionBar" android:label="@string/title_activity_r2d2" android:name="com.example.blink.r2d2"/>
<activity android:name="com.example.blink.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

```
public void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.activity_r2d2);  
  
    ImageView image = (ImageView) findViewById(R.id.imageView);  
  
    String imageString = "  
data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAMCAgICAgMCAgIDAwMDBAYEBAQEBAgGBg  
UGCQgKCgkICQkKDA8MCgsOCwkJDRENDg8QEBEQCgwSxIQEw8QEBD/2wBDAQMDAwQDBAgEBAQCwkLEBAQEBAQEBA  
QEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBD/wgARCAGEAmwDAREAAhEBAxEB/8QAHQAB  
AAICAwEBAAAAAAAAAAAAAIA..."  
  
    byte[] imageBytes = Base64.decode(imageString, 0);  
  
    Bitmap decodedImage = BitmapFactory.decodeByteArray(imageBytes, 0,  
imageBytes.length);  
  
    image.setImageBitmap(decodedImage);  
  
}
```

```
echo "<base64-string>" | base64 -d > image.jpeg
```

```
[virus@kali] - [~/Desktop]
>.....
L77C9MeIBdun5/iBige6KsIrvn2oyxZEZLCTeFuf72g6LwjaHwE/4E/wCBKBR8IWJEsIwFlwCu+YQAAj+gUY1FXf/ALCtJULF6KuXLL/65f/EACQQAQABAMDBAICAwEBAQAAAAERACEXQVfHEHGRQFCBos
CKMPDBoEgCJD/2GaIEQAAT80/wZenvUvblwFgRocnrgRhdQ/wBdpUjAo+KsMV53sHhKkGVmFsPLpJoztWf0oXct/eSniWt/aS1lIHd/mZVQH+Z/2Xz7ZhfqC4q/YGKNCvmrj/77DX/dsvkK0J8raMn
Qaye3FyqFgEBCpJauU8VKovf9eYr2f12Ml2m8cPMw6K3bG7d8X5eSkigAX08qE957SPmkWhkLk3Jf0z5JTF4fJwHfFQVBUDFqYdqb0B30GL8XlJTLX0Lb2wPwNF9p4tCetKwBz64fz25N06i1IBhQFv0B
Rt/K2o1fWdE1013Xl9y1JtarJGocS+rXfuo6K13l21wZwVtEFQ416nV54Ngg2qCqCajmXh8F10g/pQbzpQ3/y7qUwL7U1hsAzeieSpAGSv8tU9FqYHC1qm1Tn+ykBAyV7JUpi2Um16yxgUdXB3xof
8AlJqDw1PzeuioZdAH2LI33p118k8+6AkN6H8T6bf7DUIeLh2G9v5NvNQ5FqYwqABARwWmH4Lk059mzRR4JzkoPul1Jm18-8rtQeQh16v8tU9rUqbtD/VcAPeNgB/51XKN1vuxXmpj4n8+TJ+WTiG529
F8V8XACQV9J91Nwmj07epaLw+PaqU6L5mW6d+6p2GvsmjYq/1tW6LYVdWd3m18k2L8abT2y0tR8lQoN+Yswd6sJgdThDpXp/Agz+woD+O6GAMoCBjF5WtXkUd1i0F08Jdyovcn3B8uAYKwB0UJ1yDru
F9EG9d9dHvAMNkmget68l0/pFovJhV8Y6TfL5mS3Vg/5AADF3JC0RY1aMhJ05A47fC98JWCQJ2I5aGgQvCmXAsYAvUAMdbi1dHououieZmGgUvFQDQ504F7QZIE1a6vpm3pXqWwK867lafawf64
U01DUke0C7hi1i04K8vNWNXBWj0Y1AycV7WCXD+Yq8tXqCwXp/CMXfQ2oQrQ64bBux/G85/qVcUL+Pi+LrF5L/m7fF98P8xtYqUv1LBBAt1YAL2t9dQ6Ck/y2J5V64G5I1wS44f2z8BzmQdk/D
ns4kell80NQTuMlIKLrFvukuAtJcKPEJB9vIudb0t/NDc4cseQFKX5F7ZFcErE8P15vA+AES8b3pKzMDyUgtXhW9RjYqF55N44hwbjcVqHouCNkXElxvBTUgRwHmUcu+TJnKK/196a28bW7ThOk
UWbnd5Ffmg2CJg5ySkTRGf48b++28UNdg3NKhIT4WemTALvUtg66vBLpBkicYq9rR1L2tSRnK5iXQCE22axMga3H08Q34151cvBesC4k16ZkHbCdqCnHm1LrU5gt65G9em4UoiV06o2UdV9r2RXN1X
n6p6A412-1BT/1doAzr1INKS1J33+yD4n41r06WdKX0DXPKAMRehsM0Q2oYq21f/wASxm0Qf5qbj/3ZTUEU5q4Q3/CXM0N53BmF6Bw2rvn9mC01Q1ML7f15VLqC1O4ay+XxU/41idJuzT80sRoaqE/L
08F1ATvYfWbCwborj1h1HVC0AUXK5WZstHac4396CeqJoaHmcOf29zKfYqU6f4VvN1j312/dskVUxM6E1KfYcVLVSyuaal3/4AB13QexpF7FMLV0L3pUvHvaoommdKYCs+PuH8MGW3B81NqvrV9PwC4W4E3
JYF7Tq03eXmKcid0Mv4n/ABTQCPUSKDX3TnzqtE4136CpaQnV52GvN2KtY9j2Rt0J42r/9Rmnc+ZqgNBX5E+1Kmw8Q/K51/2HustK1APKBXmPzNpTUbKHoak0e0fB7Z5N055636D186p9wZ9vC9UdP
CGRW9153PvAZV72M147b1kbn1m0Rqbdh8gSRtIMN02B8u8KjQJG5+9PuiPcbgk1kYbU1QRWC2DASRps6D8uX04yAA0vVrB9TdpwG5J3v3n3Uhcya109wWdH41C7E0924g+DWMKrf94L0IBAS56FK
F32Ld1Y0c0r27reachNTp+rf8A6D0tXK1LnkuG/6wXSSG4V02-KtCy9t1q6b7g6vdpnrWk-WuPflUfUwD00X4y1XAAV/PctUSgmpK9W3CvJn1V6L951fZ5472r0CNkXU1ZrHr49W044UgW6M1
h8VbUkUxTm/SDB0rQRWbV+Yoaahr82AdP22mk2Z8yUuXn131G5hXmU1Xpny8A335XU1Q6g4wkh7Uk6d811B8UFVB0L5TfXKMHgK+7W0DFpFgG0t87Z0C0W154SAst1/v4h1tU2QXVEYtZAF
b0dJ9fUMKLn2Dz1o1k2JZ1EDrCue4Ceo4Y3o09+324yCmC83H5u1Y8r7pL0y5H8ip1tEO3UyW1E65/8DlV1tAMM1Pdqe7X1Zm3p1m2VAcT54Lx/60vFLu5CJDNJgLKASVcCL25L2UmYU0
8IGW1G6E0dH5YkppD55TcJAGu04A4KnnKtSc1GwU6S4UoXsclJrdNoaAXtKcJLQWQY1Arh6eCgXN0AD1ipJPHNLEtEsqgEldFdpJ3mN1ESQ2ImPUYTh+7euVudUxTSA0Zf7Zg7Ac5b1jWxJGL
gP3X2A02W4EYK0CwUzKtErvnmvkn+n9MJ1m0K30J0riiYt+K25oEaIKLq3m14hCX00D2+DQmYALAAATYkUjDuh3D304GVRSE41htFNWnR5RFPtmnrg4J3WS4A4ZCJD3AD2dK6ke611n/SLnpsbmC0
DBFErUw5njg1i1WVjvkw3nAoBYvBmpI4GcQq250tAv8pghEm4uyuc/20633k3MwFaoMpmCyr3qeoY154MXLUaP7RJCJCJCWJCIECsSTPKZnKzEAFzocb7LU0DTN6Y012XxtKwxiY05SPATGcZ
KxHem3K37wEabCDZKdUyXU/p4P1/B3C0ALhB87K0gBw0U5Y7aGAg12RocYpJgd9pJ9js0aag3pABX4ZndW6LUkRt9c405Fv89S9CNtKtBYK124q6604GSV1c+RusR0KdN9V56WTm3pJ3+
618C1WR1SNRm8f5pV60X00uHrEPk0ocXkA57gY9JkwNjy1tQatY512K55dY0QABKLl0tdB6g2tK03UwE/tbPU1U3wCneK141z/f/7TjPLAEAC1YJ3B+67Dv0zhG0kLBSvnc18G6ABu15318X1j
mWcYqGfGE0Nd9vMaLU1lJgSE1TLRCLTHT00Y1D999P9vjs69W1F0UB1N2L79UeEGU5S40LL2pZqGqUadLKG1NA1+p98e/HP5HSPCCUUTxcmJmJ1gZ7BTWqunRZ2/AJ6eJ0u56G2U5W/NKcA/C6
x/asu2JA46z899C2eP2XtK7c5r1CELGRHt00Y21SgkZ1yktymbzADwPmFMDY5L7Q3BAJ142Ue4rQp3jDQGQ+PxR57F4q4kqW4v4B5NS0M2LKCVU4Ym112P7MwYAL19K2WQ0W0GKJ3sJrKwP3QY5ES
YVJ1Qn1CAmAwNdM6LBH81R5EGEgkwl80nDVG9J3v8p8J9sorpAKTME86L11nQhQX0qZyRhcYUwSh5eH2TXy99rH/vK8hGAbdR8Q1Q20AaGLDAR80J4AgEJ3KtCkN4X+D5U7YfQ0+K0rt1uc
bAHH1M5S2uFkyV5BHP0tdHdK5ySGA8r/PtO4yVIngP7eqCjEwH9rUJd7ESMAE8r7bL2DYkEYbnsJ2D155+L3n1Awg3Kmp0BMPMQACtAk6d5awxmnm31BZNfEq4lME3v+H6Sg05v3BQJue10xt1R09D
ueAffcrtWbdeE+ej1j+hXAKm4SQ6MnFEBJZ5Z0YUyHsAMGud4YAL180AECpmME0F7Q3qQL1ImanGLhNB7C522ZrU2Pgy1KYCTGmF5m04LsI0nPMKc1z1eXZKM19TKuzN0pDERpRqqEBo+eg7
7NRpJKW4yapigSpQfmqkdMkDnSpA4LK6fY+fuUAGMdY2406s1u2m+BJM5U5uAGTCsXyNF3VAhNR8FVfEmht7k7W0bK7V5WmK0taM5Vo15y8h/wC9ycaKcpqj/sqQKCoHnp+CR1Tn17Wu1
D8K3C0KXyQvYqP5o3c1Jg4HmJ3XmYvYtPbWkrGdCE00B8H5fNrLXN86jY0rCQkx8t43QoFCE4pF4W2dW3YqJbgy0V24598E2vAC2AWbYkKtE9552TBAAEZanEGZCpxTWfEMjB3
35g8BRCCZr202eNCoZncESRMQYkUxYVJMMNLTGkbJ3dKpdu0F8ASJBB4XyWdXEqEqlgrVpXvNN5L136UmrM9KVL05CysAAEKnIM2Wk3B0dp5o/1Yt1QRQAg10G6Wd7414WbwpPVbXGqY1ABT
JMBVafwS8W5hYtcPcisEFCRAALa1Jl0Ue209gDKPLF3V5GLSRxmJGKt88gYuo8YzqW5NwBG0w8CUY9APGB1Yx1B5T1EepNLJDFEBM1JmD65qCQJ15x5qA0r6cb0tm2eQnmqW7FLAkxgAAfV6GQZDJJ
yx1RG/8X/20=] [base64 -d -> image.jpg
```

Mở file ảnh lên:



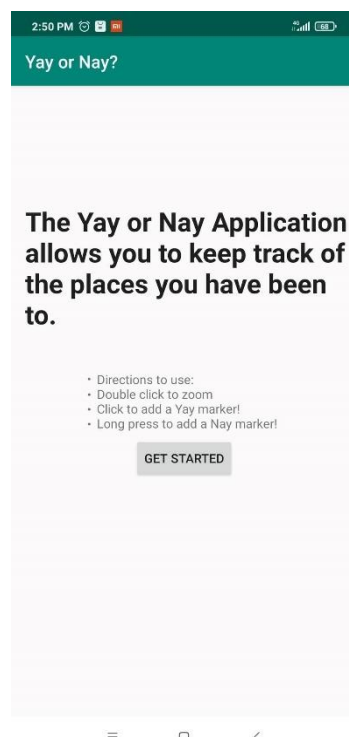
Flag: CTF{PUCKMAN}

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

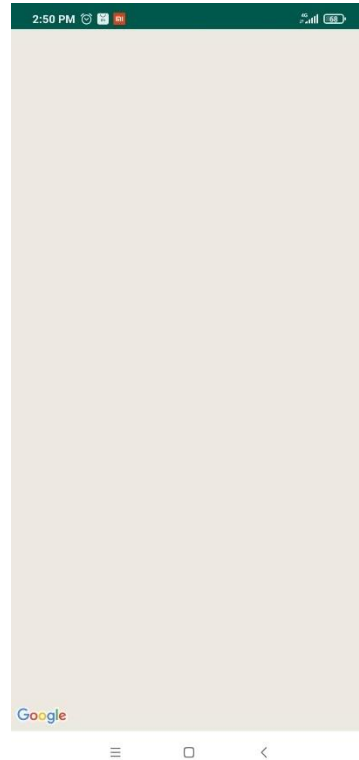
- Mô tả: Một ứng dụng có tính năng ghi nhớ các địa điểm mà người dùng muốn hay không muốn tham quan chỉ bằng dấu tick đơn giản trên bản đồ. Tìm flag.
- Tài nguyên: kb03_yon.apk
- Yêu cầu – Gợi ý: Decompile, chú ý CSDL của ứng dụng.

Gợi ý:

Ứng dụng có giao diện chính như sau:



Nhấn vào **Get Started** để sử dụng:



Có lẽ API Map khá cũ rồi nên không thấy gì hết. Theo hint đề thì ta xem database của App, mình dùng SQLiteStudio để view dữ liệu trong DB:

Grid view		Form view		
		Total rows loaded: 199		
	date	latitude	longitude	color
1	02/03/2019	37.7842927	-122.4053593	120
2	02/03/2019	37.7838412	-122.4041845	0
3	02/07/2019	37.7863323436302	-122.42828886955978	120
4	02/07/2019	37.78513679327185	-122.40235358476639	120
5	02/07/2019	37.782343920755	-122.40469984710217	0
6	02/07/2019	37.78198221749665	-122.40199450403453	0
7	02/07/2019	37.76736156477806	-122.4675929546356	120
8	02/07/2019	37.77400601208542	-122.42873679846527	120
9	02/07/2019	37.77385071412083	-122.42837738245726	0
10	02/07/2019	37.77321785829125	-122.428442761302	0
11	02/07/2019	37.77201626535686	-122.42656253278257	0
12	02/07/2019	37.77301114541446	-122.42201820015909	120
13	02/06/2019	37.77373410792617	-122.41757109761237	120
14	02/06/2019	37.70683838819129	-122.34466664493084	120
15	02/06/2019	37.70870495630154	-122.44387753307818	0
16	02/06/2019	37.6842005525645	-122.44341988116501	0
17	02/06/2019	37.68474342576108	-122.4414910376072	0

- Cột đầu tiên là thời điểm mà user đánh dấu vị trí
- Cột 2 và 3 là tọa độ X,Y của địa điểm
- Cột cuối là giá trị màu (120 hoặc 0)

Thông tin trên được thể hiện tại hàm **onMapReady()** trong class **MapsActivity**. Hàm này mỗi khi map được load lên thì sẽ đồng thời load tất cả các điểm đánh dấu cũ từ trong cơ sở dữ liệu ra. Thông tin của một điểm đánh dấu được định nghĩa bằng class **Location** :


```
public Location(Date date, double latitude, double longitude, float color) {

    this.date = date;

    this.latitude = latitude;

    this.longitude = longitude;

    this.color = color;

}
```

Đoạn code lấy thông tin điểm đánh dấu ra tại **onMapReady()**:

```
DatabaseUtils dbUtil = new DatabaseUtils(getApplicationContext());

ArrayList<Location> locations = dbUtil.fetchLocations();

Iterator<Location> it = locations.iterator();

while (it.hasNext()) {

    Location location = it.next();

    LatLng temp = new LatLng(location.latitude, location.longitude);

    float color = 120.0f;

    String label = "Yay!";

    if (location.color == 0.0d) {

        color = 0.0f;

        label = "Nay!";

    }

    this.mMap.addMarker(new
MarkerOptions().position(temp).title(label).icon(BitmapDescriptorFactory.defaultMarker(color)));

}

LatLng bSidesSF = new LatLng(37.7842927d, -122.4037178d);

this.mMap.moveCamera(CameraUpdateFactory.newLatLng(bSidesSF));

this.mMap.animateCamera(CameraUpdateFactory.zoomTo(10.0f));
```

Từ cơ sở dữ liệu phía trên, ta chạy App để load các điểm marker lên Map xem thử phân bố như thế nào. Khi chạy app trên *Simulator Device* của *Android Studio* thì bị lỗi. Đọc **Logcat** :

```
2022-05-30 14:34:40.951 12247-12313/com.example.yayornay E/Google Maps Android API: Authorization failure. Please see https://developers.google.com/maps/documentation/android-api/sign-in
2022-05-30 14:34:40.952 12247-12313/com.example.yayornay E/Google Maps Android API: In the Google Developer Console (https://console.developers.google.com)
Ensure that the "Google Maps Android API v2" is enabled.
Ensure that the following Android Key exists:
API Key: AIzaSyCHUNLbULEWw1LSPzU3vLIRT3P0TC7WW0g
Android Application (<cert_fingerprint>;<package_name>): 64:0B:75:60:1B:8E:95:C9:30:DD:01:04:16:87:82:C9:80:58:2D:01:com.example.yayornay
```

Xác thực **API Key** không thành công. Có thể API Key bị lỗi thời, hết hạn sử dụng => Cố gắng tìm và đăng ký **API Maps (Google)** nhưng mà không free

→ Dừng kịch bản

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:

Đầu tiên vẫn theo truyền thống, ta kiểm tra file **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    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:supportsRtl="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>
```

Phiên bản hệ thống tương thích là SDKverion 26 hoặc 27 → Android 8.0 hoặc 8.1

Chương trình có activity chính là **com.example.myapplication.MainActivity**. Chương trình lại gọi hàm **AsyncTaskC1385a** để xử lý và ta thấy có API gọi tới endpoint **weather.json**

Tuy nhiên khi chạy app thì bị lỗi khi xử lý với endpoint:

```
2022-06-01 19:39:05.479 5089-5024/? I/com.b.b.b.d: Failed to detect whether we are running on Google Compute Engine.
2022-06-01 19:39:06.265 5089-5024/? W/System.err: java.io.FileNotFoundException: https://storage.googleapis.com/weather-companion/weather.json?GoogleAccessId=weather-companion-serv
2022-06-01 19:39:06.265 5089-5024/? W/System.err:     at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getInputStream(HttpURLConnectionImpl.java:251)
2022-06-01 19:39:06.265 5089-5024/? W/System.err:     at com.android.okhttp.internal.huc.DelegatingHttpsURLConnection.getInputStream(DelegatingHttpsURLConnection.java:210)
2022-06-01 19:39:06.265 5089-5024/? W/System.err:     at com.android.okhttp.internal.huc.HttpsURLConnectionImpl.getInputStream(Unknown Source:0)
2022-06-01 19:39:06.265 5089-5024/? W/System.err:     at com.example.myapplication.a.a(Unknown Source:496)
```

➔ Dừng kịch bản 4