BÁO CÁO THỰC HÀNH

Môn học: Pháp chứng kỹ thuật số

Lab 5: Mobile Forensics

GVHD: Đoàn Minh Trung

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT334.P11.ANTT

Nhóm 12

STT	Họ và tên	MSSV	Email
1	Nguyễn Lê Thảo Ngọc	21521191	21521191@gm.uit.edu.vn
2	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%

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

BÁO CÁO CHI TIẾT

A. KỊCH BẢN 1

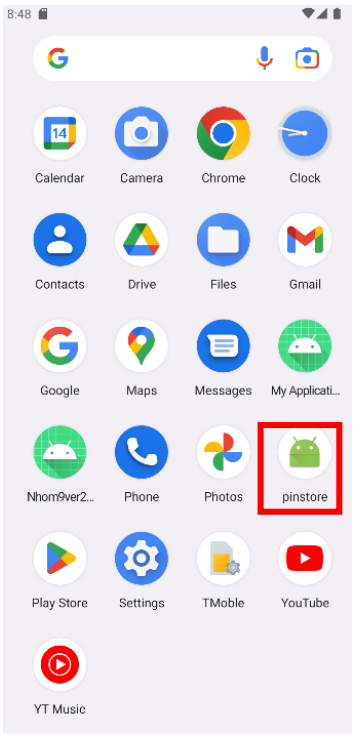
1. Kịch bản 01. Thực hiện phân tích ứng dụng Android

- 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.

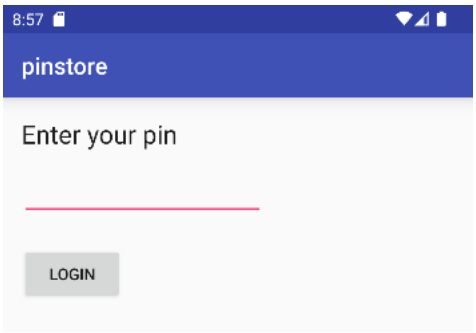
- Vào ứng dụng Android Studio và mở máy ảo android lên. Sau khi mở máy ảo, vào genymotion kiểm tra xem máy đã được bật lên chưa bằng lệnh ".\adb devices".
- Nhóm sẽ chèn file apk đã cho vào máy bằng lệnh ".\adb install <địa chỉ file>".

```
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb devices
List of devices attached
emulator-5554   device

PS C:\Program Files\Genymobile\Genymotion\tools> .\adb install "E:\semester7\Phap_chung_ky_thuat_so\Lab\Lab05-Mobile-Forensics\RES_mobile-forensics\resources-session05\pinstore.apk"
Performing Streamed Install
Success
PS C:\Program Files\Genymobile\Genymotion\tools>
```

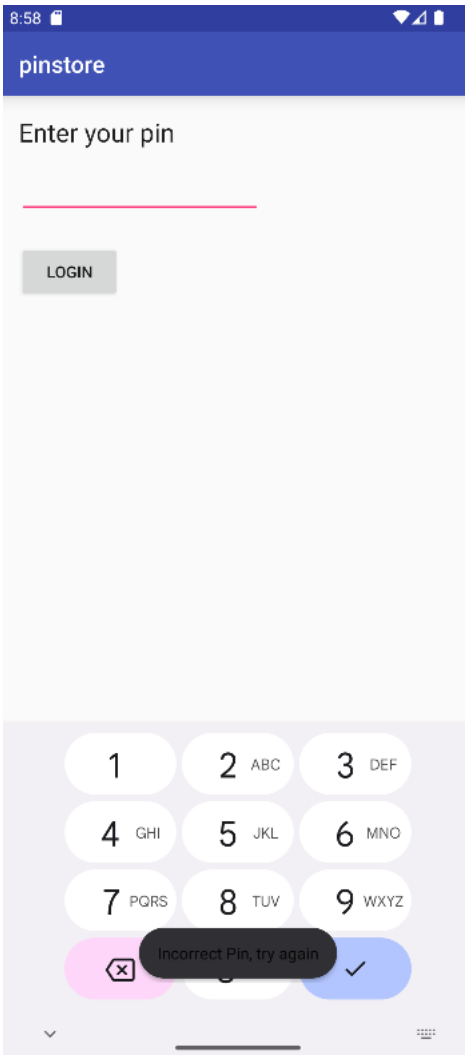
- App Pinstore đã được cài đặt thành công.



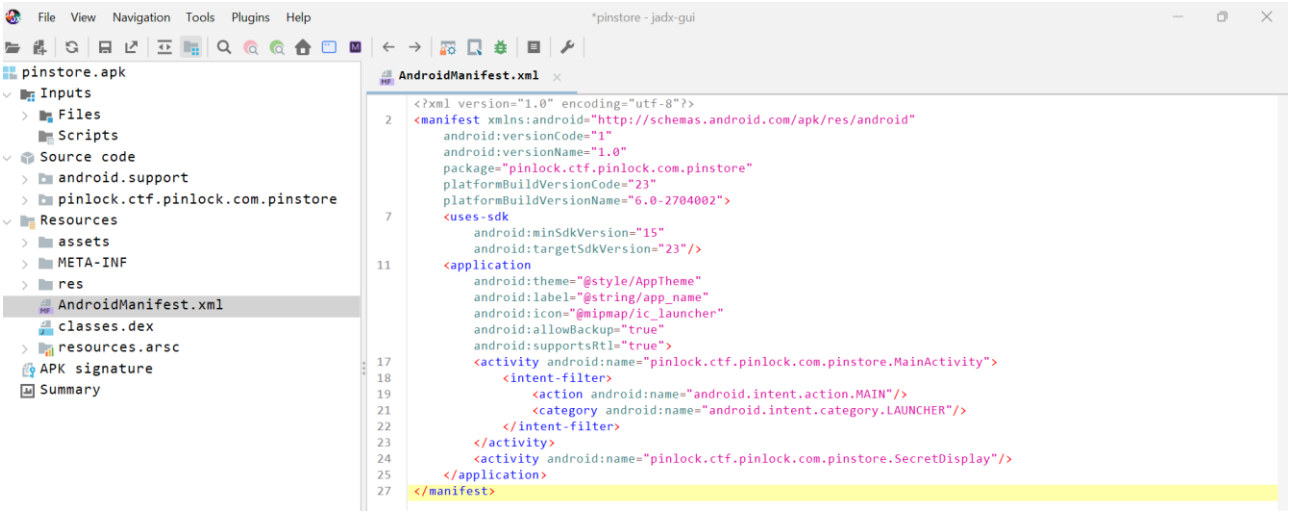
- Giao diện chính của app.



- App có tính năng chính là nhập mã pin. Khi nhập mã pin sai, màn hình sẽ hiện lên thông báo “Incorrect pin, try again”. Vậy nhiệm vụ của chúng ta là tìm mã pin đúng.

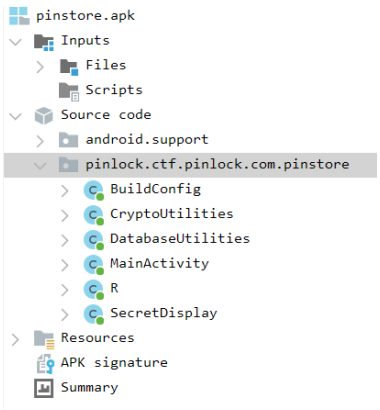


- Tải công cụ Jadx để tiến hành xem code của file apk. Nhóm sẽ thử vào file AndroidManifest.xml để xem bên trong có khai báo gì đặc biệt không.

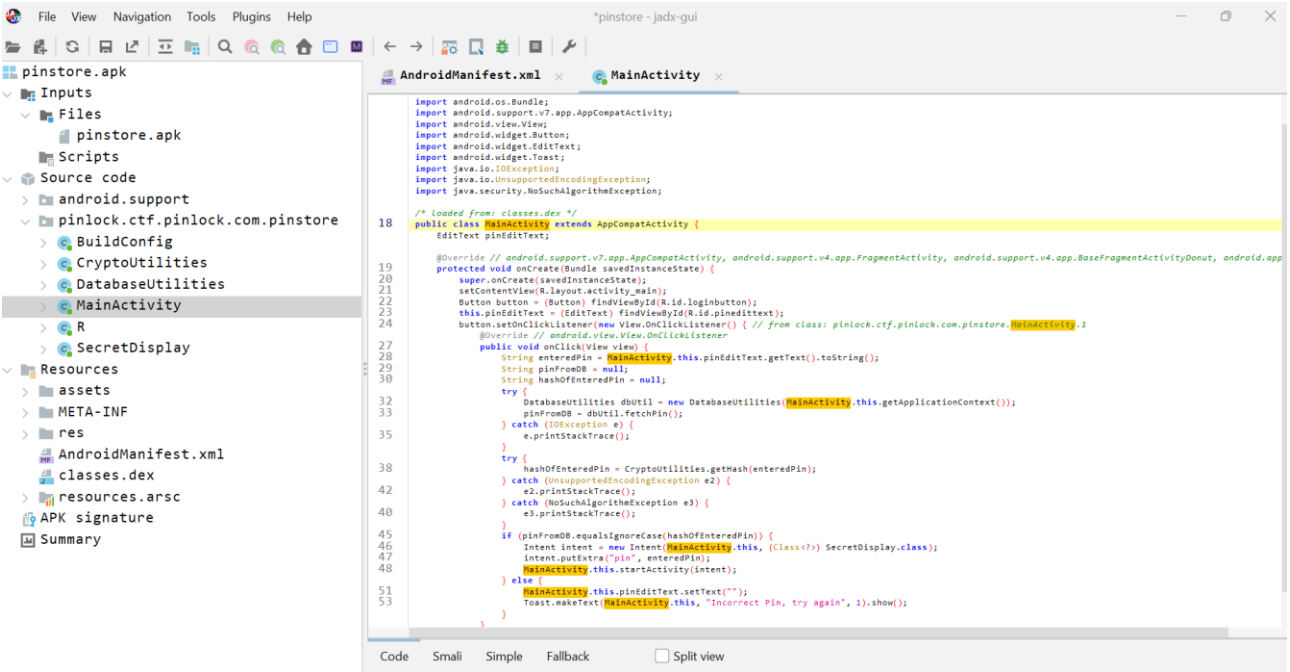


⇨ Có vẻ không có gì đặc biệt ở đây.

- Trong file apk này có tổng cộng 6 class.



- Nhóm sẽ xem đoạn code chính trong class MainActivity trước.



- Đoạn code trên sẽ lấy chuỗi pin mà người dùng nhập vào ô input, sau đó dùng một hàm hash trong class CryptoUtilities để hash chuỗi pin nhận được.
- Sau khi có được mã hash thì nó sẽ so sánh với mã hash của pin được lưu trữ trong database. Nếu mã pin đúng thì sẽ chuyển tiếp đến Secret Display, nếu không thì in ra màn hình "Incorrect pin, try again".
- Vào class CryptoUtilities thì thấy mã pin được hash bằng SHA-1.

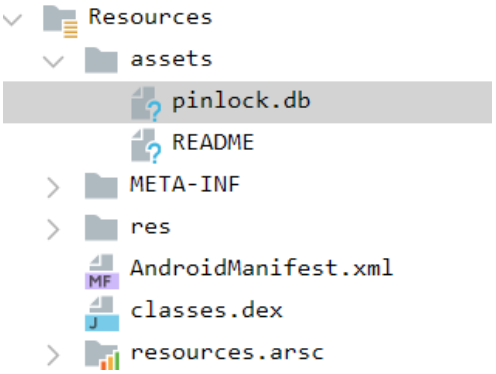
```
public SecretKeySpec getKey(String version) throws Exception {
    if (version.equalsIgnoreCase("v1")) {
        Log.d("Version", version);
        byte[] keyBytes = "t0ps3kr3tk3y".getBytes("UTF-8");
        MessageDigest md = MessageDigest.getInstance("SHA-1");
        SecretKeySpec keySpec = new SecretKeySpec(Arrays.copyOf(md.digest(keyBytes), 16), "AES");
        return keySpec;
    }
    Log.d("Version", version);
    byte[] salt = "SampleSalt".getBytes();
    char[] pinArray = this.pin.toCharArray();
    SecretKeyFactory secretKeyFactory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
    KeySpec ks = new PBEKeySpec(pinArray, salt, 1000, 128);
    SecretKey secretKey = secretKeyFactory.generateSecret(ks);
    SecretKeySpec keySpec2 = new SecretKeySpec(secretKey.getEncoded(), "AES");
    return keySpec2;
}
```

- Vậy mã hash pin được lưu trữ trong database, nhóm sẽ vào class DatabaseUtilities xem thử.

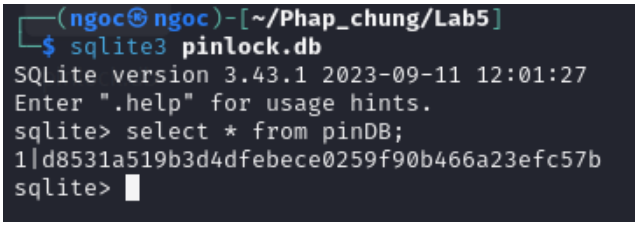
```
/* loaded from: classes.dex */
public class DatabaseUtilities extends SQLiteOpenHelper {
    private final Context appcontext;
    private SQLiteDatabase db;
    private static String pathToDB = "/data/data/pinlock.ctf.pinlock.com.pinstore/databases/";
    private static String dbName = "pinlock.db";

    public DatabaseUtilities(Context context) throws IOException {
        super(context, dbName, (SQLiteDatabase.CursorFactory) null, 1);
        this.appcontext = context;
        createDB();
    }

    public String fetchSecret() throws IOException {
        openDB();
        Cursor cursor = this.db.rawQuery("SELECT entry FROM secretsDBv1", null);
        String secret = "";
        if (cursor.moveToFirst()) {
            secret = cursor.getString(0);
        }
        Log.d("secret", secret);
        cursor.close();
        return secret;
    }

    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;
    }
```

⇨ Database có tên pinlock.db và bên trong có một table tên pinDB.

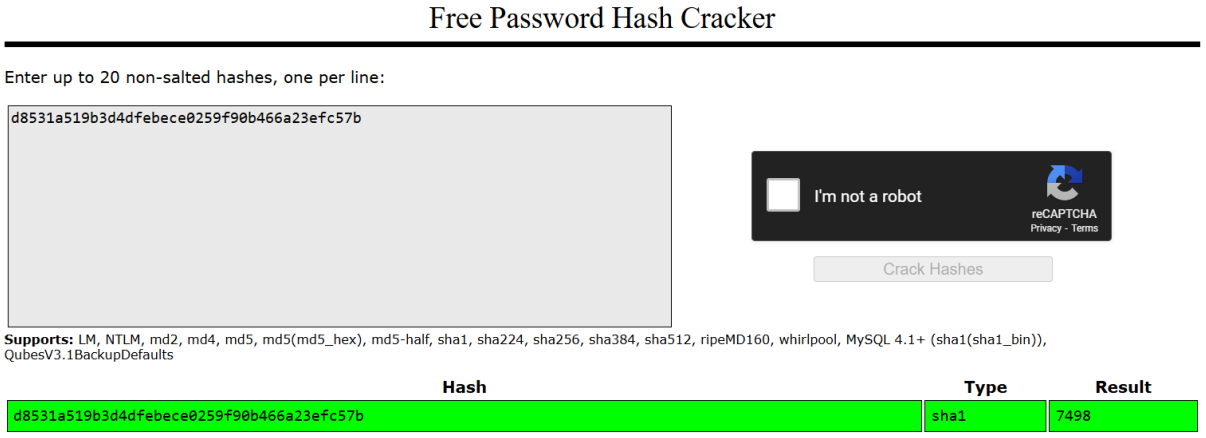
- Sau khi tìm kiếm thì em tìm thấy pinlock.db ở đường dẫn này.



- Export file này ra và sử dụng sqlite để đọc dữ liệu bên trong nó. Như đã đề cập ở trên, trong database này có một table pinDB nên hãy thử truy cập vào table này.

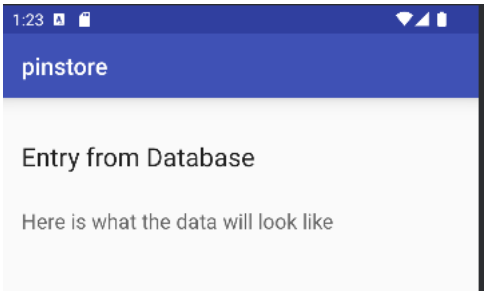


- Dùng một công cụ bất kỳ để giải mã chuỗi trên.



⇨ Mã pin là 7498.

- Dùng mã pin vừa tìm được nhập vào app Pinstore được kết quả.

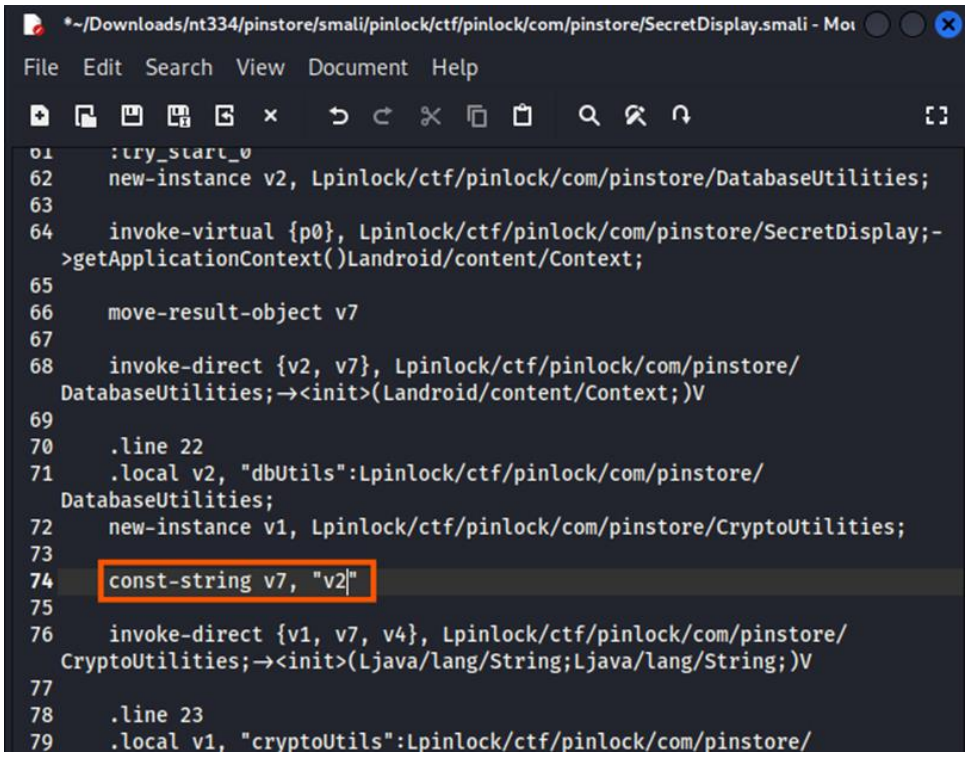


⇨ Đã đến được Secret Display nhưng chưa có flag.

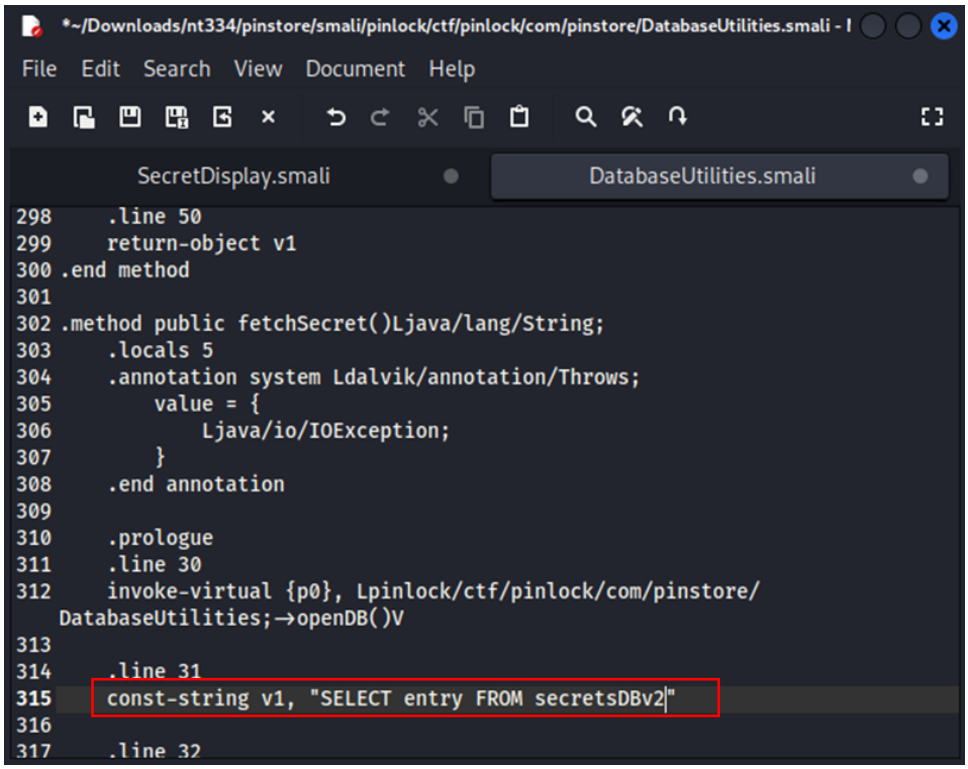
- Thử xem lại class CryptoUtilities, Phương thức getKey nhận tham số version và trả về một đối tượng SecretKeySpec (khóa mã hóa) cho thuật toán AES.
- Trong hàm getKey hiện tại đang sử dụng phiên bản v1 và nếu sử dụng v1 thì hàm sẽ sử dụng chuỗi key cố định là "t0ps3kr3tk3y".
- Nếu version là phiên bản khác, nó sẽ dùng kỹ thuật PBKDF2 (Password-Based Key Derivation Function 2) để tạo khóa từ pin và một chuỗi salt cố định (SampleSalt).

```
public SecretKeySpec getKey(String version) throws Exception {
    if (version.equalsIgnoreCase("v1")) {
        Log.d("Version", version);
        byte[] keyBytes = "t0ps3kr3tk3y".getBytes("UTF-8");
        MessageDigest md = MessageDigest.getInstance("SHA-1");
        SecretKeySpec keySpec = new SecretKeySpec(Arrays.copyOf(md.digest(keyBytes), 16), "AES");
        return keySpec;
    }
    Log.d("Version", version);
    byte[] salt = "SampleSalt".getBytes();
    char[] pinArray = this.pin.toCharArray();
    SecretKeyFactory secretKeyFactory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
    KeySpec ks = new PBEKeySpec(pinArray, salt, 1000, 128);
    SecretKey secretKey = secretKeyFactory.generateSecret(ks);
    SecretKeySpec keySpec2 = new SecretKeySpec(secretKey.getEncoded(), "AES");
    return keySpec2;
}

public String encrypt(String plaintext) throws Exception {
    byte[] plaintextBytes = plaintext.getBytes();
    this.cipher.init(1, this.key);
    byte[] ciphertext = this.cipher.doFinal(plaintextBytes);
    Log.d("Status", Base64.encodeToString(ciphertext, 2));
    return Base64.encodeToString(ciphertext, 2);
}
```

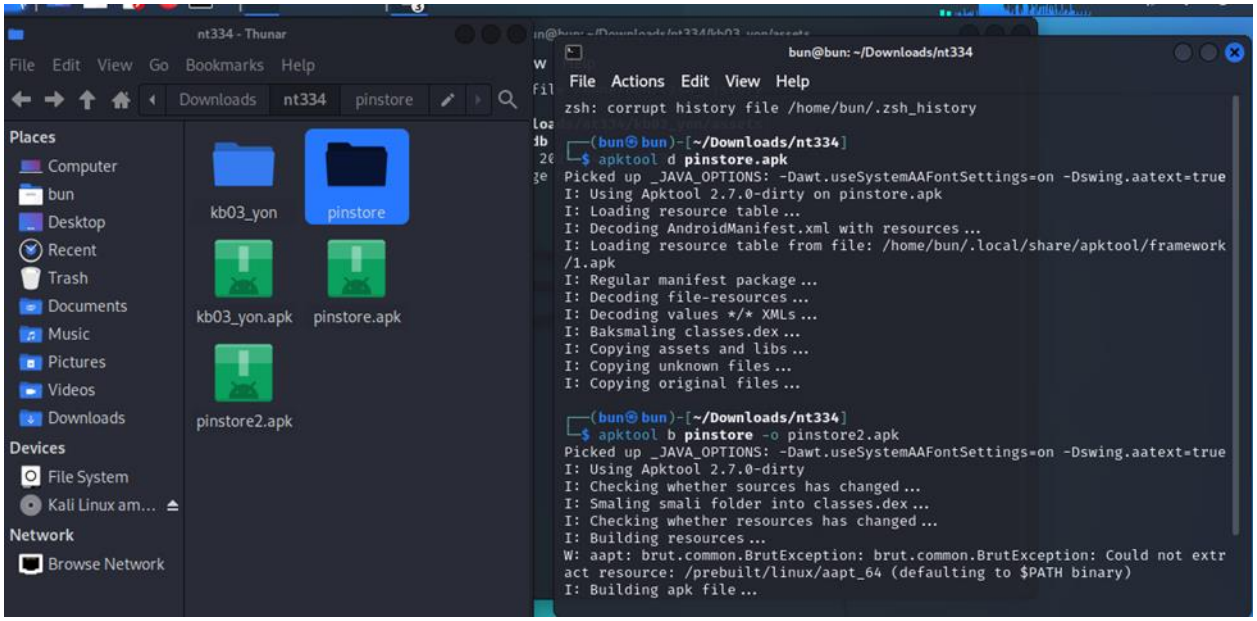
- Vậy bây giờ, chúng ta sẽ đổi version từ v1 sang v2.
  - Đổi trong file SecretDisplay.smali

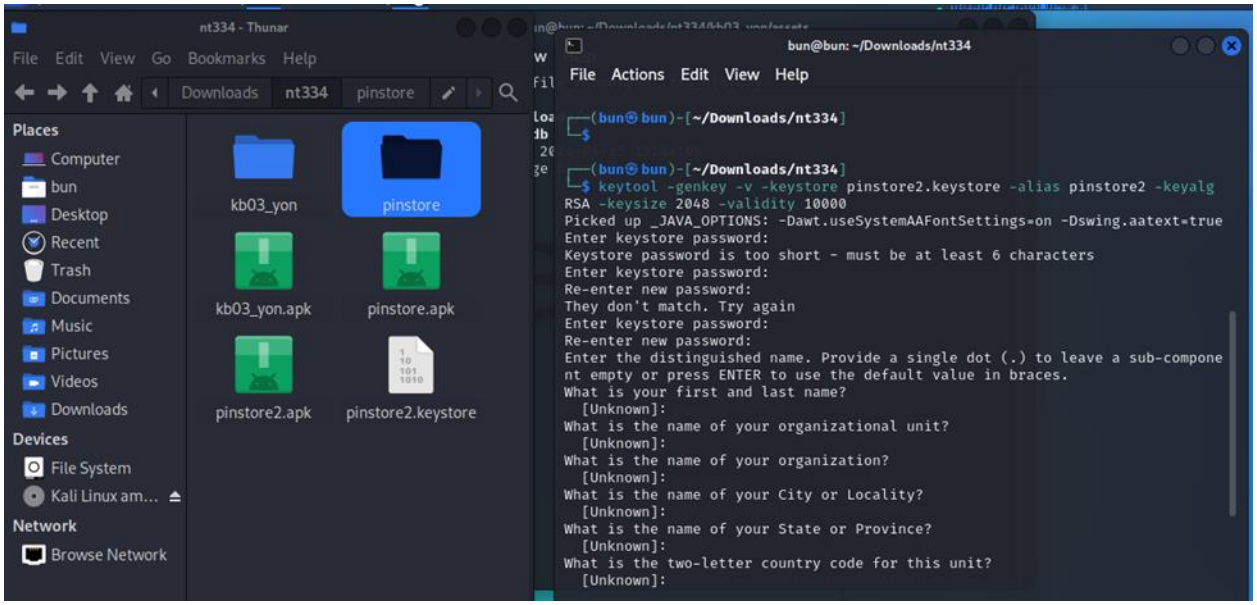


- Đổi trong file DatabaseUtilitites.smali



- Sau khi sửa đổi code thì nhóm sẽ tiến hành build và kí lại cho file apk.



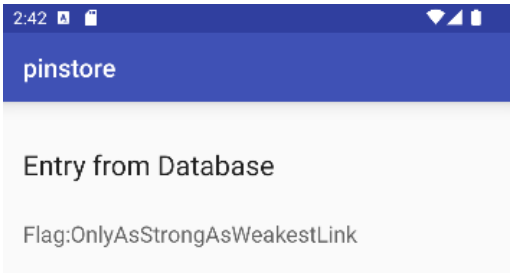



```
┌──(bun㉿bun)-[~/Downloads/nt334]
└─$ apksigner sign --ks pinstore2.keystore pinstore2.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Keystore password for signer #1:
```

- Tải app lên máy như bình thường.

```
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb install "E:\semester7\Phap_chung_ky_thuat_so\Lab\Lab05-Mobile-Forensics\pinstore2.apk"
Performing Streamed Install
Success
```

- Nhập mã pin vừa tìm được lúc nãy và ta nhận được flag.



⇨ Flag: OnlyAsStrongAsWeakestLink

B. KỊCH BẢN 2

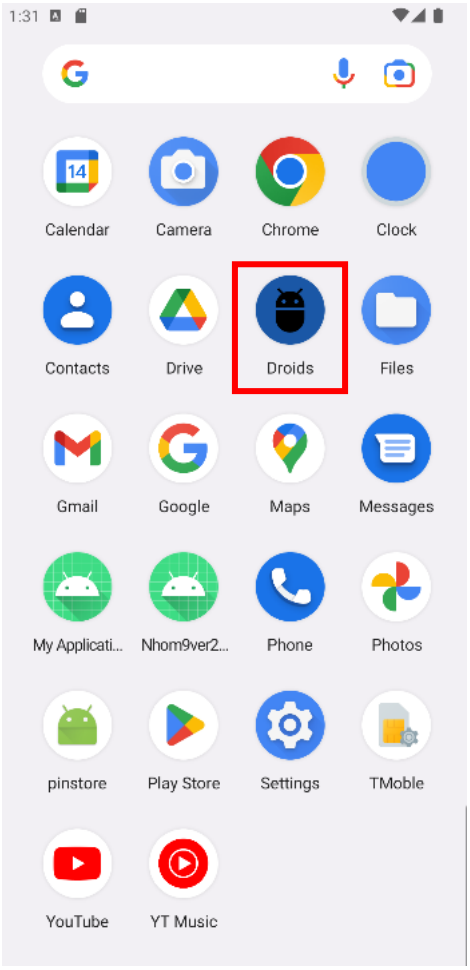
2. Kịch bản 02. Thực hiện phân tích tập tin ứng dụng thu được.

- 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{....}

- Tương tự như kịch bản trên, đầu tiên, nhóm sẽ tải ứng dụng lên thiết bị android.

```
PS C:\Program Files\Genymobile\Genymotion\tools> .\adb devices
List of devices attached
emulator-5554   device

PS C:\Program Files\Genymobile\Genymotion\tools> .\adb install "E:\semester7\Phap_chung_ky_thuat_so\Lab\Lab05-Mobile-Forensics\RES_mobile-forensics\resources-session05\kb02_zha.apk"
Performing Streamed Install
Success
PS C:\Program Files\Genymobile\Genymotion\tools>
```

- Ứng dụng sao khi được tải có tên là Droids.



- Màn hình chính của ứng dụng như bên dưới. Hiện tại app chỉ có mỗi tấm ảnh và không có tính năng gì để tương tác.

- Xem nội dung tập tin AndroidManifest.xml

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.blink"
    platformBuildVersionCode="1"
    platformBuildVersionName="1">
    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="27"/>
    <application
        android:theme="@style/AppTheme"
        android:label="@string/app_name"
        android:icon="@mipmap/ic_launcher"
        android:debuggable="true"
        android:allowBackup="true"
        android:supportsRtl="true"
        android:roundIcon="@mipmap/ic_launcher_round">
        <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>
    </application>
</manifest>
```

⇨ AndroidManifest.xml có hai activity chính là MainActivity và r2d2.

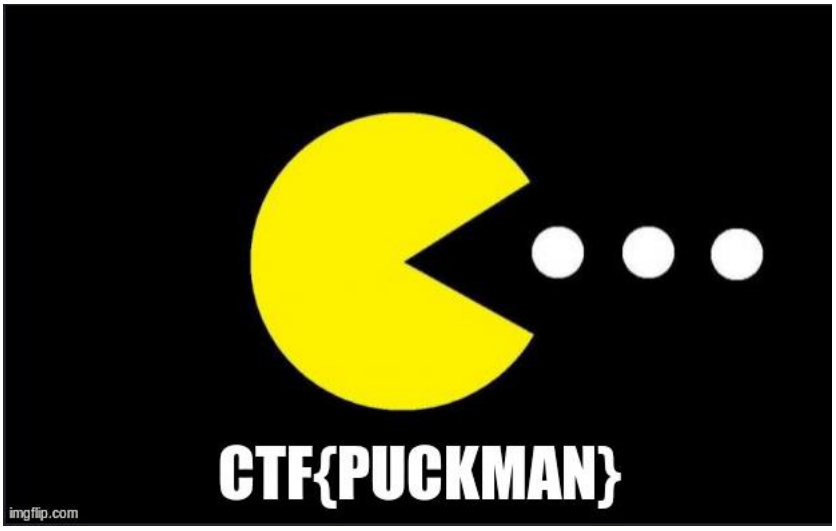
- Khi mở ứng dụng lên, MainActivity sẽ được mặc định hiển thị trước tiên, do đó, tấm hình bên trên khi chúng ta vừa vào ứng dụng chắc chắn là của MainActivity. Vì vậy, class này chẳng có gì để khai thác.
- Thử tìm hiểu class r2d2.

```
package com.example.blink;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Base64;
import android.widget.ImageView;

/* loaded from: classes.dex */
public class r2d2 extends AppCompatActivity {
    @Override // android.support.v7.app.AppCompatActivity, android.support.v4.app.FragmentActivity, android.support.v4.app.SupportActivity, android.app.Activity
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_r2d2);
        ImageView image = (ImageView) findViewById(R.id.imageView);
        String imageString = " 
        byte[] imageBytes = Base64.decode(imageString, 0);
        Bitmap decodedImage = BitmapFactory.decodeByteArray(imageBytes, 0, imageBytes.length);
        image.setImageBitmap(decodedImage);
    }
}
```

- Bên trong đoạn code có chứa một tấm ảnh, tuy nhiên là chuỗi trên đã bị encode bởi base64. Tiến hành export chuỗi trên và đưa vào decode (chuỗi bắt đầu từ "/9j/...")

```
┌──(ngoc㉿ngoc)-[~/Phap_chung/Lab5]
└─$ base64 -d r2d2.txt > image.jpeg
```

- Kết quả nhận được.



⇨ Flag: CTF{PUCKMAN}