APKrypt

Description

- Can you become a VIP.

Objective

- `Reverse engineer the APK file and decrypt the flag.`

Difficulty

- `Easy`

Flag

- `HTB{3nj0y_y0ur_v1p_subscr1pt1on}`

Release:

- /release/APKrypt.zip (`b9913b674cb4a4977fa20398ce55aa64435b41cf7b1f306cc8b2df27a376c213`)

Notes

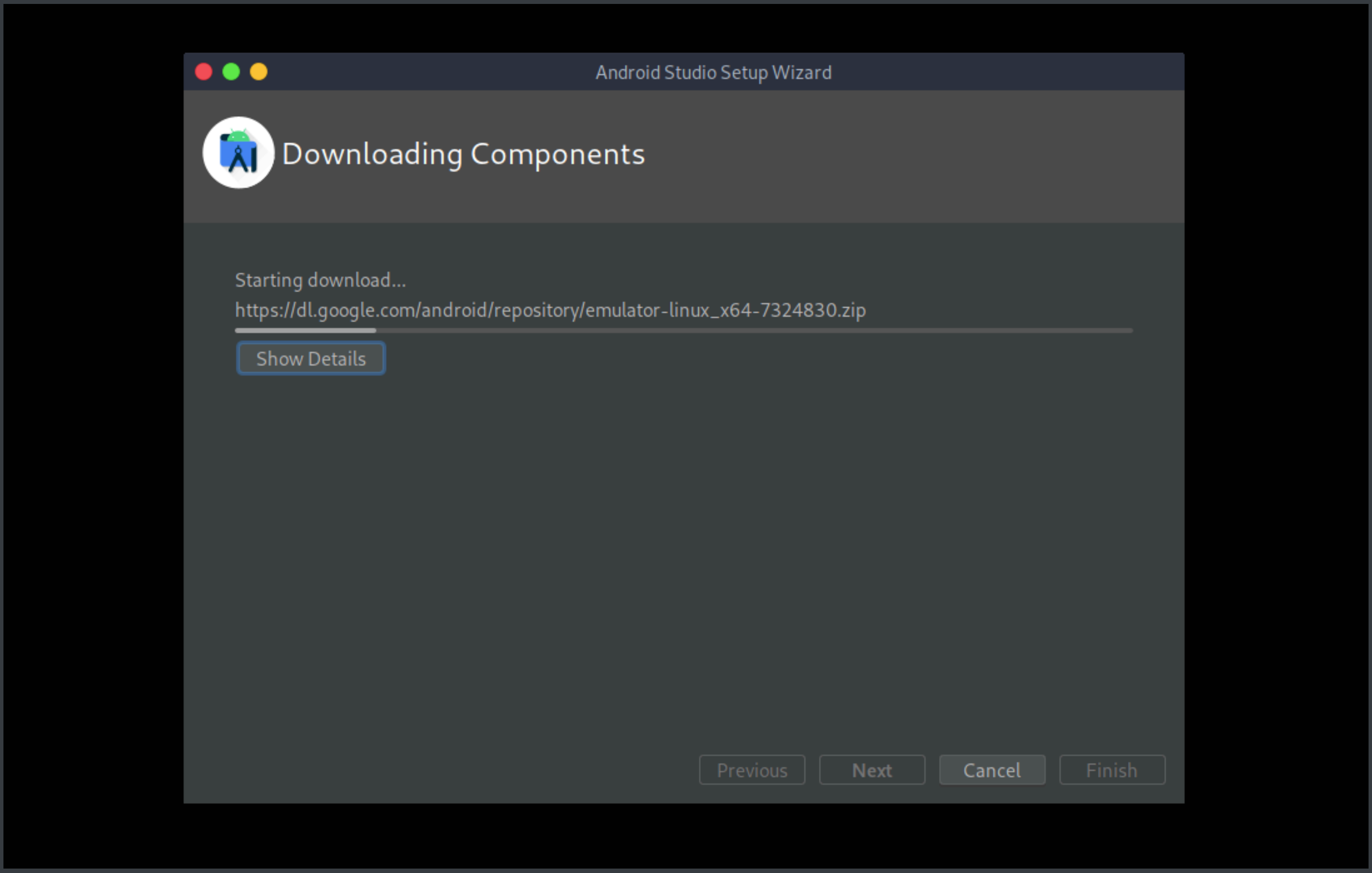
Android Emulator will perform much better on a native operating system (not a virtual machine).

Challenge

Unzipping the `APKrypt.zip` file reveals the file `APKrypt.apk` . In order to run the `APKey.apk` file, we have to set up an Android emulator. To achieve this, we are going to use Android Studio IDE.

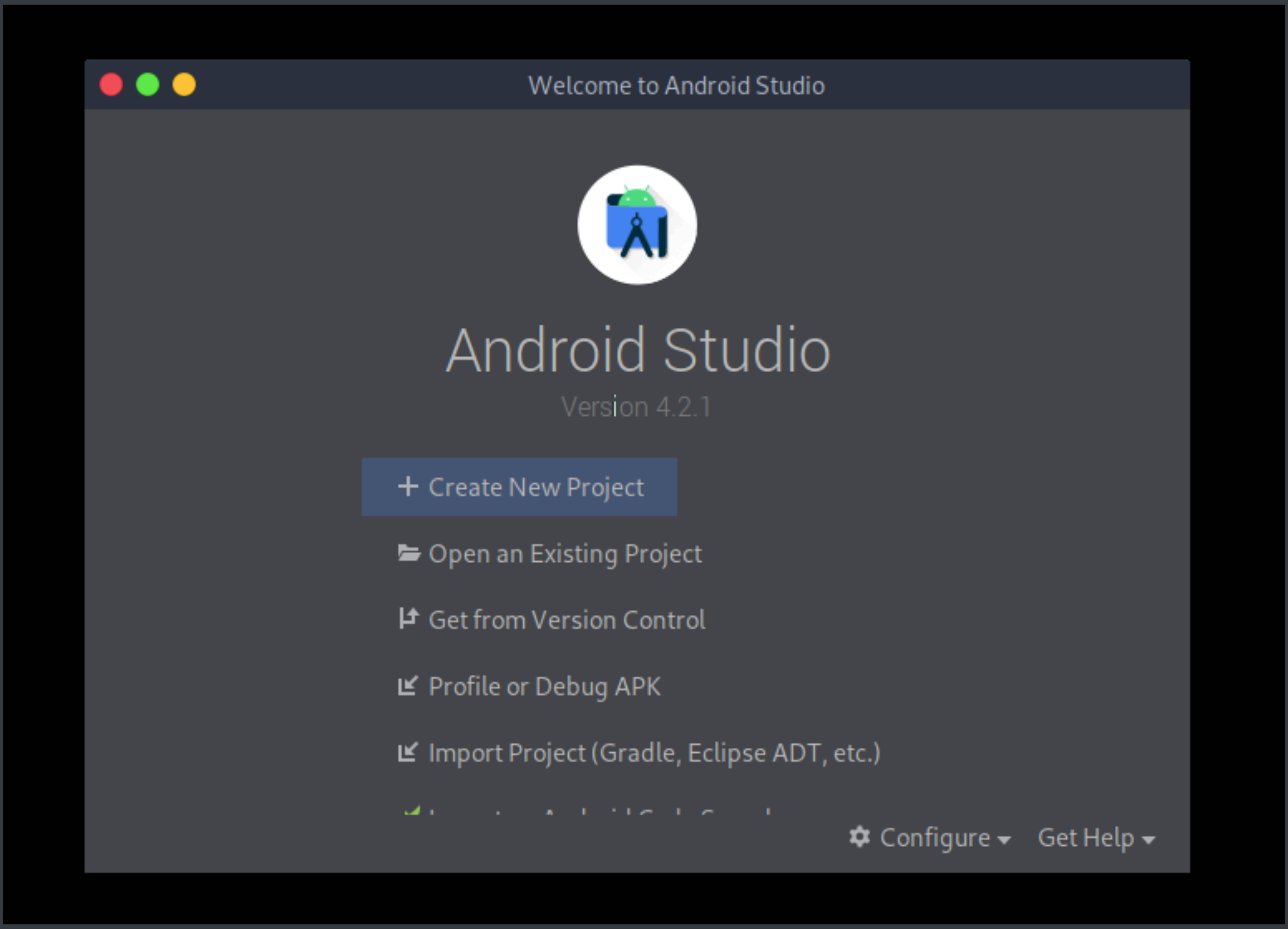
```
wget https://redirector.gvt1.com/edgedl/android/studio/ide-
zips/4.2.1.0/android-studio-ide-202.7351085-linux.tar.gz
tar xvzf android-studio-ide-202.7351085-linux.tar.gz
sh android-studio/bin/studio.sh
```

On the setup wizard we click `OK`, then we click on `Next`, and finally click on `Finish`. Next, we wait for the Android Studio to download the components.

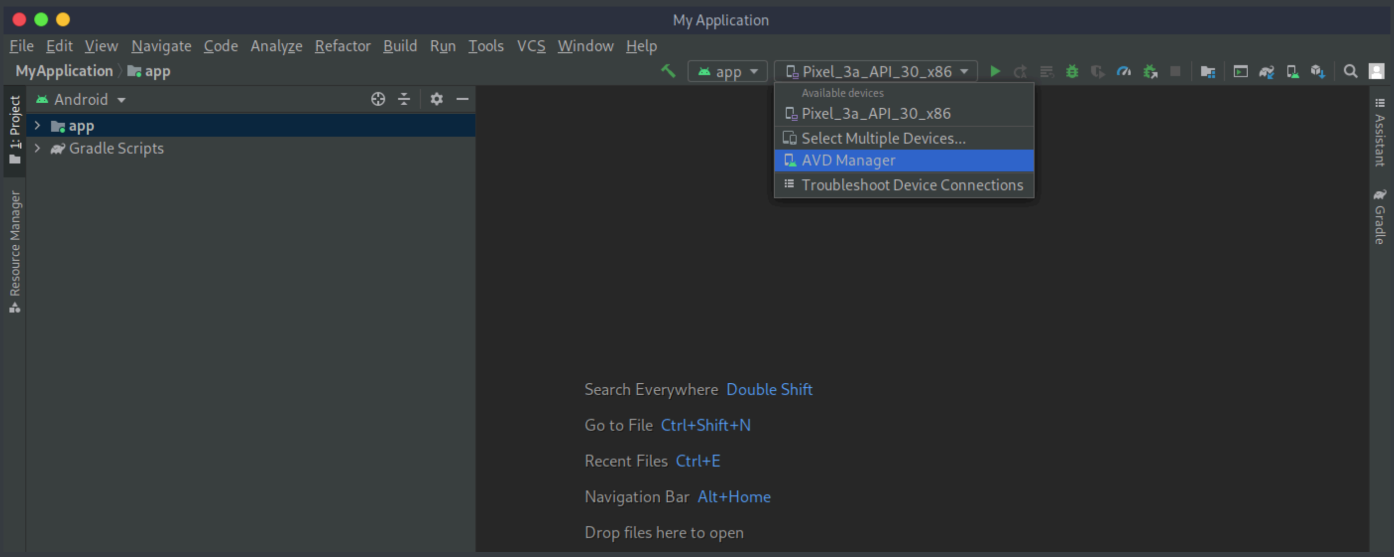


Starting download...
https://dl.google.com/android/repository/emulator-linux_x64-7324830.zip

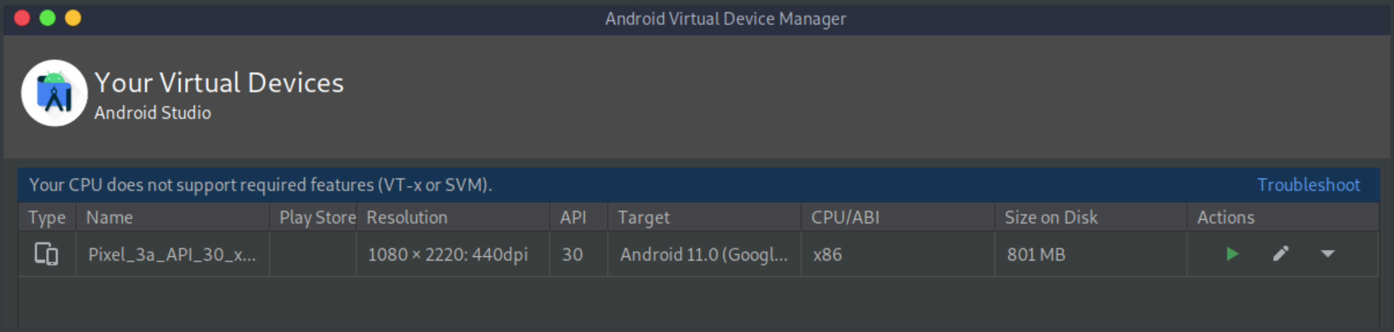
Once it's done, we click `Finish` once again.



Then we click `Next` and finally we click on `Finish`. Now that we have create a new project, we wait for some more files to get downloaded automatically from the IDE. When that's done, click on the top centre of the IDE and select `AVD Manager`.



On the AVD Manager menu, click on the green "play" button to start the emulator.



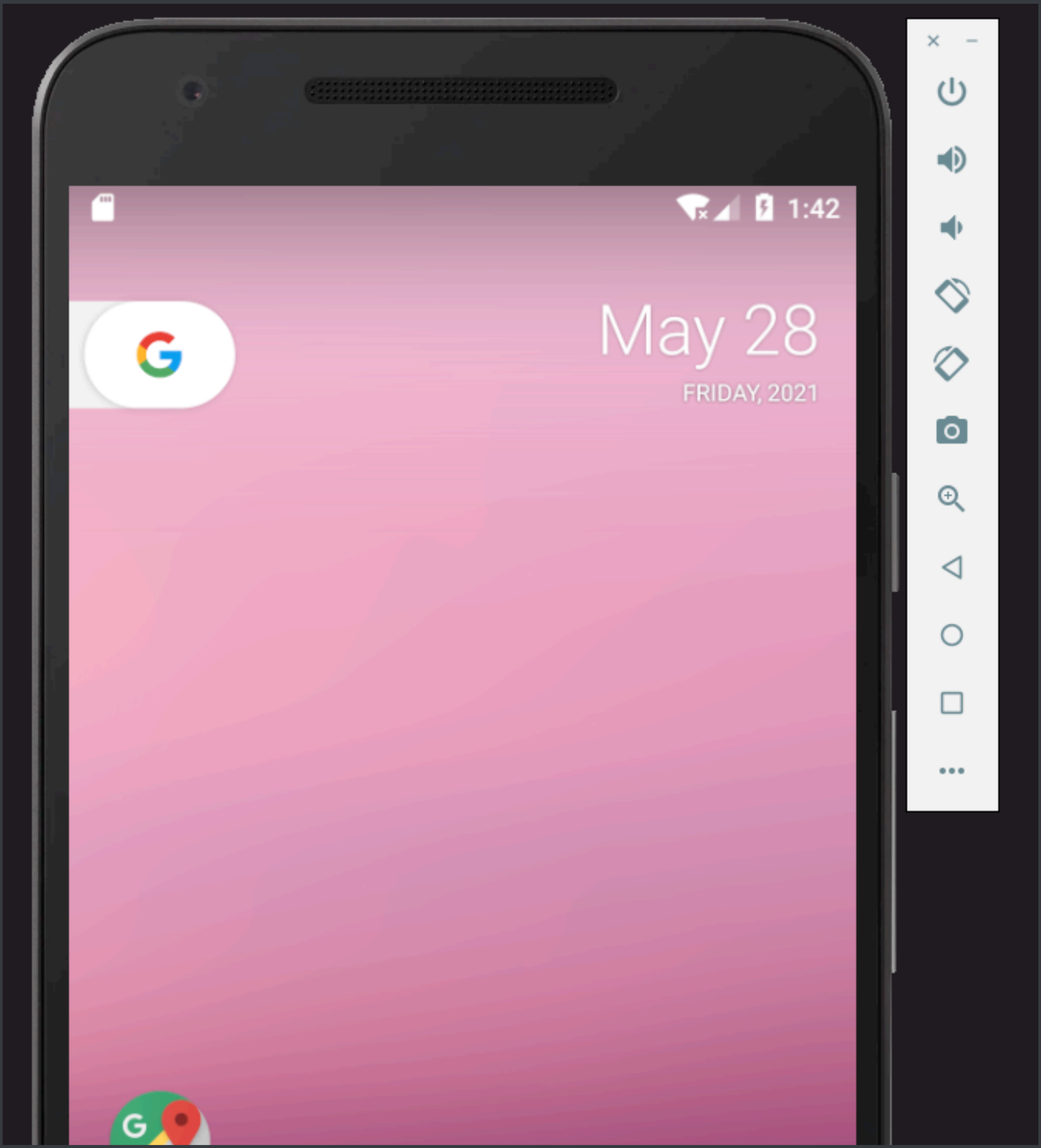
Android Virtual Device Manager

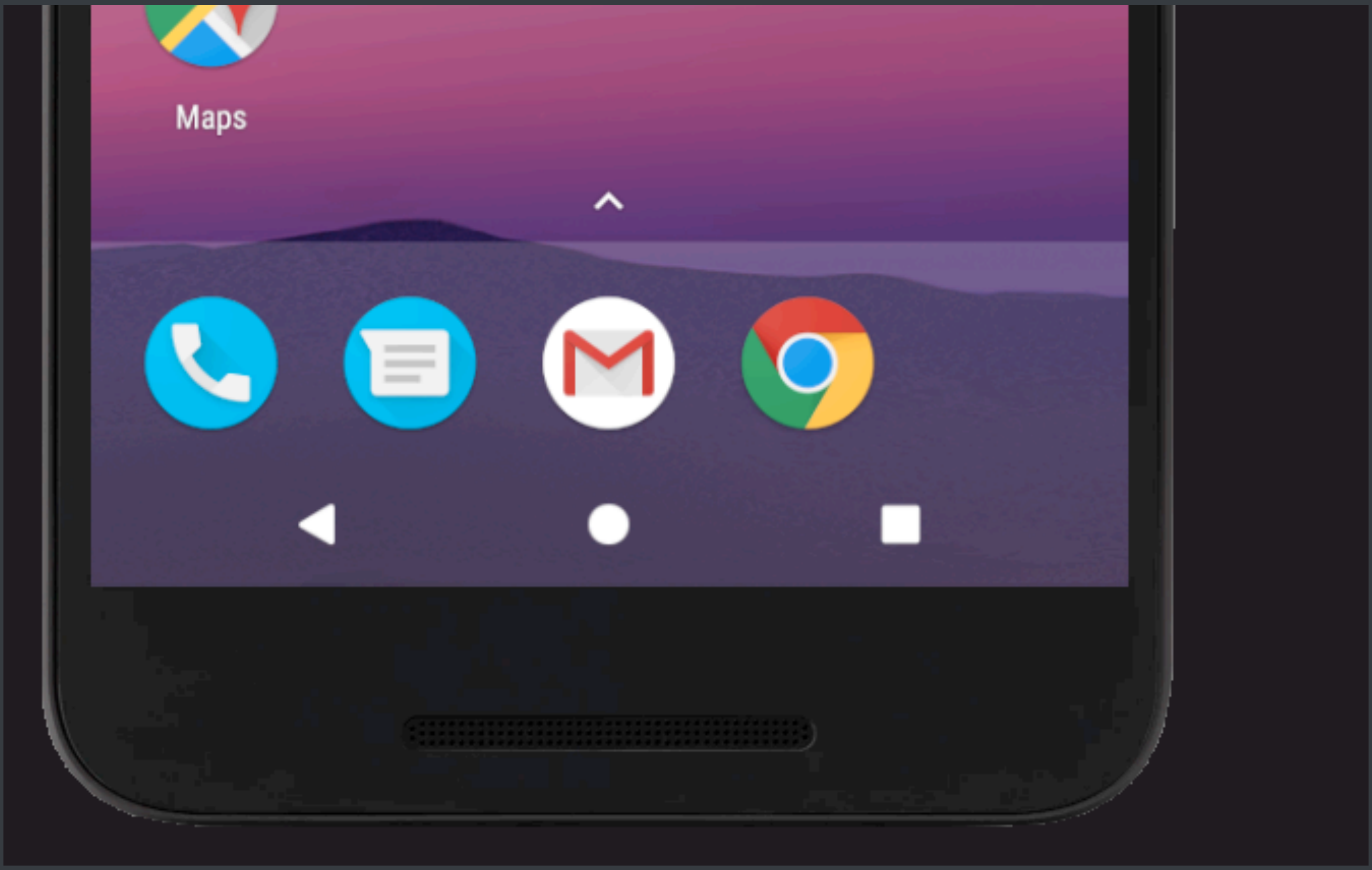
Your Virtual Devices
Android Studio

Your CPU does not support required features (VT-x or SVM). Troubleshoot

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel_3a_API_30_x...		1080 × 2220: 440dpi	30	Android 11.0 (Googl...	x86	801 MB	

Once the device is started, It should be looking like this.



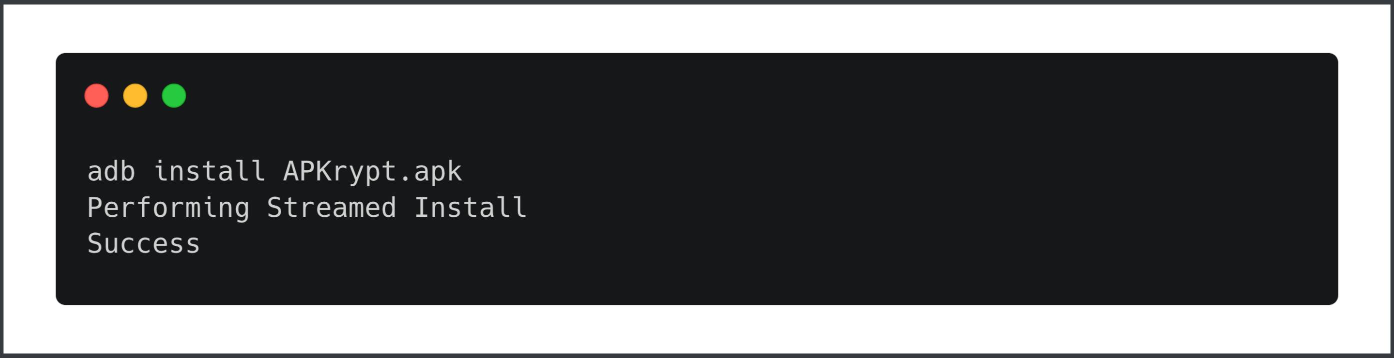


Then, we install `adb` so we can communicate with it.

```
sudo apt-get install adb
```

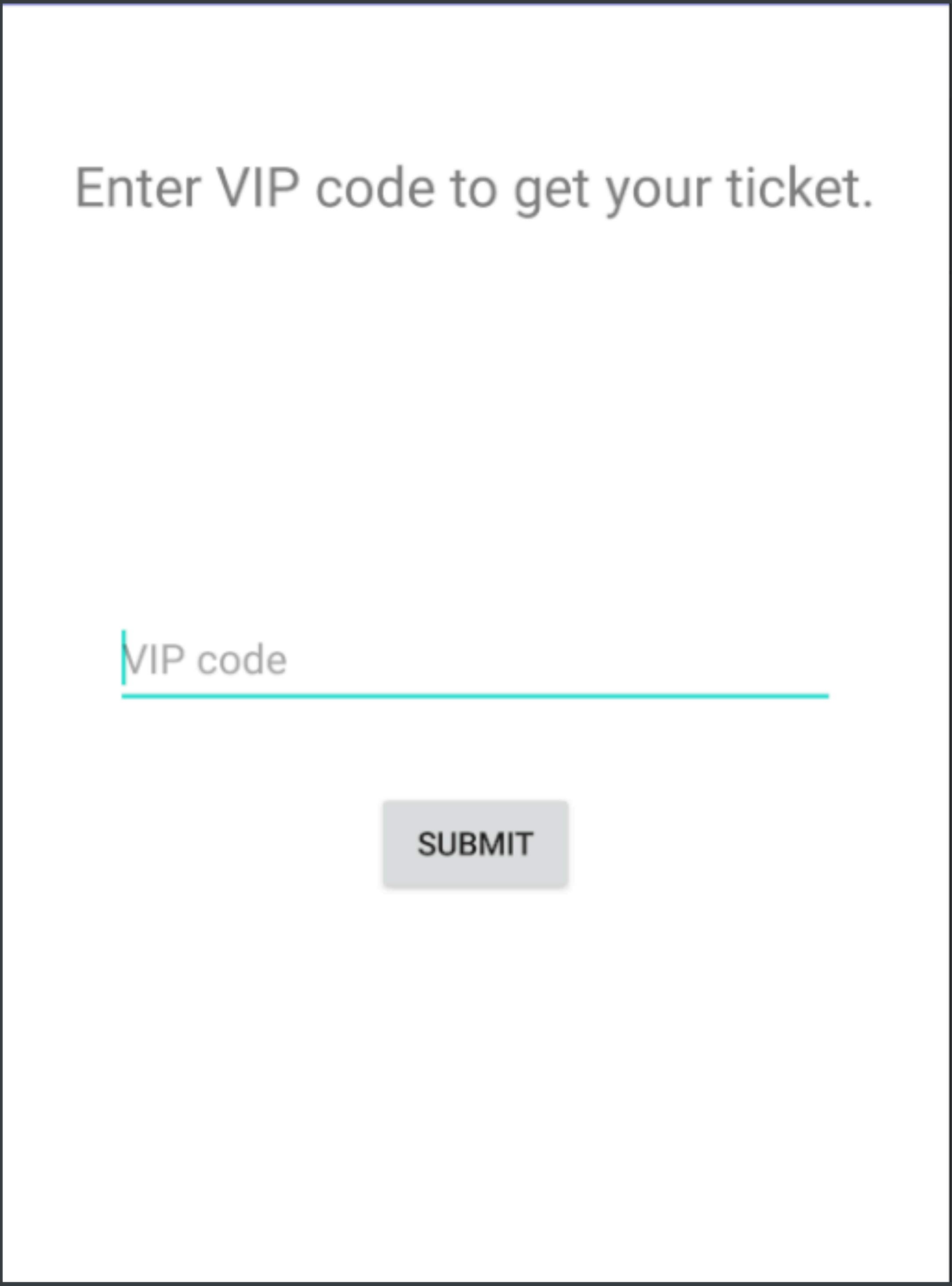
While the device is running, we can execute the following command to install the application on the device.

```
adb install APKrypt.apk
```



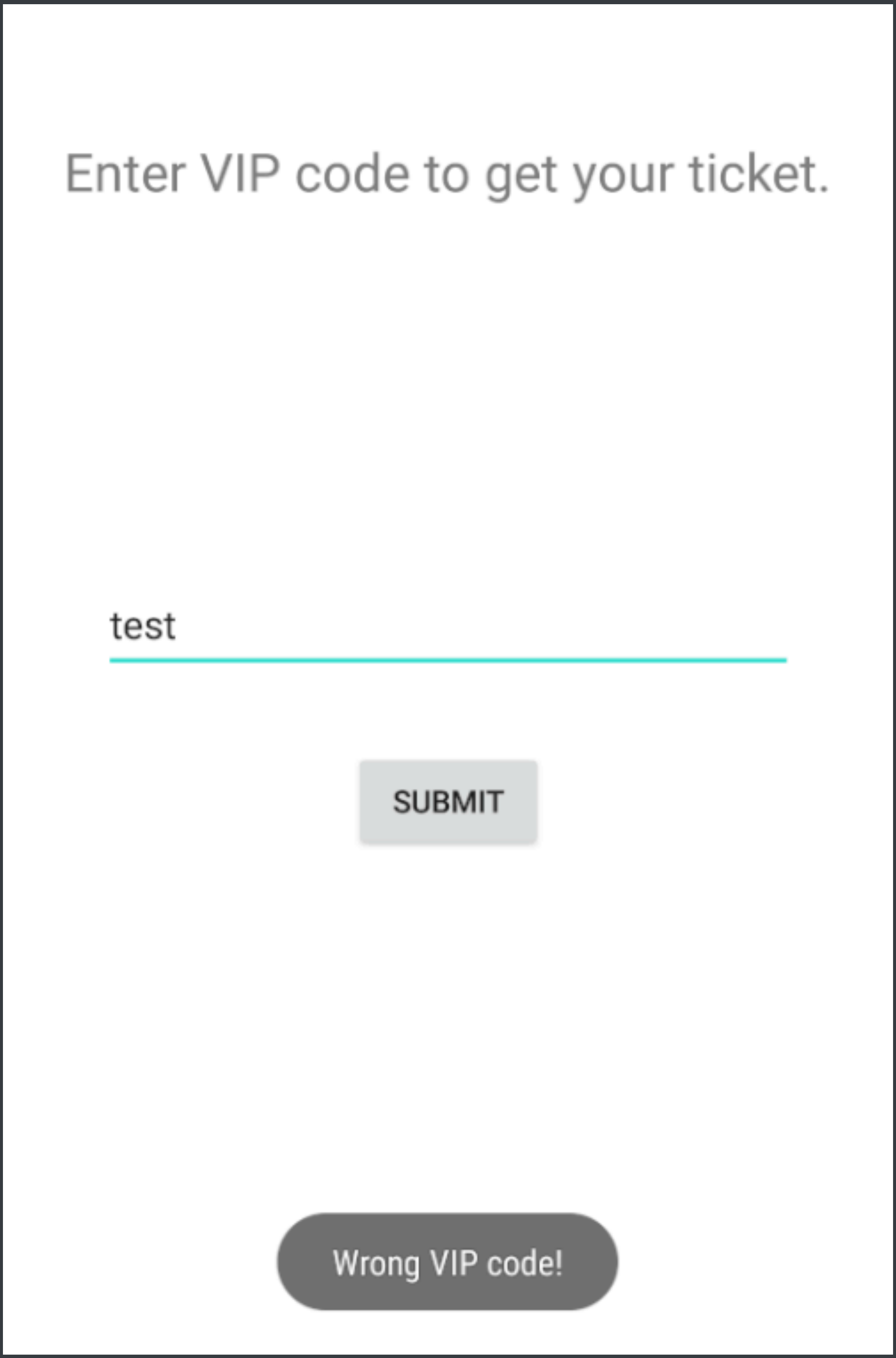
```
adb install APKrypt.apk
Performing Streamed Install
Success
```

Finally, from the device, we can locate and start application we just installed.



Enter VIP code to get your ticket.

This is an application featuring a system that issues VIP tickets. Let's put a random code to see the output.


Enter VIP code to get your ticket.

test

Wrong VIP code!

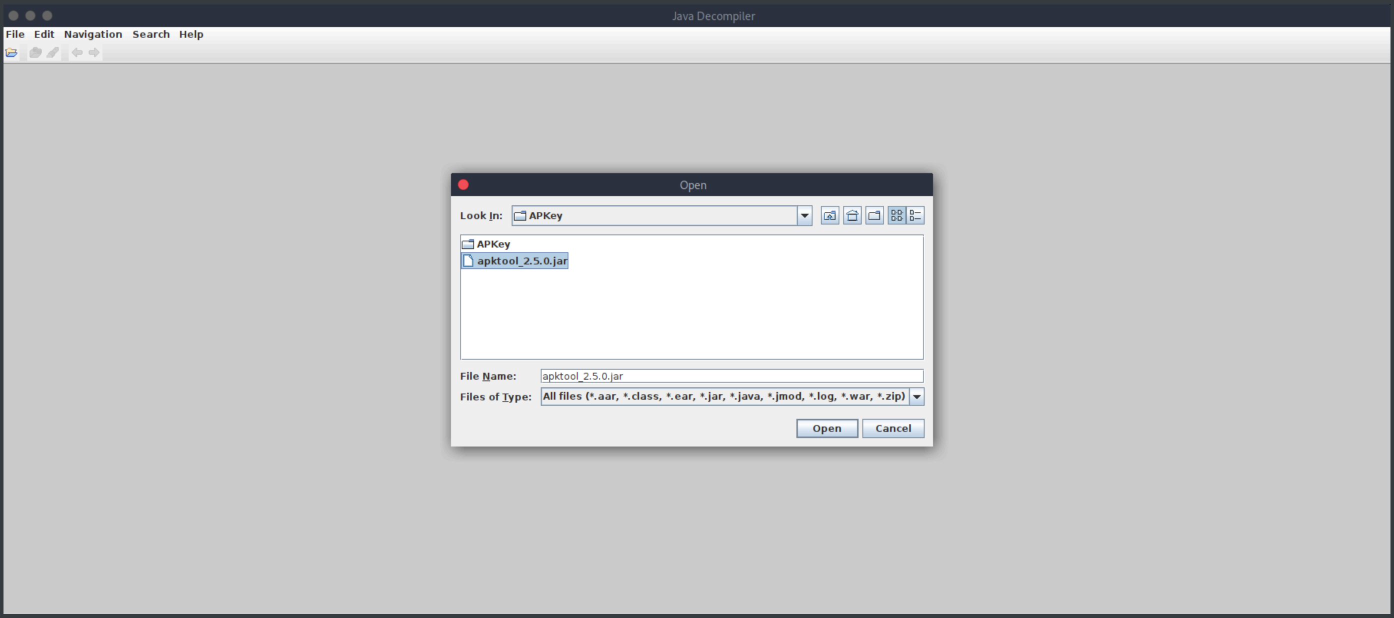
The output is `Wrong VIP code!`. Let's reverse the APK file. Using `d2j-dex2jar` we can create a JAR file, and then using JD-GUI we can read the source code of the APK file.

```
sudo apt-get install dex2jar
sudo apt-get install jd-gui
```

Finally, we run the following.

```
d2j-dex2jar APKrypt.apk
jd-gui
```

On the top left we choose the file icon and we select the JAR file we just created. Then we click `Open`.



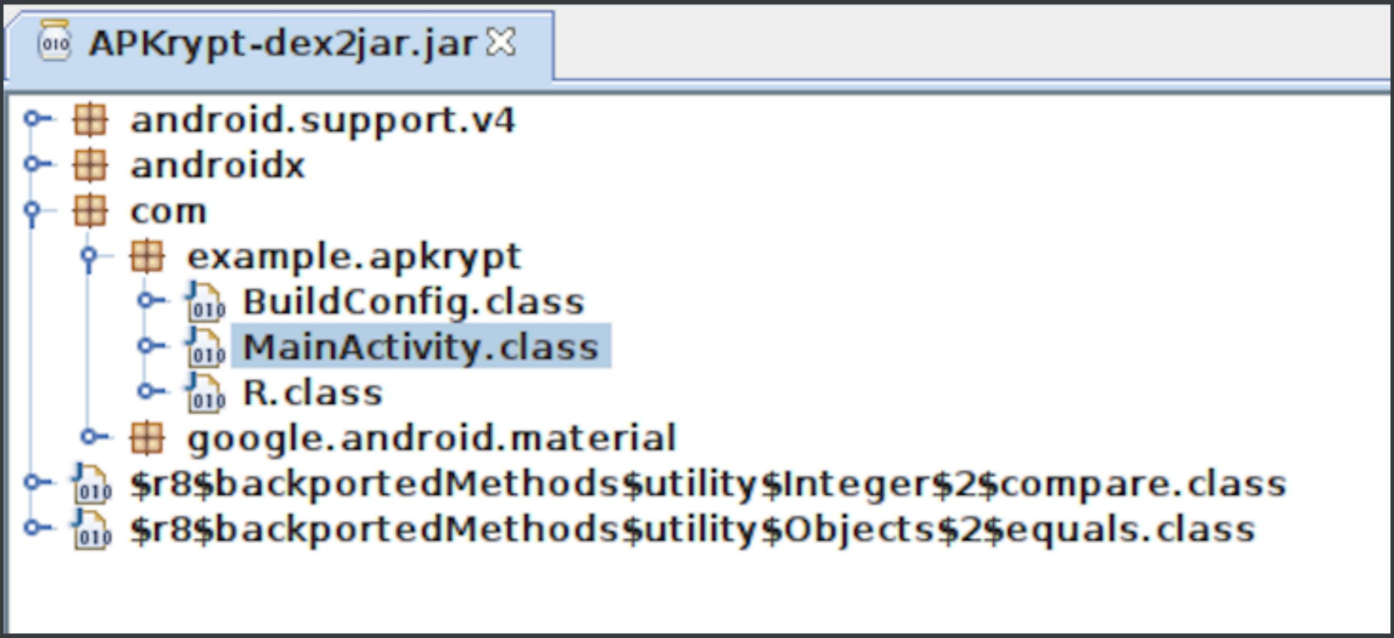
Look In: APKey

APKey
apktool_2.5.0.jar

File Name: apktool_2.5.0.jar

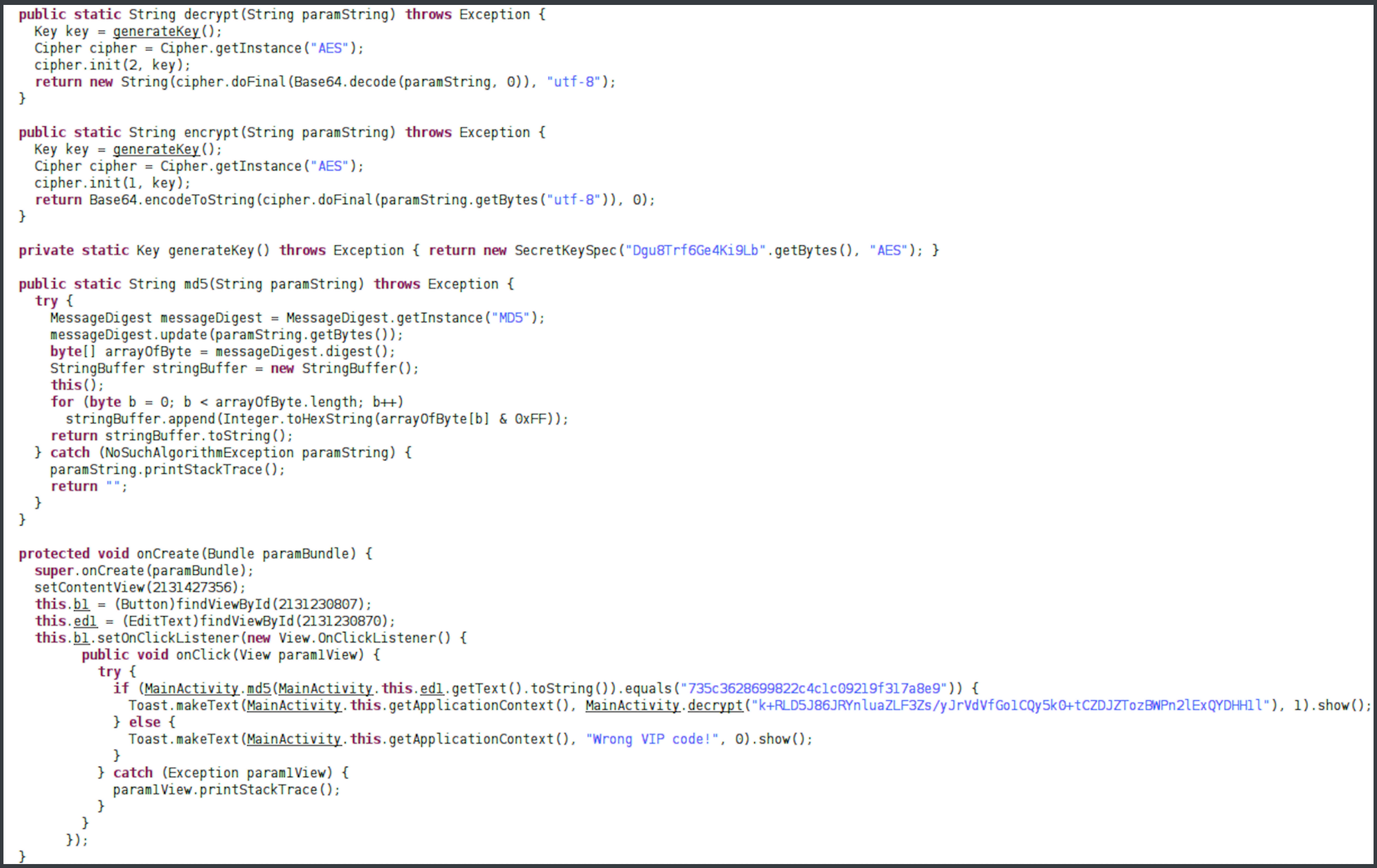
Files of Type: All files (*.aar, *.class, *.ear, *.jar, *.java, *.jmod, *.log, *.war, *.zip)

Let's read the source code in the `MainActivity.class`.



```
android.support.v4
androidx
com
    example.apkrypt
        BuildConfig.class
        MainActivity.class
        R.class
    google.android.material
$r8$backportedMethods$utility$Integer$2$compare.class
$r8$backportedMethods$utility$Objects$2$equals.class
```

Reading the source code, we conclude that the VIP code (flag) is encrypted using AES.

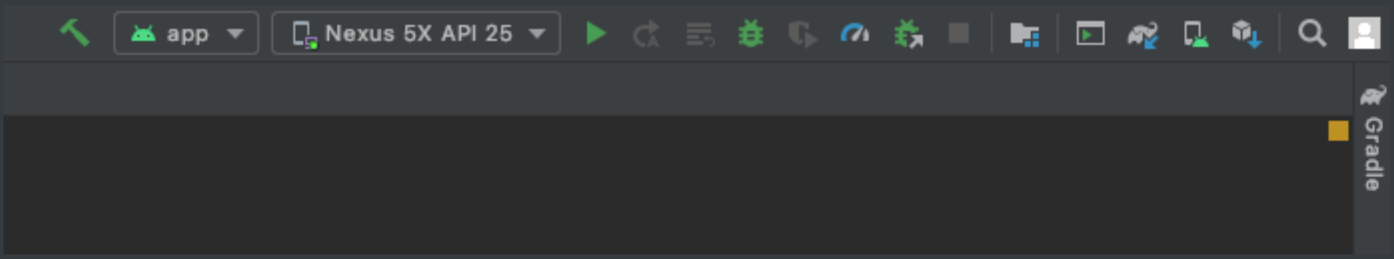


```
public static String decrypt(String paramString) throws Exception {
  Key key = generateKey();
  Cipher cipher = Cipher.getInstance("AES");
  cipher.init(2, key);
  return new String(cipher.doFinal(Base64.decode(paramString, 0)), "utf-8");
}

public static String encrypt(String paramString) throws Exception {
  Key key = generateKey();
  Cipher cipher = Cipher.getInstance("AES");
  cipher.init(1, key);
  return Base64.encodeToString(cipher.doFinal(paramString.getBytes("utf-8")), 0);
}

private static Key generateKey() throws Exception { return new SecretKeySpec("Dgu8Trf6Ge4Ki9Lb".getBytes(), "AES"); }

public static String md5(String paramString) throws Exception {
  try {
    MessageDigest messageDigest = MessageDigest.getInstance("MD5");
    messageDigest.update(paramString.getBytes());
    byte[] arrayOfByte = messageDigest.digest();
    StringBuffer stringBuffer = new StringBuffer();
    this();
    for (byte b = 0; b < arrayOfByte.length; b++)
      stringBuffer.append(Integer.toHexString(arrayOfByte[b] & 0xFF));
    return stringBuffer.toString();
  } catch (NoSuchAlgorithmException paramString) {
    paramString.printStackTrace();
    return "";
  }
}

protected void onCreate(Bundle paramBundle) {
  super.onCreate(paramBundle);
  setContentView(2131427356);
  this.b1 = (Button)findViewById(2131230807);
  this.ed1 = (EditText)findViewById(2131230870);
  this.b1.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1View) {
          try {
            if (MainActivity.md5(MainActivity.this.ed1.getText().toString()).equals("735c3628699822c4c1c09219f317a8e9")) {
              Toast.makeText(MainActivity.this.getApplicationContext(), MainActivity.decrypt("k+RLD5J86JRYnluaZLF3Zs/yJrVdVfGo1CQy5k0+tCZDJZTozBWPn2lExQYDHH1l"), 1).show();
            } else {
              Toast.makeText(MainActivity.this.getApplicationContext(), "Wrong VIP code!", 0).show();
            }
          } catch (Exception param1View) {
            param1View.printStackTrace();
          }
        }
      });
}
```

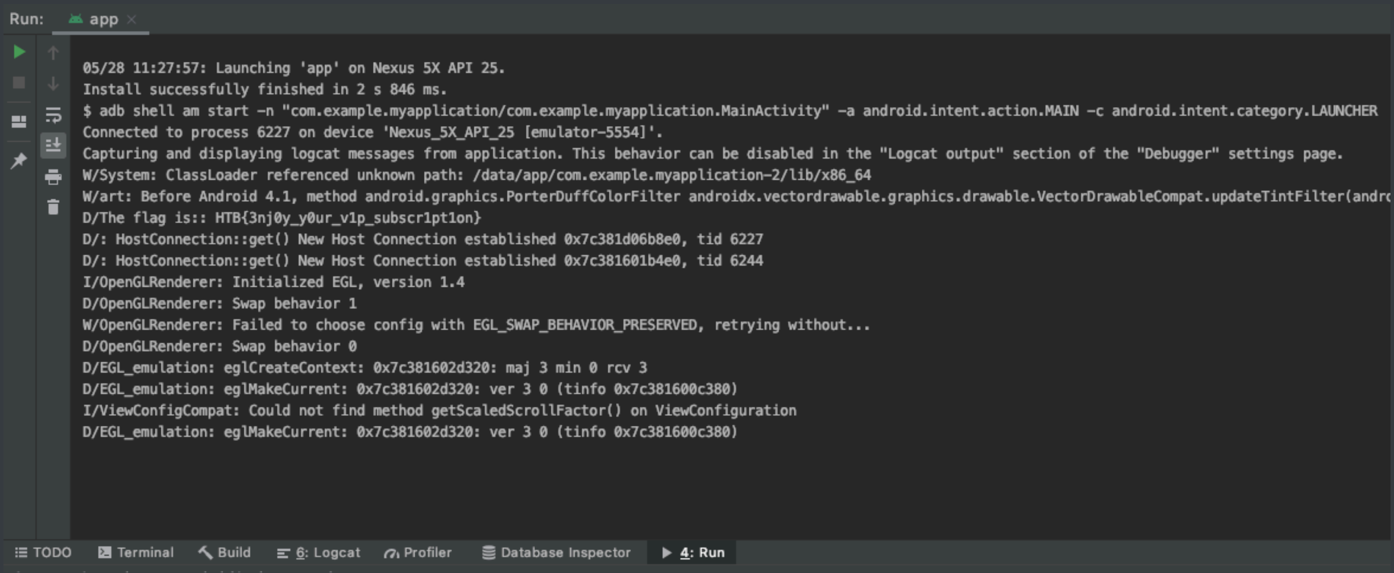
In the `MainActivity.java` of the project we created earlier on android studio, we add the following code to decrypt the flag, using the secrete key `Dgu8Trf6Ge4Ki9Lb` that is shown above.

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        try {
            decrypt();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

        public static void decrypt() throws Exception {
            Key key = generateKey();
            Cipher cipher = Cipher.getInstance("AES");
            cipher.init(Cipher.DECRYPT_MODE, key);
            byte[] decryptedValue64 =
Base64.decode("k+RLD5J86JRYnluaZLF3Zs/yJrVdVfGo1CQy5k0+tCZDJZTozBWPn2l
ExQYDHH1l", Base64.DEFAULT);
            byte [] decryptedByteValue =
cipher.doFinal(decryptedValue64);
            String decryptedValue = new
String(decryptedByteValue,"utf-8");

            Log.d("The flag is: ", decryptedValue);
        }

        private static Key generateKey() throws Exception {
            Key key = new SecretKeySpec("Dgu8Trf6Ge4Ki9Lb".getBytes(),
"AES");
            return key;
        }
}
```

On the top right, we click on the green "play" button to start the application.



On the run tab, we can see the output of the execution.



```
05/28 11:27:57: Launching 'app' on Nexus 5X API 25.
Install successfully finished in 2 s 846 ms.
$ adb shell am start -n "com.example.myapplication/com.example.myapplication.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
Connected to process 6227 on device 'Nexus_5X_API_25 [emulator-5554]'.
Capturing and displaying logcat messages from application. This behavior can be disabled in the "Logcat output" section of the "Debugger" settings page.
W/System: ClassLoader referenced unknown path: /data/app/com.example.myapplication-2/lib/x86_64
W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter androidx.vectordrawable.graphics.drawable.VectorDrawableCompat.updateTintFilter(andro
D/The flag is:: HTB{3nj0y_y0ur_v1p_subscr1pt1on}
D/: HostConnection::get() New Host Connection established 0x7c381d06b8e0, tid 6227
D/: HostConnection::get() New Host Connection established 0x7c381601b4e0, tid 6244
I/OpenGLRenderer: Initialized EGL, version 1.4
D/OpenGLRenderer: Swap behavior 1
W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
D/OpenGLRenderer: Swap behavior 0
D/EGL_emulation: eglCreateContext: 0x7c381602d320: maj 3 min 0 rcv 3
D/EGL_emulation: eglMakeCurrent: 0x7c381602d320: ver 3 0 (tinfo 0x7c381600c380)
I/ViewConfigCompat: Could not find method getScaledScrollFactor() on ViewConfiguration
D/EGL_emulation: eglMakeCurrent: 0x7c381602d320: ver 3 0 (tinfo 0x7c381600c380)
```

The flag has been decrypted and printed successfully.