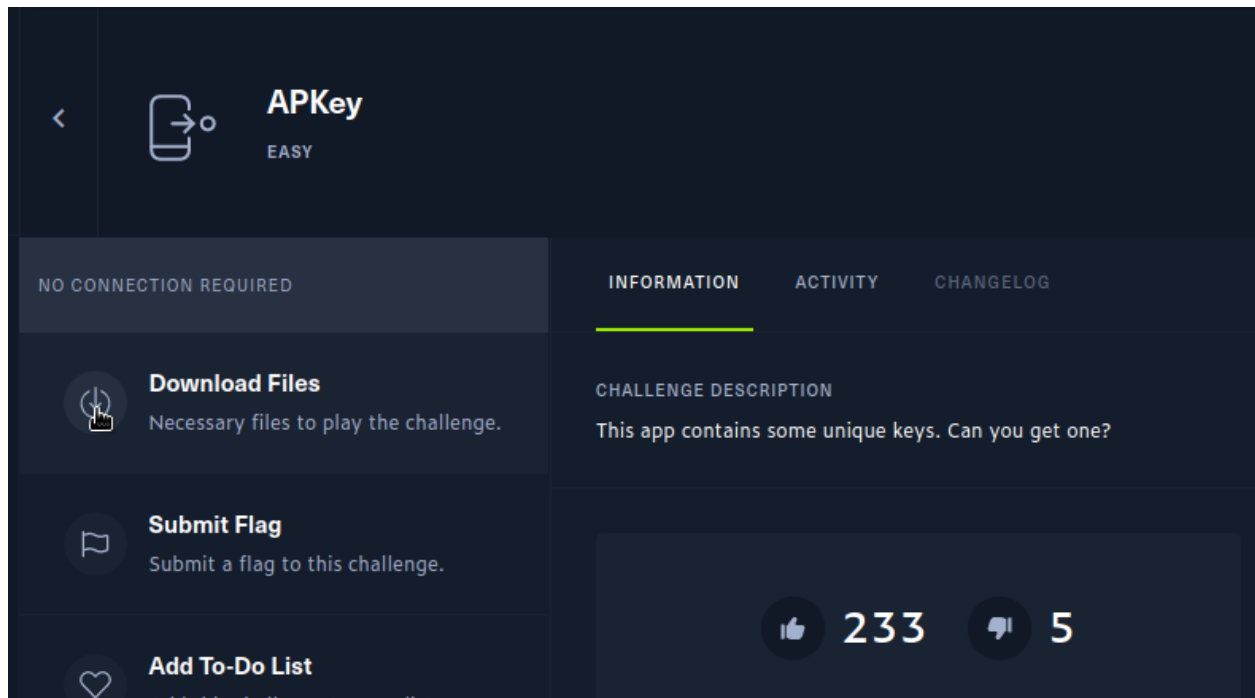
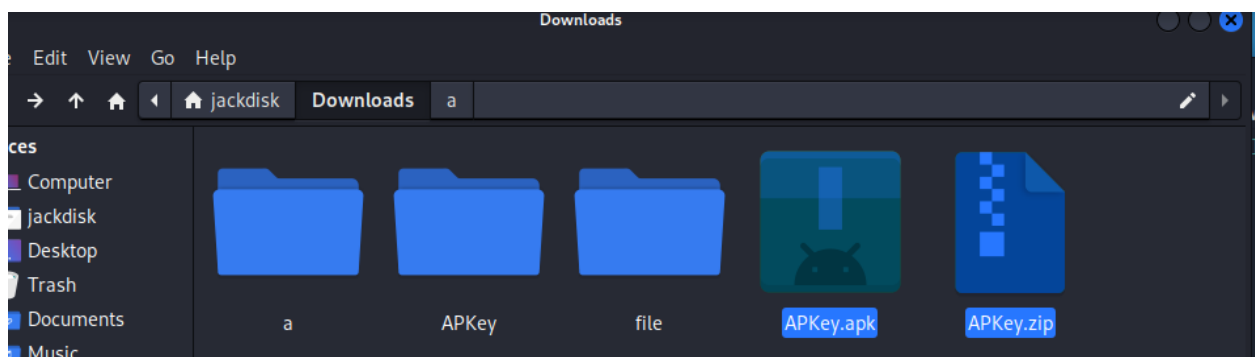


This is a challenge of Hack The Box mobile session. Whose name is APKey. Let's get started, Firstly, we have to download some files.



After downloading we have to extract that zip file. So after that we will get another file named `APKey.apk`.



Then we have to open the terminal and then go to that file where it was extracted.

We can make some changes and can use **apktool**. Let's run

```
apktool d APkey.apk
```

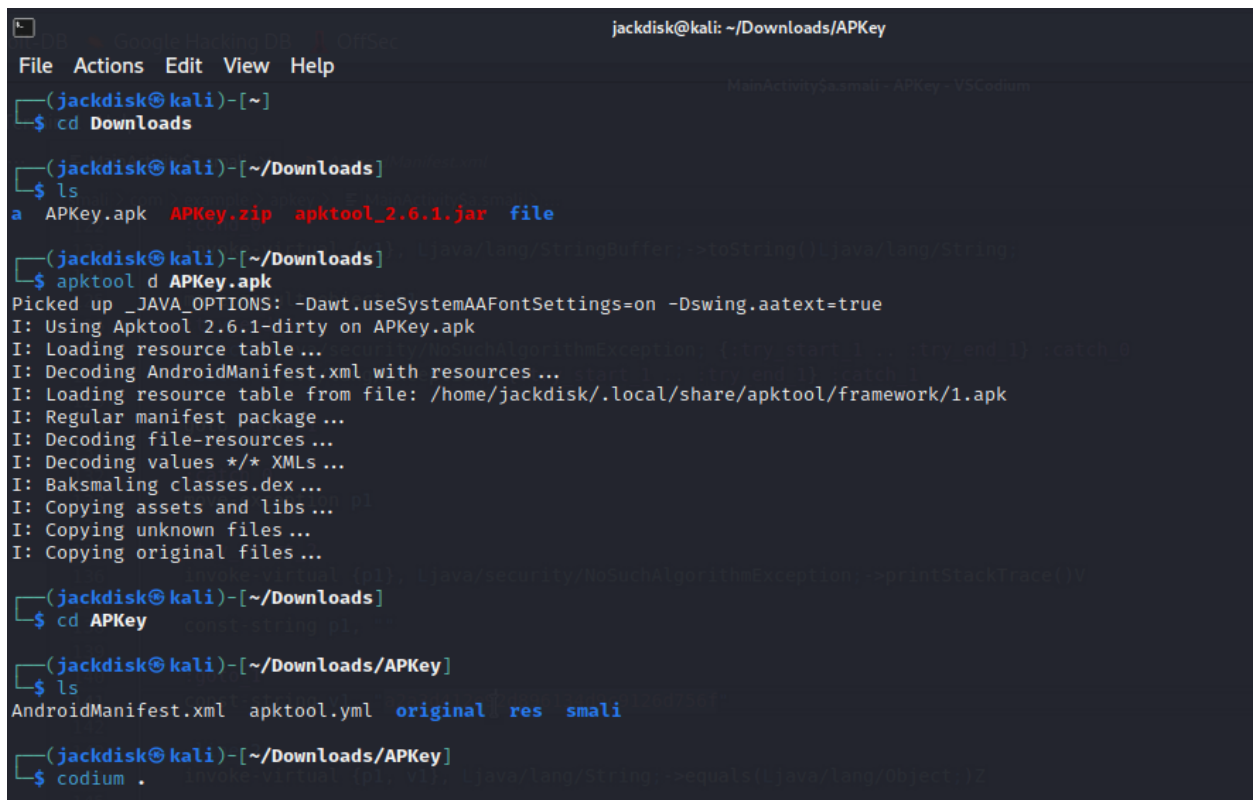
Here d for decode and we can retrieving all of the file from apk, here we've got APKey folder

We have to go to the **APKey** folder.

```
cd APKey
```

We've got all these smali files. So we have to open this up in **VSCodium**.

```
Codium .
```



```
jackdisk@kali: ~/Downloads/APKey
File Actions Edit View Help
MainActivity$a.smali - APKey - VSCodeium

(jackdisk@kali)~]
$ cd Downloads

(jackdisk@kali)~[/Downloads]
$ ls
a APKey.apk  APKey.zip  apktool_2.6.1.jar  file

(jackdisk@kali)~[/Downloads]
$ apktool d APKey.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1-dirty on APKey.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /home/jackdisk/.local/share/apktool/framework/1.apk
I: Regular manifest package ...
I: Decoding file-resources ...
I: Decoding values */* XMLs ...
I: Baksmaling classes.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...

(jackdisk@kali)~[/Downloads]
$ cd APKey

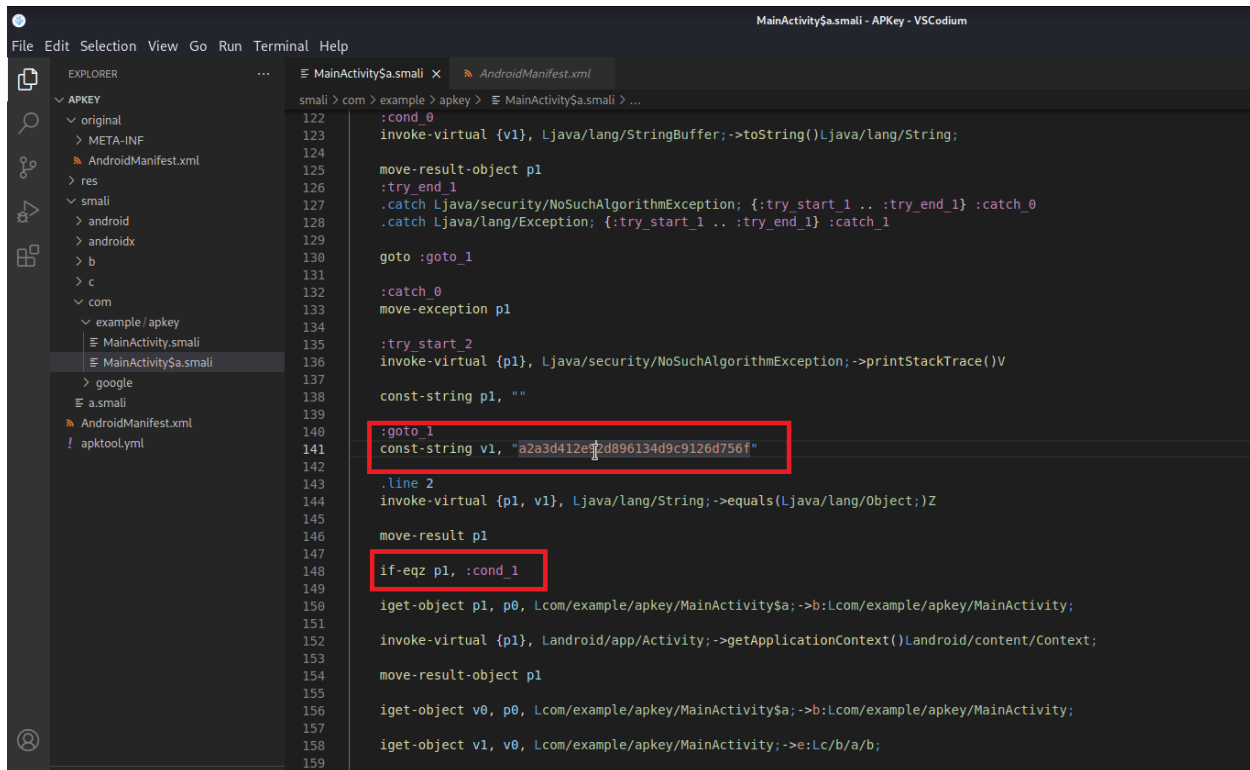
(jackdisk@kali)~[/Downloads/APKey]
$ ls
AndroidManifest.xml  apktool.yml  original  res  smali

(jackdisk@kali)~[/Downloads/APKey]
$ codium .
```

Make necessary modifications on smali files. We are looking out for the **md5** hash because these values here are used to initialize the ase key.

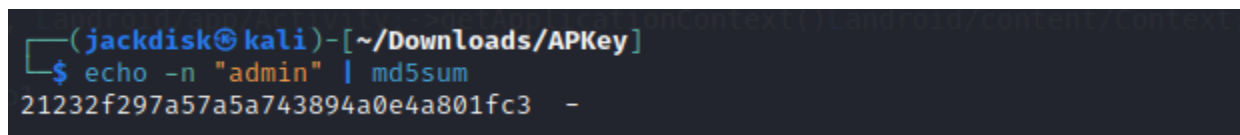
MainActivity\$a.smali file has the hash.

Here we have the string we can see it is checking if equal and then down here we have if equals do conditions zero.



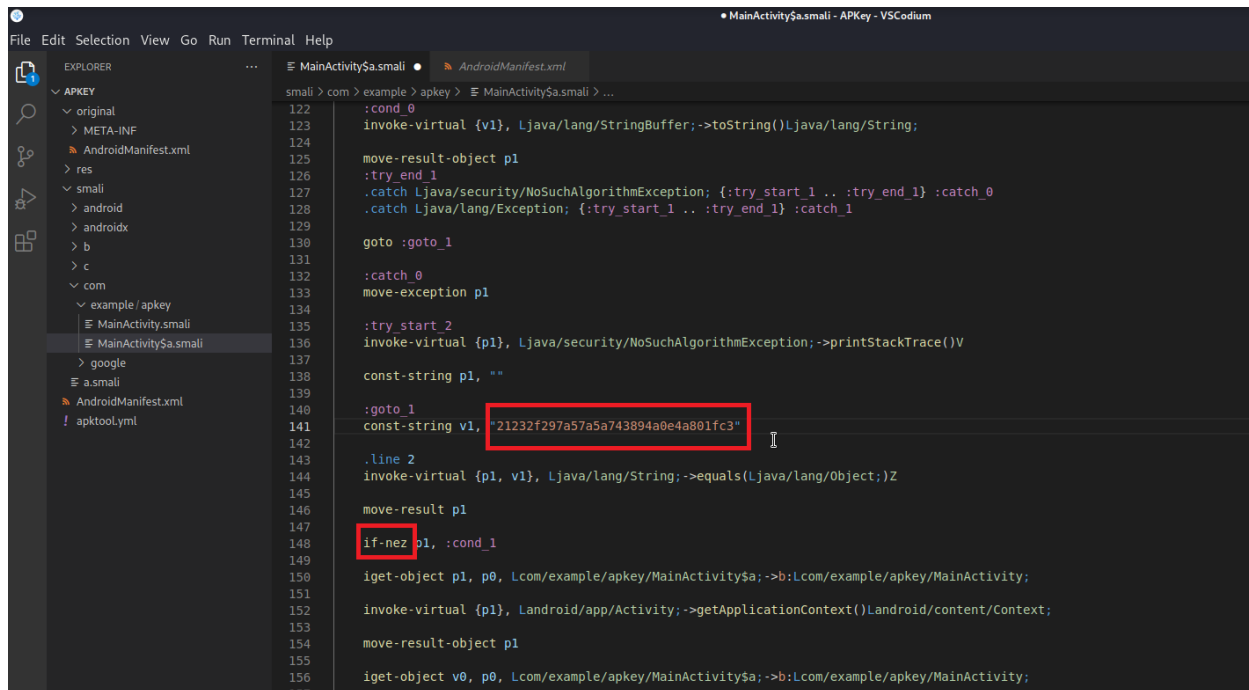
```
122 :cond_0
123 invoke-virtual {v1}, Ljava/lang/StringBuffer;.>toString()Ljava/lang/String;
124
125 move-result-object p1
126 :try_end_1
127 :catch Ljava/security/NoSuchAlgorithmException; {:try_start_1 .. :try_end_1} :catch_0
128 :catch Ljava/lang/Exception; {:try_start_1 .. :try_end_1} :catch_1
129
130 goto :goto_1
131
132 :catch_0
133 move-exception p1
134
135 :try_start_2
136 invoke-virtual {p1}, Ljava/security/NoSuchAlgorithmException;.>printStackTrace()V
137
138 const-string p1, ""
139
140 :goto_1
141 const-string v1, "a2a3d412e32d896134d9c9126d756f"
142
143 :line 2
144 invoke-virtual {p1, v1}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
145
146 move-result p1
147
148 if-eqz p1, :cond_1
149
150 iget-object p1, p0, Lcom/example/apkey/MainActivity$a;.>b:Lcom/example/apkey/MainActivity;
151
152 invoke-virtual {p1}, Landroid/app/Activity;.>getApplicationContext()Landroid/content/Context;
153
154 move-result-object p1
155
156 iget-object v0, p0, Lcom/example/apkey/MainActivity$a;.>b:Lcom/example/apkey/MainActivity;
157
158 iget-object v1, v0, Lcom/example/apkey/MainActivity;.>e:Lc/b/a/b;
```

We could get a md5. we took a copy of that.



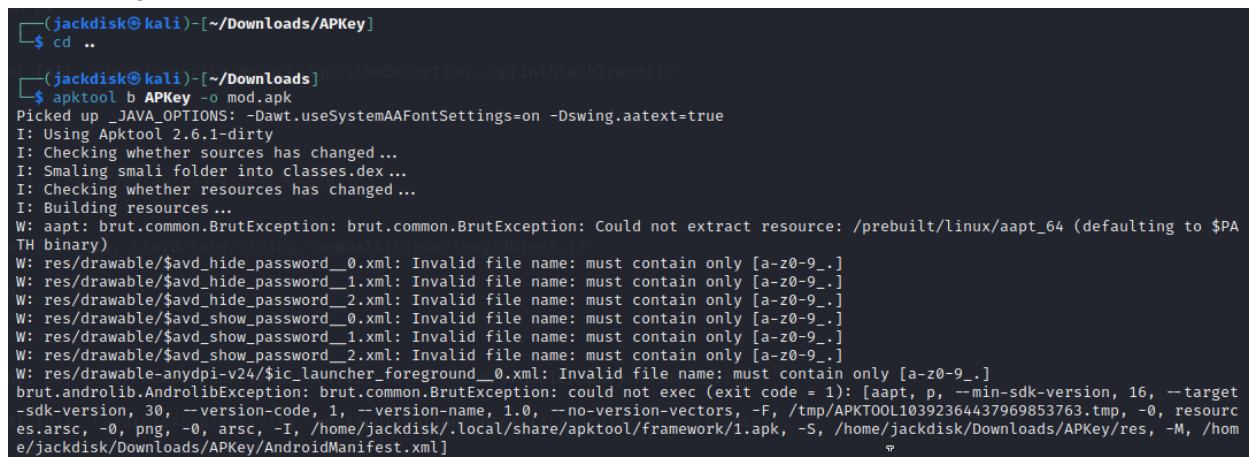
```
(jackdisk@kali)-[~/Downloads/APKey]
$ echo -n "admin" | md5sum
21232f297a57a5a743894a0e4a801fc3 -
```

We took a copy of that so we could paste it on line 141. We have to change some in line 148 eqz to nez . eqz= equal to zero & nez= not equal to zero.



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  APKEY
    original
    META-INF
    AndroidManifest.xml
    res
    smali
      android
      androidx
      b
      c
      com
        example/apkey
          MainActivity.smali
          MainActivity$a.smali
      google
      a.smali
      AndroidManifest.xml
      apktool.yml
  MainActivity$a.smali
    smali > com > example > apkey > MainActivity$a.smali > ...
    122 :cond_0
    123 invoke-virtual {v1}, Ljava/lang/StringBuffer;.>toString()Ljava/lang/String;
    124
    125 move-result-object p1
    126 :try_end_1
    127 :catch Ljava/security/NoSuchAlgorithmException; {:try_start_1 .. :try_end_1} :catch_0
    128 :catch Ljava/lang/Exception; {:try_start_1 .. :try_end_1} :catch_1
    129
    130 goto :goto_1
    131
    132 :catch_0
    133 move-exception p1
    134
    135 :try_start_2
    136 invoke-virtual {p1}, Ljava/security/NoSuchAlgorithmException;.>printStackTrace()V
    137
    138 const-string p1, ""
    139
    140 :goto_1
    141 const-string v1, "21232f297a57a5a743894a0e4a801fc3"
    142
    143 :line 2
    144 invoke-virtual {p1, v1}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
    145
    146 move-result p1
    147
    148 if-nez p1, :cond_1
    149
    150 iget-object p1, p0, Lcom/example/apkey/MainActivity$a;.>b:Lcom/example/apkey/MainActivity;
    151
    152 invoke-virtual {p1}, Landroid/app/Activity;.>getApplicationContext()Landroid/content/Context;
    153
    154 move-result-object p1
    155
    156 iget-object v0, p0, Lcom/example/apkey/MainActivity$a;.>b:Lcom/example/apkey/MainActivity;
```

Recompile apk with `apktool b -o mod.apk`. We now have a new recompiled apk file with the changed smali code (mod.apk).



```
(jackdisk@kali)-[~/Downloads/APKey]
$ cd ..

(jackdisk@kali)-[~/Downloads]
$ apktool b APKey -o mod.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1-dirty
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
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 $PATH binary)
W: res/drawable/$avd_hide_password_0.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable/$avd_hide_password_1.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable/$avd_hide_password_2.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable/$avd_show_password_0.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable/$avd_show_password_1.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable/$avd_show_password_2.xml: Invalid file name: must contain only [a-z0-9_.]
W: res/drawable-anydpi-v24/$ic_launcher_foreground_0.xml: Invalid file name: must contain only [a-z0-9_.]
brut.androlib.AndrolibException: brut.common.BrutException: could not exec (exit code = 1): [aapt, p, --min-sdk-version, 16, --target-sdk-version, 30, --version-code, 1, --version-name, 1.0, --no-version-vectors, -F, /tmp/APKTOOL10392364437969853763.tmp, -0, resource.es.arsc, -0, png, -0, arsc, -I, /home/jackdisk/.local/share/apktool/framework/1.apk, -S, /home/jackdisk/Downloads/APKey/res, -M, /home/jackdisk/Downloads/APKey/AndroidManifest.xml]
```

We had to create a certificate and sign the mod.apk in order to re-install it on the Android emulator, as after the modification the checksum and the signature.

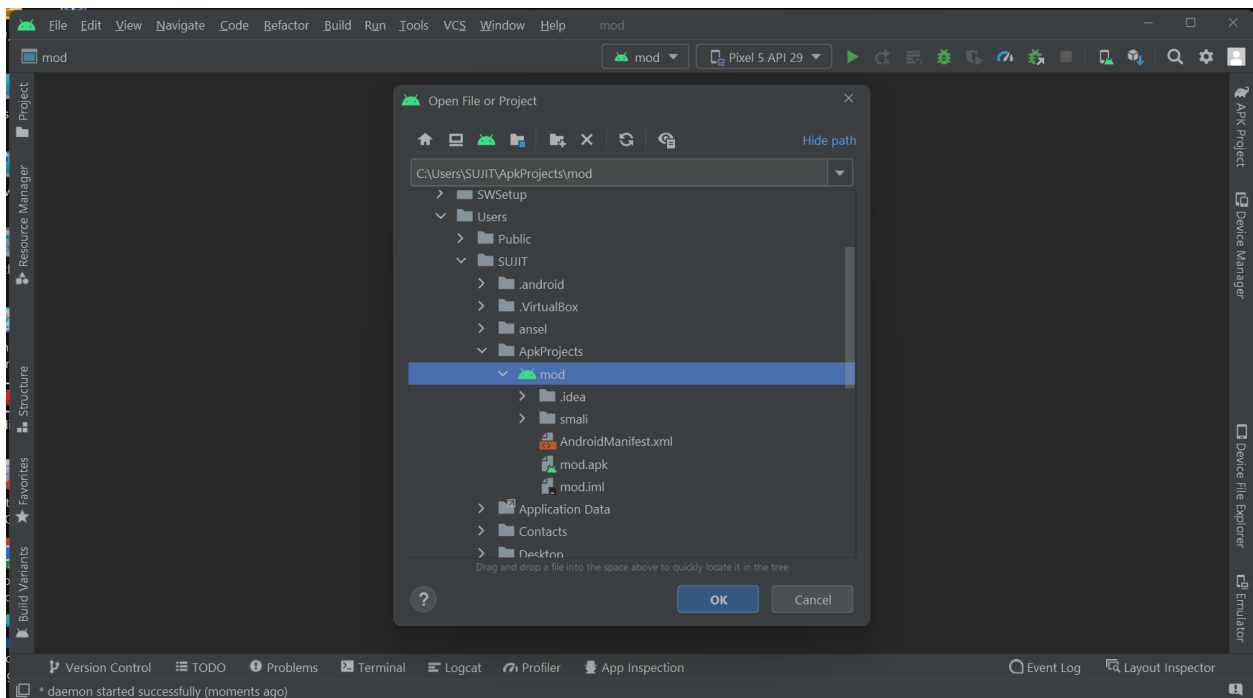
```
keytool -genkey -v -keystore my-release-key.keystore -alias mod_apk -keyalg RSA -keysize 2048 -validity 10000
```

```
jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystore my-release-key.keystore mod.apk mod_apk
```

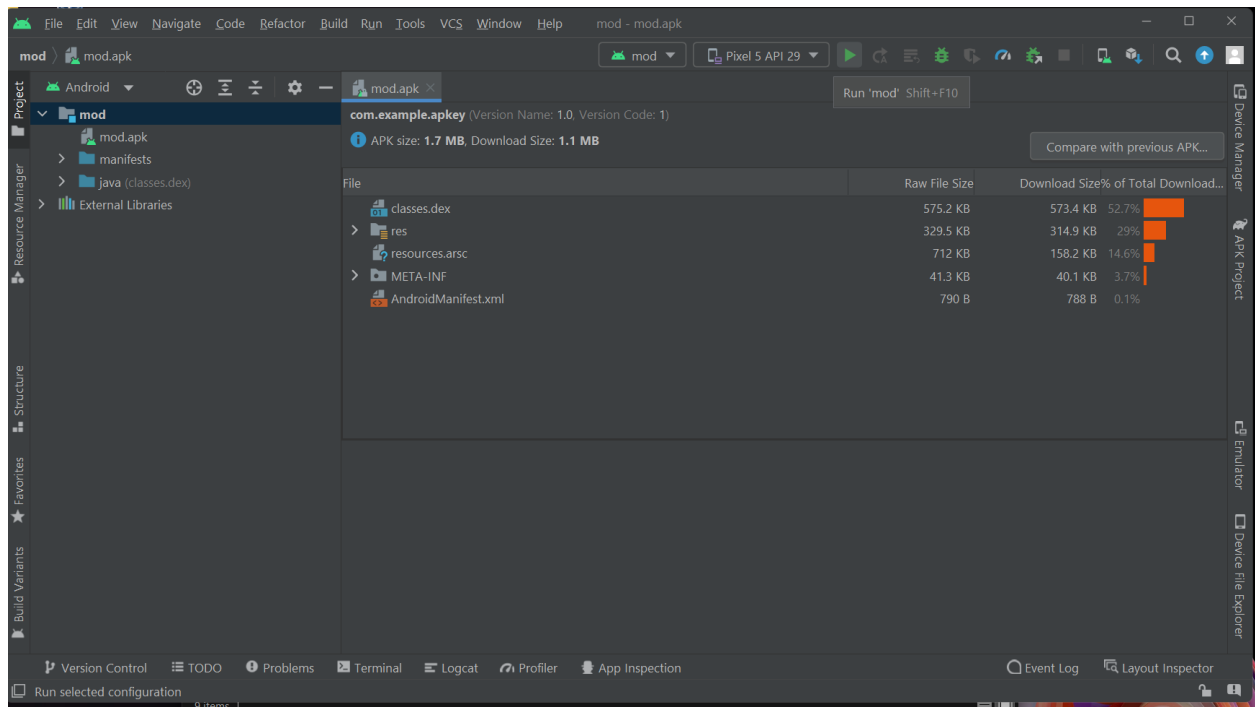
```
(jackdisk@kali)-[~/Downloads]
$ keytool -genkey -v -keystore my-release-key.keystore -alias mod_apk -keyalg RSA -keysize 2048 -validity 10000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Re-enter new password:
What is your first and last name?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

(jackdisk@kali)-[~/Downloads]
$ jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystore my-release-key.keystore mod.apk mod_apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter Passphrase for keystore:
jarsigner: unable to open jar file: mod.apk
```

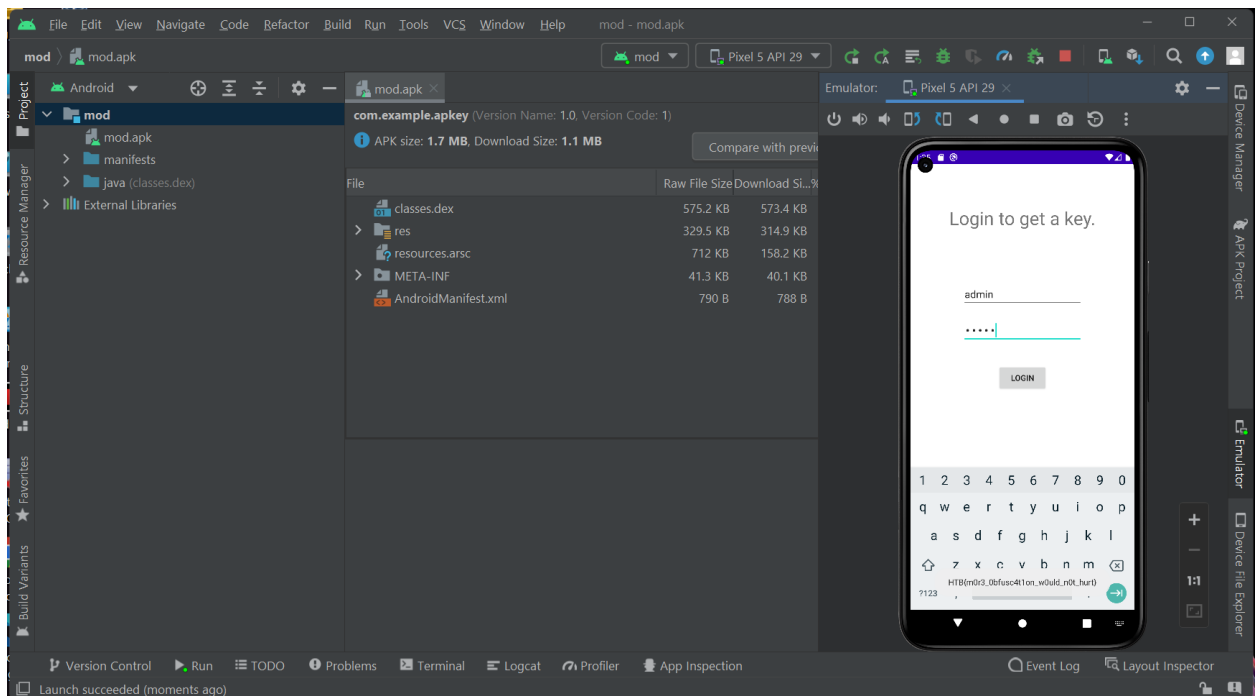
Install mod.apk on the device, let me open up android studio which is a program used for debugging apks .



So you can install different versions of different mobile devices to this you can see here I have got pixel 5 API 29 at the top you can install different versions as your requirement.



Let's play and connect to the emulator. It takes a little while to play. After launching successfully, I type username `admin` and password `admin`. click on Login Flag pops up.



Flag - HTB{m0r3_0bfusc4t1on_w0uld_n0t_hurt}