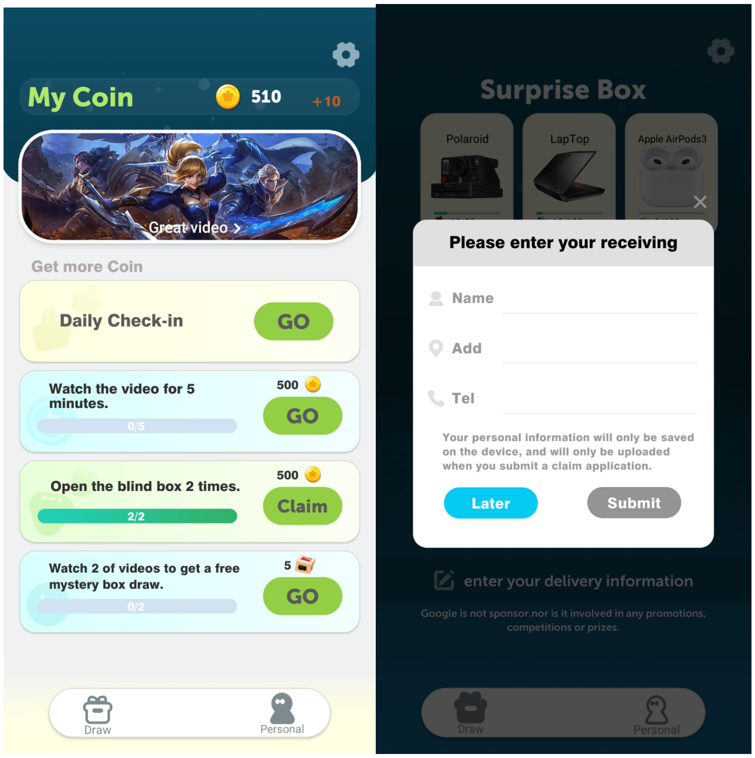
Forensic Analysis of Android Applications

Hands on Experiences: TubeBox An APK Application to Analyze – Case Study

1) For this round, I purposefully found a malicious application and installed it onto my device in order to perform an analysis on what actions it performs on the device, as well as doing a code analysis in APKStudio. The application I selected is called TubeBox, which is a GPT (get-paid-to) app that claims to pay users for watching videos and completing surveys. However, this application is very poorly reviewed and has been shown to not pay users anything for their efforts. It is likely that this is merely an adware platform, and it is possible that the app tracks and stores user information for potentially malicious purposes.

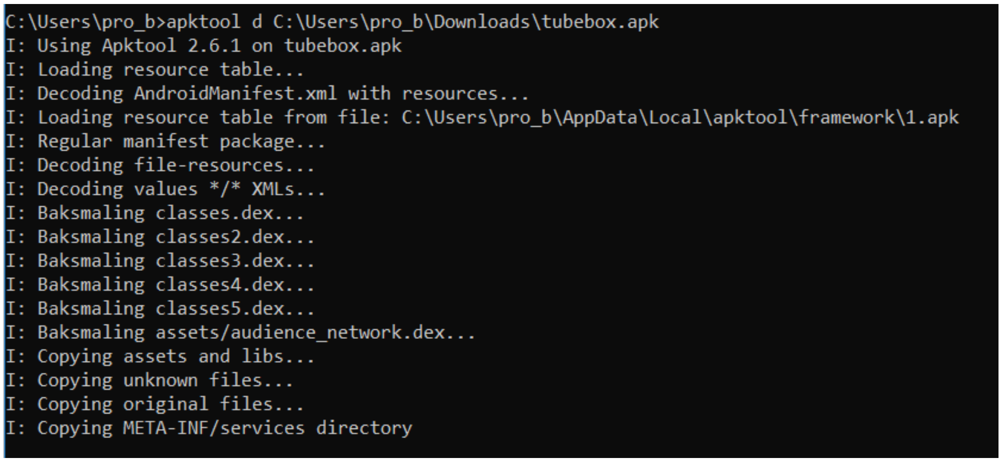
2) This application is no longer on the Play Store, so I used the following link to download the .apk file and install it onto my device.
https://apkcombo.com/tubebox/com.tube.box.surprise.game/download/apk
Upon installing the app, we are presented with the following screens on the device. As we can see, the user is prompted with several daily “earning opportunities” and is also prompted to enter their personal information. When we choose a daily task, we are taken to a video platform similar to Youtube and are presented with coins for completing tasks. We can then use these coins to claim rewards. However, every step of the way we are presented with several ads and surveys, offering more coins and rewards. The cash-out method is very suspicious and it is very likely that this is simply an adware platform designed to bolster CPC (cost per click) numbers.



Your personal information will only be saved on the device, and will only be uploaded when you submit a claim application.

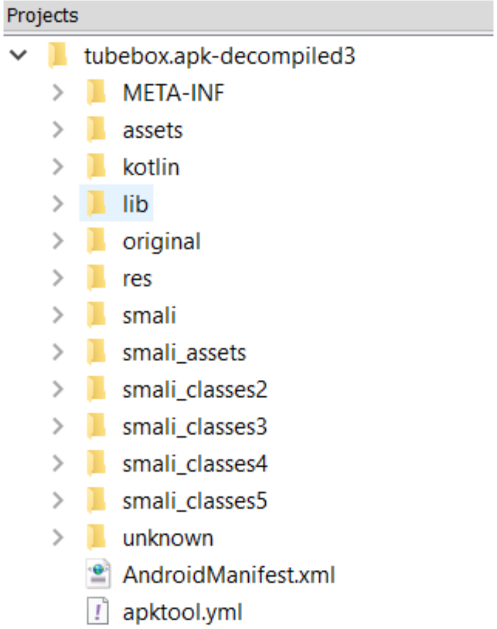
Google is not sponsor.nor is it involved in any promotions, competitions or prizes.

3) Next, in order to perform the code analysis, I unpacked the apk file on my computer and opened the decompiled folder in APKStudio.



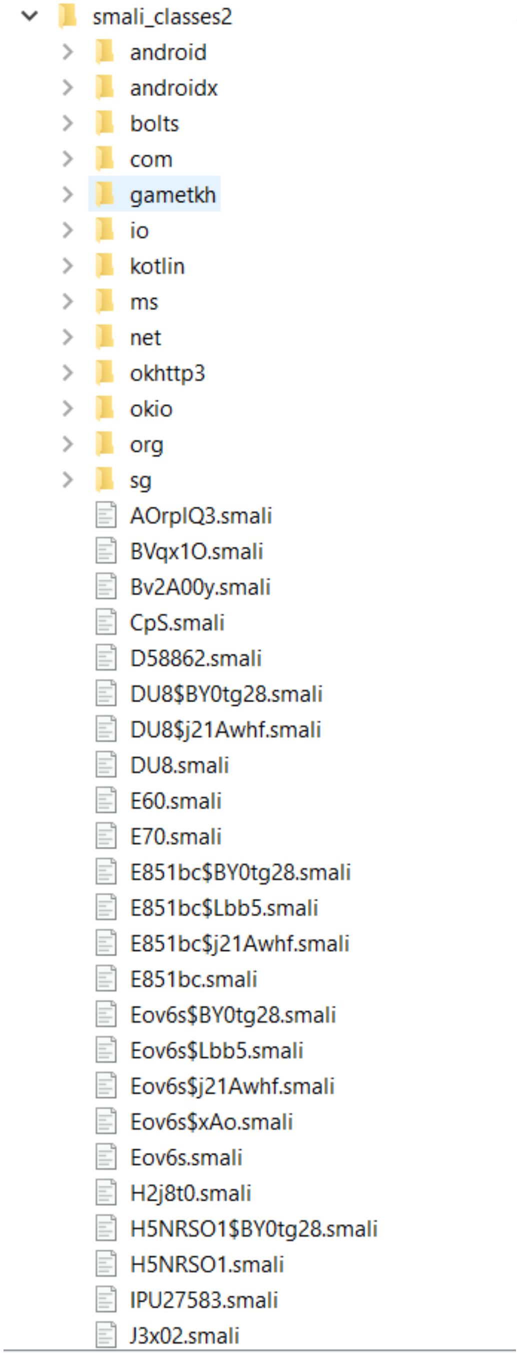
```
C:\Users\pro_b>apktool d C:\Users\pro_b\Downloads\tubebox.apk
I: Using Apktool 2.6.1 on tubebox.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\pro_b\AppData\Local\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes4.dex...
I: Baksmaling classes5.dex...
I: Baksmaling assets/audience_network.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
```

In APKStudio, we are presented with these folders, as shown in the screenshot below. It is notable that there are multiple smali assets folders, which is interesting as normally there is a single smali folder for most apk files.

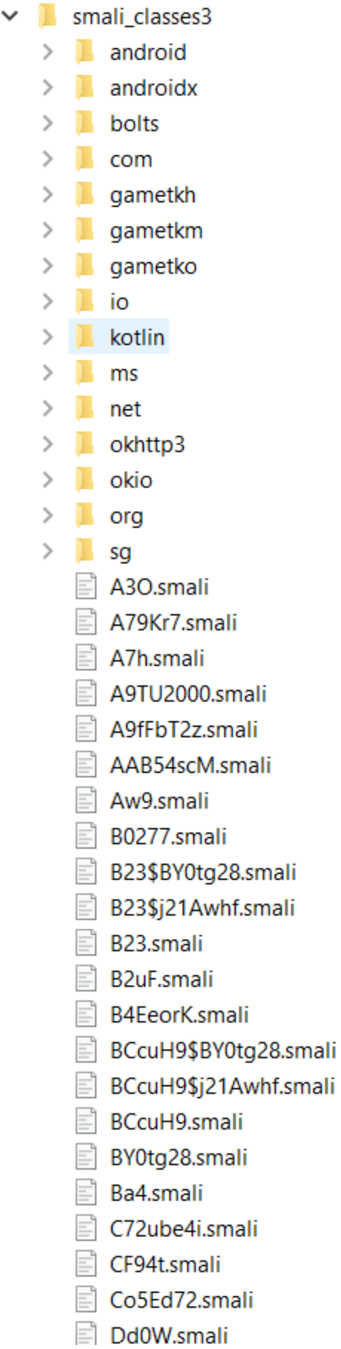


```
Projects
tubebox.apk-decompiled3
    META-INF
    assets
    kotlin
    lib
    original
    res
    smali
    smali_assets
    smali_classes2
    smali_classes3
    smali_classes4
    smali_classes5
    unknown
    AndroidManifest.xml
    apktool.yml
```

When we open one of the smali assets folders, we can see that there are hundreds of smali files in each of the folders, each with random characters as the file name.



```
smali_classes2
  android
  androidx
  bolts
  com
  gametkh
  io
  kotlin
  ms
  net
  okhttp3
  okio
  org
  sg
  AOrplQ3.smali
  BVqx1O.smali
  Bv2A00y.smali
  CpS.smali
  D58862.smali
  DU8$BY0tg28.smali
  DU8$j21Awhf.smali
  DU8.smali
  E60.smali
  E70.smali
  E851bc$BY0tg28.smali
  E851bc$Lbb5.smali
  E851bc$j21Awhf.smali
  E851bc.smali
  Eov6s$BY0tg28.smali
  Eov6s$Lbb5.smali
  Eov6s$j21Awhf.smali
  Eov6s$xAo.smali
  Eov6s.smali
  H2j8t0.smali
  H5NRSO1$BY0tg28.smali
  H5NRSO1.smali
  IPU27583.smali
  J3x02.smali
```



```
smali_classes3
  android
  androidx
  bolts
  com
  gametkh
  gametkm
  gametko
  io
  kotlin
  ms
  net
  okhttp3
  okio
  org
  sg
  A3O.smali
  A79Kr7.smali
  A7h.smali
  A9TU2000.smali
  A9fFbT2z.smali
  AAB54scM.smali
  Aw9.smali
  B0277.smali
  B23$BY0tg28.smali
  B23$j21Awhf.smali
  B23.smali
  B2uF.smali
  B4EeorK.smali
  BCcuH9$BY0tg28.smali
  BCcuH9$j21Awhf.smali
  BCcuH9.smali
  BY0tg28.smali
  Ba4.smali
  C72ube4i.smali
  CF94t.smali
  Co5Ed72.smali
  Dd0W.smali
```

4) This is suspicious, as typical Android applications display traditional naming conventions for their file names, not random combinations of numbers and characters. As there are hundreds and hundreds of these smali files, it would be impossible to search through each and every one of them in orer to discover malicious code. However, it is likely that the user's data is being collected in order to generate tailored videos and advertisements in the video platform.

5) Upon doing some research about TubeBox, I found this video where a person completed several video tasks and entered their information in order to get a pay out. They were placed in a very long queue in order to wait to get their payment, and ultimately never moved up in the queue nor received the payment they were told they earned.

   https://www.youtube.com/watch?v=GjuEi3tLB_I&ab_channel=Vinsane

6) At the time of this submission, the TubeBox app has been taken down off the Google Play Store, but the .apk file can still be found on websites like the one given above and installed onto a user's device. Google rightfully removed this app off the Play Store as it is merely an adware application that provides nothing to the user and only serves to generate advertising revenue for the developers.