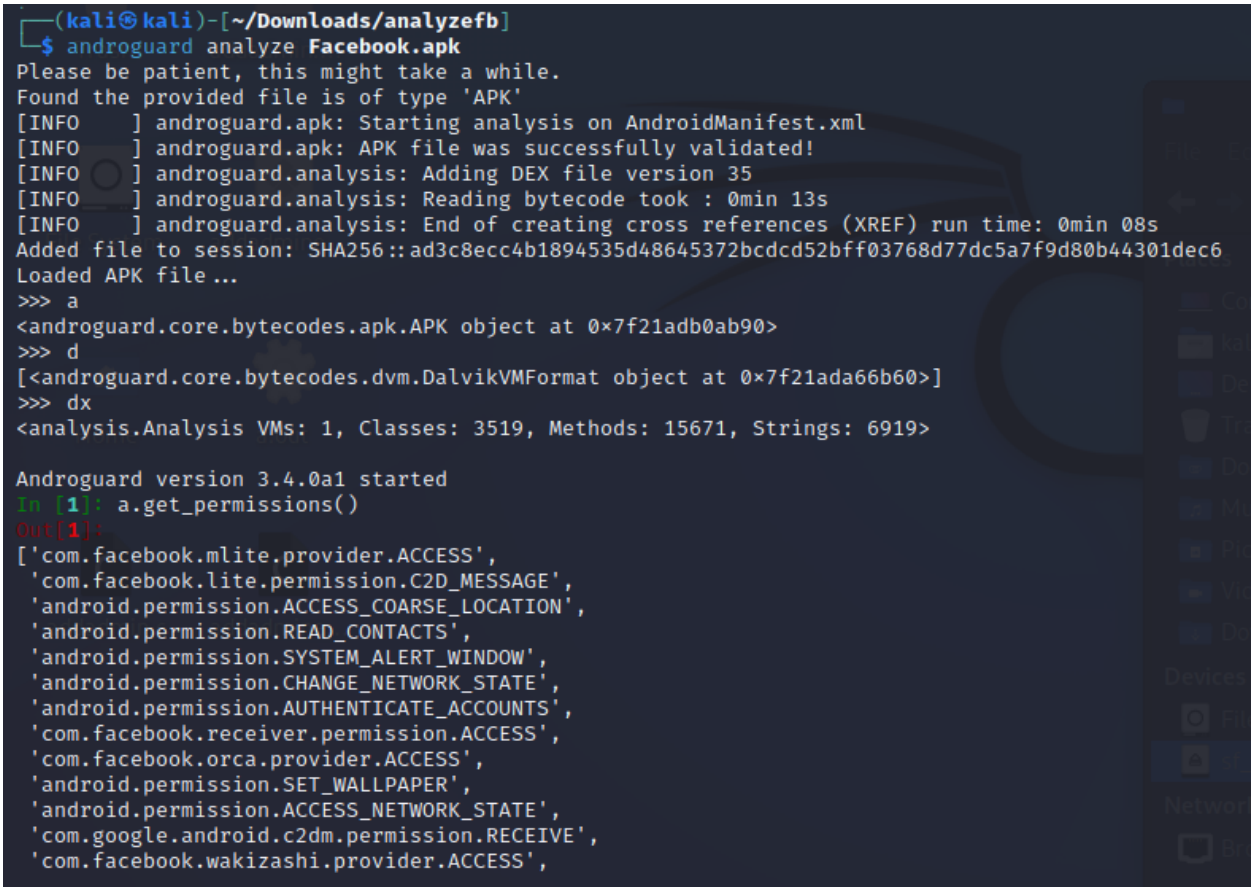
Malicious APK File Analysis
No. 9

To begin analysis, androguard analysis tool installed on kali linux is used to analyze the given apk file. The permissions of the android application are listed with a.get_permissions() command as seen in image below.



```
 'android.permission.VIBRATE',
 'android.permission.RECEIVE_BOOT_COMPLETED',
 'android.permission.WRITE_CALENDAR',
 'android.permission.INTERNET',
 'android.permission.READ_CALENDAR',
 'android.permission.CAMERA',
 'com.facebook.katana.provider.ACCESS',
 'android.permission.GET_TASKS',
 'android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS',
 'android.permission.READ_PHONE_STATE',
 'android.permission.WRITE_EXTERNAL_STORAGE',
 'android.permission.GET_ACCOUNTS',
 'android.permission.WRITE_CONTACTS',
 'android.permission.READ_SMS',
 'android.permission.CALL_PHONE',
 'com.sonymobile.home.permission.PROVIDER_INSERT_BADGE',
 'com.huawei.android.launcher.permission.CHANGE_BADGE',
 'android.permission.RECEIVE_SMS',
 'android.permission.READ_PROFILE',
 'com.oppo.launcher.permission.WRITE_SETTINGS',
 'android.permission.READ_CALL_LOG',
 'com.android.launcher.permission.INSTALL_SHORTCUT',
 'android.permission.WRITE_CALL_LOG',
 'com.huawei.android.launcher.permission.READ_SETTINGS',
 'android.permission.MANAGE_ACCOUNTS',
 'com.huawei.android.launcher.permission.WRITE_SETTINGS',
 'android.permission.RECORD_AUDIO',
 'android.permission.DOWNLOAD_WITHOUT_NOTIFICATION',
 'android.permission.ACCESS_WIFI_STATE',
 'android.permission.WRITE_SETTINGS',
 'com.facebook.permission.prod.FB_APP_COMMUNICATION',
 'android.permission.SEND_SMS',
 'com.htc.launcher.permission.READ_SETTINGS',
 'android.permission.BROADCAST_STICKY',
```

The a.get_activities() command is then used to list the activities of the application. This shows four main activities indicating four possible entry points. It appears the com.facebook.appupdate.WaitForInitActivity entry point can be used for execution of malicious code.

```
In [2]: a.get_activities()
Out[2]:
['com.facebook.lite.MainActivity',
 'com.facebook.lite.photo.AlbumGalleryActivity',
 'com.facebook.lite.photo.PreviewActivity',
 'com.facebook.appupdate.WaitForInitActivity']
```

To get more details on the permissions, a.get_details_permissions() is used to return more information about the permissions. This command labels permissions as normal or dangerous based on the possible level of control a permission gives.

```
In [7]: a.get_details_permissions()
Out[7]:
```

Permissions such as READ_CALL_LOG and READ_SMS being used by facebook lite application appear to be suspicious and as such it is labeled by the tool as “dangerous”.

```
 'android.permission.WRITE_CONTACTS': ['dangerous',
  'modify your contacts',
  "Allows the app to\n    modify the data about your contacts stored on your phone, including the\n    frequency with which you've called, emailed, or communicated in
other ways\n    with specific contacts. This permission allows apps to delete contact\n    data."],
 'android.permission.READ_SMS': ['dangerous',
  'read your text messages (SMS or MMS)',
  'Allows the app to read SMS\n      messages stored on your phone or SIM card. This allows the app to read all\n      SMS messages, regardless of content or confident
iality.'],
 'android.permission.CALL_PHONE': ['dangerous',
  'directly call phone numbers',
  "Allows the app to call phone numbers\n      without your intervention. This may result in unexpected charges or calls.\n      Note that this doesn't allow the app t
o call emergency numbers.\n      Malicious apps may cost you money by making calls without your\n      confirmation."],
```

```
'android.permission.READ_CALL_LOG': ['dangerous',
 'read call log',
 "Allows the app to read\n      your phone's call log, including data about incoming and outgoing calls.\n      This permission allows apps to save your call log data
 and malicious apps\n      may share call log data without your knowledge."],
'com.android.launcher.permission.INSTALL_SHORTCUT': ['dangerous',
 'install shortcuts',
 'Allows an application to add\n      Homescreen shortcuts without user intervention.'],
```

Listing the classes of the application shows some suspicious classes under an X directory. Given the number of classes under the directory labeled X, it calls for further investigation into the function those classes are performing in the application.

```
In [4]: dx.get_classes()
Out[4]: dict_values([<analysis.ClassAnalysis LX/00;>, <analysis.ClassAnalysis LX/01;>, <analysis.ClassAnalysis LX/02;>, <analysis.ClassAnalysis LX/03;>, <analysis.Clas
sAnalysis LX/04;>, <analysis.ClassAnalysis LX/05;>, <analysis.ClassAnalysis LX/06;>, <analysis.ClassAnalysis LX/07;>, <analysis.ClassAnalysis LX/08;>, <analysis.ClassA
nalysis LX/09;>, <analysis.ClassAnalysis LX/0A;>, <analysis.ClassAnalysis LX/0B;>, <analysis.ClassAnalysis LX/0C;>, <analysis.ClassAnalysis LX/0D;>, <analysis.ClassAna
lysis LX/0E;>, <analysis.ClassAnalysis LX/0F;>, <analysis.ClassAnalysis LX/0G;>, <analysis.ClassAnalysis LX/0H;>, <analysis.ClassAnalysis LX/0I;>, <analysis.ClassAnaly
sis LX/0J;>, <analysis.ClassAnalysis LX/0K;>, <analysis.ClassAnalysis LX/0L;>, <analysis.ClassAnalysis LX/0M;>, <analysis.ClassAnalysis LX/0N;>, <analysis.ClassAnalysi
s LX/0O;>, <analysis.ClassAnalysis LX/0P;>, <analysis.ClassAnalysis LX/0Q;>, <analysis.ClassAnalysis LX/0R;>, <analysis.ClassAnalysis LX/0S;>, <analysis.ClassAnalysis
LX/0T;>, <analysis.ClassAnalysis LX/0U;>, <analysis.ClassAnalysis LX/0V;>, <analysis.ClassAnalysis LX/0W;>, <analysis.ClassAnalysis LX/0X;>, <analysis.ClassAnalysis LX
/0Y;>, <analysis.ClassAnalysis LX/0Z;>, <analysis.ClassAnalysis LX/0a;>, <analysis.ClassAnalysis LX/0b;>, <analysis.ClassAnalysis LX/0c;>, <analysis.ClassAnalysis LX/0
d;>, <analysis.ClassAnalysis LX/0e;>, <analysis.ClassAnalysis LX/0f;>, <analysis.ClassAnalysis LX/0g;>, <analysis.ClassAnalysis LX/0h;>, <analysis.ClassAnalysis LX/0i;
>, <analysis.ClassAnalysis LX/0j;>, <analysis.ClassAnalysis LX/0k;>, <analysis.ClassAnalysis LX/0l;>, <analysis.ClassAnalysis LX/0m;>, <analysis.ClassAnalysis LX/0n;>,
 <analysis.ClassAnalysis LX/0o;>, <analysis.ClassAnalysis LX/0p;>, <analysis.ClassAnalysis LX/0q;>, <analysis.ClassAnalysis LX/0r;>, <analysis.ClassAnalysis LX/0s;>, <
analysis.ClassAnalysis LX/0t;>, <analysis.ClassAnalysis LX/0u;>, <analysis.ClassAnalysis LX/0v;>, <analysis.ClassAnalysis LX/0w;>, <analysis.ClassAnalysis LX/0x;>, <an
alysis.ClassAnalysis LX/0y;>, <analysis.ClassAnalysis LX/0z;>, <analysis.ClassAnalysis LX/10;>, <analysis.ClassAnalysis LX/11;>, <analysis.ClassAnalysis LX/12;>, <anal
ysis.ClassAnalysis LX/13;>, <analysis.ClassAnalysis LX/14;>, <analysis.ClassAnalysis LX/15;>, <analysis.ClassAnalysis LX/16;>, <analysis.ClassAnalysis LX/17;>, <analys
is.ClassAnalysis LX/18;>, <analysis.ClassAnalysis LX/19;>, <analysis.ClassAnalysis LX/1A;>, <analysis.ClassAnalysis LX/1B;>, <analysis.ClassAnalysis LX/1C;>, <analysis
.ClassAnalysis LX/1D;>, <analysis.ClassAnalysis LX/1E;>, <analysis.ClassAnalysis LX/1F;>, <analysis.ClassAnalysis LX/1G;>, <analysis.ClassAnalysis LX/1H;>, <analysis.C
lassAnalysis LX/1I;>, <analysis.ClassAnalysis LX/1J;>, <analysis.ClassAnalysis LX/1K;>, <analysis.ClassAnalysis LX/1L;>, <analysis.ClassAnalysis LX/1M;>, <analysis.Cla
ssAnalysis LX/1N;>, <analysis.ClassAnalysis LX/1O;>, <analysis.ClassAnalysis LX/1P;>, <analysis.ClassAnalysis LX/1Q;>, <analysis.ClassAnalysis LX/1R;>, <analysis.Class
Analysis LX/1S;>, <analysis.ClassAnalysis LX/1T;>, <analysis.ClassAnalysis LX/1U;>, <analysis.ClassAnalysis LX/1V;>, <analysis.ClassAnalysis LX/1W;>, <analysis.ClassAn
alysis LX/1X;>, <analysis.ClassAnalysis LX/1Y;>, <analysis.ClassAnalysis LX/1Z;>, <analysis.ClassAnalysis LX/1a;>, <analysis.ClassAnalysis LX/1b;>, <analysis.ClassAnal
ysis LX/1c;>, <analysis.ClassAnalysis LX/1d;>, <analysis.ClassAnalysis LX/1e;>, <analysis.ClassAnalysis LX/1f;>, <analysis.ClassAnalysis LX/1g;>, <analysis.ClassAnalys
is LX/1h;>, <analysis.ClassAnalysis LX/1i;>, <analysis.ClassAnalysis LX/1j;>, <analysis.ClassAnalysis LX/1k;>, <analysis.ClassAnalysis LX/1l;>, <analysis.ClassAnalysis
 LX/1m;>, <analysis.ClassAnalysis LX/1n;>, <analysis.ClassAnalysis LX/1o;>, <analysis.ClassAnalysis LX/1p;>, <analysis.ClassAnalysis LX/1q;>, <analysis.ClassAnalysis L
X/1r;>, <analysis.ClassAnalysis LX/1s;>, <analysis.ClassAnalysis LX/1t;>, <analysis.ClassAnalysis LX/1u;>, <analysis.ClassAnalysis LX/1v;>, <analysis.ClassAnalysis LX/
1w;>, <analysis.ClassAnalysis LX/1x;>, <analysis.ClassAnalysis LX/1y;>, <analysis.ClassAnalysis LX/1z;>, <analysis.ClassAnalysis LX/20;>, <analysis.ClassAnalysis LX/21
;>, <analysis.ClassAnalysis LX/22;>, <analysis.ClassAnalysis LX/23;>, <analysis.ClassAnalysis LX/24;>, <analysis.ClassAnalysis LX/25;>, <analysis.ClassAnalysis LX/26;>
, <analysis.ClassAnalysis LX/27;>, <analysis.ClassAnalysis LX/28;>, <analysis.ClassAnalysis LX/29;>, <analysis.ClassAnalysis LX/2A;>, <analysis.ClassAnalysis LX/2B;>,
<analysis.ClassAnalysis LX/2C;>, <analysis.ClassAnalysis LX/2D;>, <analysis.ClassAnalysis LX/2E;>, <analysis.ClassAnalysis LX/2F;>, <analysis.ClassAnalysis LX/2G;>, <a
nalysis.ClassAnalysis LX/2H;>, <analysis.ClassAnalysis LX/2I;>, <analysis.ClassAnalysis LX/2J;>, <analysis.ClassAnalysis LX/2K;>, <analysis.ClassAnalysis LX/2L;>, <ana
lysis.ClassAnalysis LX/2M;>, <analysis.ClassAnalysis LX/2N;>, <analysis.ClassAnalysis LX/2O;>, <analysis.ClassAnalysis LX/2P;>, <analysis.ClassAnalysis LX/2Q;>, <analy
sis.ClassAnalysis LX/2R;>, <analysis.ClassAnalysis LX/2S;>, <analysis.ClassAnalysis LX/2T;>, <analysis.ClassAnalysis LX/2U;>, <analysis.ClassAnalysis LX/2V;>, <analysi
s.ClassAnalysis LX/2W;>, <analysis.ClassAnalysis LX/2X;>, <analysis.ClassAnalysis LX/2Y;>, <analysis.ClassAnalysis LX/2Z;>, <analysis.ClassAnalysis LX/2a;>, <analysis.
ClassAnalysis LX/2b;>, <analysis.ClassAnalysis LX/2c;>, <analysis.ClassAnalysis LX/2d;>, <analysis.ClassAnalysis LX/2e;>, <analysis.ClassAnalysis LX/2f;>, <analysis.Cl
```

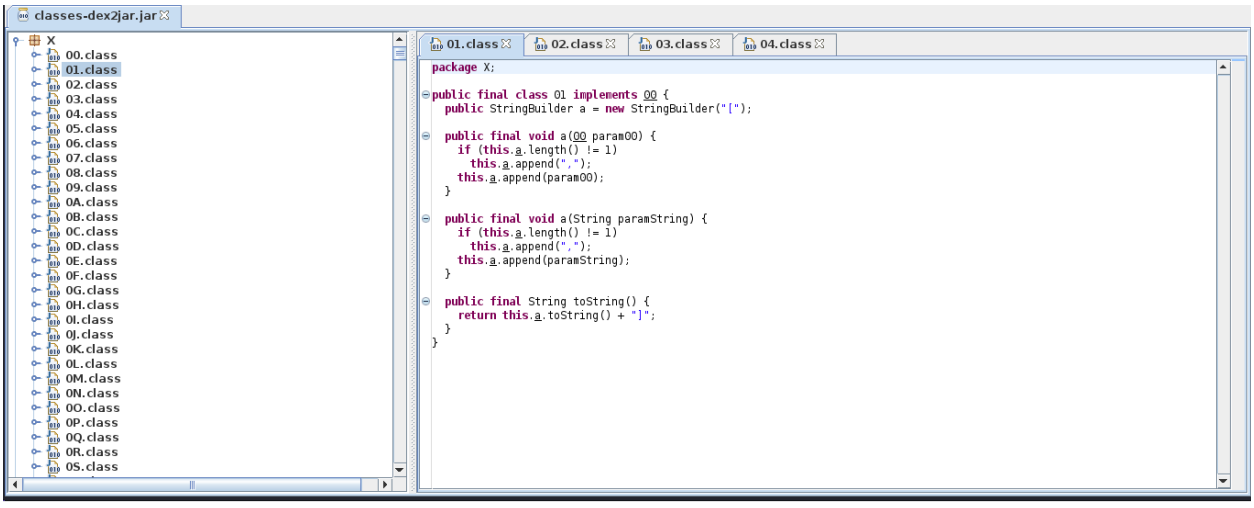
In order to view the java source code of the application, the dex2jar tool is used to convert the .dex to a java archive file (.jar) after the apk file is unzipped with the unzip command.

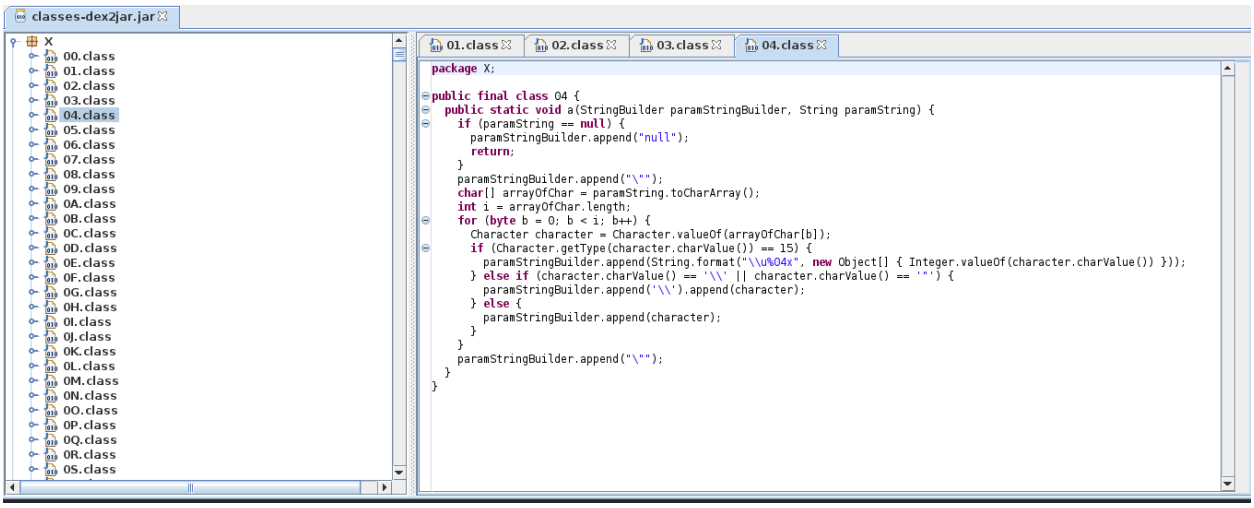
```
┌──(kali㉿kali)-[~/Downloads/analyzefb/unzippedfbapk]
└─$ d2j-dex2jar classes.dex
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
dex2jar classes.dex → ./classes-dex2jar.jar

┌──(kali㉿kali)-[~/Downloads/analyzefb/unzippedfbapk]
└─$ ll
total 6216
-rw-r--r--  1 kali kali   40200 Nov  3 23:27 AndroidManifest.xml
-rw-r--r--  1 kali kali 2339340 Nov  3 23:27 classes.dex
-rw-r--r--  1 kali kali 2705734 Nov 15 21:43 classes-dex2jar.jar
-rwxr-x---  1 kali kali 1216904 Nov 15 21:39 Facebook.apk
drwxr-xr-x 13 kali kali    4096 Nov 15 21:40 res
-rw-r--r--  1 kali kali   48760 Nov  3 23:27 resources.arsc
```

After conversion the jd-gui tool is used to open the archive file in a graphical user interface to analyze the source code of the given apk file.

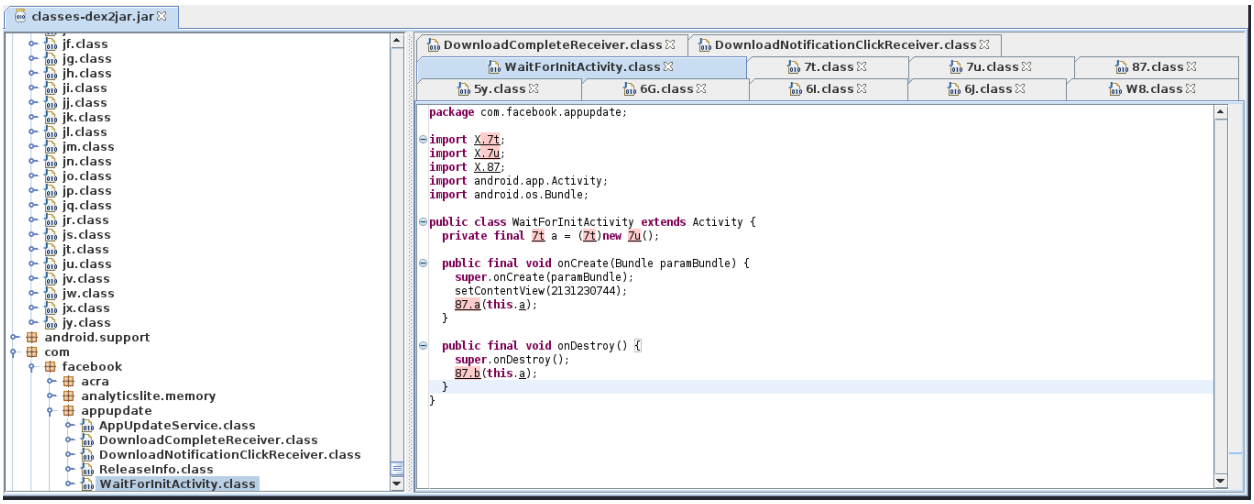
```
┌──(kali㉿kali)-[~/Downloads/analyzefb/unzippedfbapk]
└─$ jd-gui classes-dex2jar.jar
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```

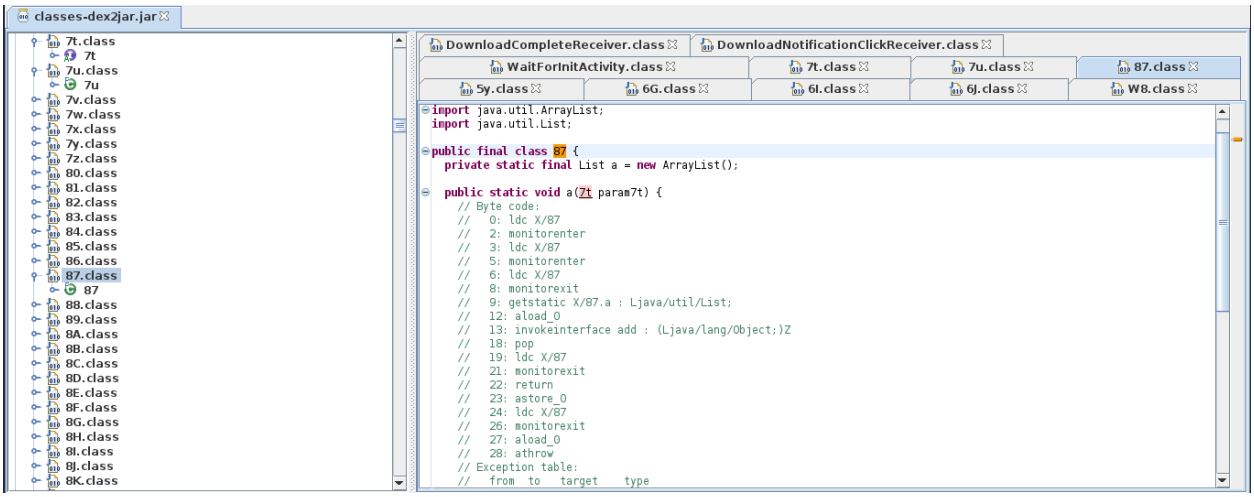
As shown in below images, the classes under the X directory appear to be malicious as it starts to build a string in class 01. In class 04, code for formatting a hex string is seen.





In below images showing classes WaitforinitActivity and 87, the WaitforinitActivity class is designed to call the function in class 87 which contains bytecode. With the characteristics shown, it can be concluded that the facebook lite apk is not an original/benign apk file from facebook but rather a compromised version which acts as a malicious Trojan file edited to download other byte streams unto a user's android device.





The apktool is used to decompile the apk file and its content is analyzed after decompiling. The directories are manually traversed to confirm possible changes to structure of directories and contents of files. Most .smali files as well as AndroidManifest.xml are checked.

```
Z:\Babel\FALL_2022\digital_forensics\assignment2stage2>java -jar apktool_2.0.0rc3.jar d Facebook.apk
I: Using Apktool 2.0.0-RC3 on Facebook.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\rhich_jay23\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...
```

The secret code is found as an xml string entity in the AndroidManifest file as shown in below image on line 498.

Secret-code=**Dhoni_itachi@257**


Z: > Babel > FALL_2022 > digital_forensics > assignment2stage2 > Facebook > AndroidManifest.xml
475
476 <activity-alias android:enabled="false" android:exported="true" android:name="com.facebook.lite.deeplinking.activities.PermalinkHomePhpAc
477
478 <intent-filter>
479
480 <action android:name="android.intent.action.VIEW"/>
481
482 <category android:name="android.intent.category.BROWSABLE"/>
483
484 <category android:name="android.intent.category.DEFAULT"/>
485
486 <data android:scheme="http"/>
487
488 <data android:scheme="https"/>
489
490 <data android:host="www.facebook.com"/>
491
492 <data android:host="m.facebook.com"/>
493
494 <data android:path="/home.php"/>
495
496 </intent-filter>
497
498 <string name="Secret-Code">Dhoni_itachi@257</string>
499
500 </activity-alias>