

# Malicious APK File Creation

## No. 16

1. Analyzing the .apk file using androguard.

```
python3 cli.py analyze malware.apk
2022-11-14 21:33:18.953 INFO | androguard.session:_init_:58 - Opening database <Database(sqlite:///androguard.db)>
2022-11-14 21:33:18.968 INFO | androguard.session:_init_:67 - Creating new session [1]
2022-11-14 21:33:18.969 INFO | main:androlyze_main:257 - Please be patient, this might take a while.
2022-11-14 21:33:18.970 INFO | main:androlyze_main:261 - Found the provided file is of type 'APK'
2022-11-14 21:33:18.971 INFO | androguard.session:addAPK:140 - add APK malware.apk:d3c950ae2ad0e51127f271ea99931e823b70970279c0501525fd96e3aa2a10fc
2022-11-14 21:33:18.981 INFO | androguard.core.apk:_apk_analysis:313 - Starting analysis on AndroidManifest.xml
2022-11-14 21:33:19.010 INFO | androguard.core.apk:_apk_analysis:370 - APK file was successfully validated!
2022-11-14 21:33:19.026 INFO | androguard.session:addDEX:174 - add DEX:7e685b2e0d1d03d9eda7f414c6ea576ae35675bbce39998959dee64da2250404c
2022-11-14 21:33:19.310 INFO | androguard.session:addDEX:180 - added DEX:7e685b2e0d1d03d9eda7f414c6ea576ae35675bbce39998959dee64da2250404c
2022-11-14 21:33:19.310 INFO | androguard.core.analysis.analysis:add:1435 - Adding DEX file version 35
2022-11-14 21:33:20.868 INFO | androguard.core.analysis.analysis:add:1458 - Added DEX in the analysis took : 0min 01s
2022-11-14 21:33:21.339 INFO | androguard.core.analysis.analysis:create_xref:1492 - End of creating cross references (XREF) run time: 0min 00s
2022-11-14 21:33:21.339 INFO | androguard.session:addAPK:160 - added APK malware.apk:d3c950ae2ad0e51127f271ea99931e823b70970279c0501525fd96e3aa2a10fc
2022-11-14 21:33:21.339 INFO | main:androlyze_main:275 - Added file to session: SHA256::d3c950ae2ad0e51127f271ea99931e823b70970279c0501525fd96e3aa2a10fc
2022-11-14 21:33:21.339 INFO | main:androlyze_main:278 - Loaded APK file ...
>>> filename
malware.apk
>>> a
<androguard.core.apk.APK object at 0x7f1369979e10>
>>> d
[<androguard.core.dex.DEX object at 0x7f13697f9510>]
>>> dx
<analysis.Analysis VMs: 1, Classes: 1267, Methods: 6809, Strings: 2176>
Androguard version 4.0 started
In [1]: a.get_signature_name()
Out[1]: 'META-INF/CERT.RSA'
```

Command: a.get\_signature\_name()  
Signature name: 'META-INF/CERT.RSA'

2. Permissions for the .apk

```
In [3]: a.get_permissions()
Out[3]:
['android.permission.SYSTEM_ALERT_WINDOW',
 'android.permission.INTERNET',
 'com.android.launcher.permission.INSTALL_SHORTCUT',
 'android.permission.READ_EXTERNAL_STORAGE',
 'android.permission.CHANGE_WIFI_STATE',
 'android.permission.CHANGE_NETWORK_STATE',
 'android.permission.WRITE_SETTINGS',
 'android.permission.ACCESS_WIFI_STATE',
 'android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS',
 'android.permission.RECEIVE_BOOT_COMPLETED',
 'android.permission.READ_PHONE_STATE',
 'android.permission.WRITE_EXTERNAL_STORAGE',
 'android.permission.MOUNT_UNMOUNT_FILESYSTEMS',
 'android.permission.VIBRATE',
 'android.permission.WAKE_LOCK',
 'android.permission.GET_TASKS',
 'android.permission.ACCESS_NETWORK_STATE']
```

Command: a.get\_permissions()

### 3. what app does (Normal part)

```
in [2]: d.get_details_permissions()
Out[2]:
{'android.permission.SYSTEM_ALERT_WINDOW': ['dangerous',
'allow the app to draw over other applications or parts of the user interface. They may interfere with your use of the interface in any application, or change what you think you are seeing in other applications.'],
'android.permission.INTERNET': ['dangerous',
'full network access',
'allow the app to create network sockets and use custom network protocols. The browser and other applications provide means to send data to the internet, so this permission is not required to send data to the internet.'],
'com.android.launcher.permission.INSTALL_SHORTCUT': ['dangerous',
'install shortcuts',
'allow an application to add home screen shortcuts without user intervention.'],
'android.permission.READ_EXTERNAL_STORAGE': ['normal',
'read the contents of your SD card',
'allow the app to read the contents of your SD card.'],
'android.permission.CHANGE_WIFI_STATE': ['dangerous',
'connect and disconnect from Wi-Fi',
'allow the app to connect to and disconnect from Wi-Fi access points and to make changes to device configuration for Wi-Fi networks.'],
'android.permission.CHANGE_NETWORK_STATE': ['normal',
'change network connectivity',
'allow the app to change the state of network connectivity.'],
'android.permission.WRITE_SETTINGS': ['normal',
'modify system settings',
'allow the app to modify the system settings data. Malicious apps may corrupt your system configuration.'],
'android.permission.ACCESS_WIFI_STATE': ['normal',
'view Wi-Fi connections',
'allow the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and name of connected Wi-Fi devices.'],
'android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS': ['normal',
'unknown permission from android reference'],
'android.permission.RECEIVE_BOOT_COMPLETED': ['normal',
'run at startup',
'allow the app to have itself started as soon as the system has finished booting. This can make it take longer to start the phone and allow the app to slow down the overall phone by always running.'],
'android.permission.READ_PHONE_STATE': ['dangerous',
'read phone status and identity',
'allow the app to access phone features of the device. This permission allows the app to determine the phone number and device IDs, whether a call is active, and the remote number connected by a call.'],
'android.permission.WRITE_EXTERNAL_STORAGE': ['dangerous',
'modify or delete the contents of your SD card',
'allow the app to write to the SD card.'],
'android.permission.MOUNT_UNMOUNT_FILESYSTEMS': ['system|signature',
'access SD Card filesystem',
'allow the app to mount and unmount filesystems for removable storage.'],
'android.permission.VIBRATE': ['normal',
'control vibration',
'allow the app to control the vibrator.'],
'android.permission.WAKE_LOCK': ['normal',
'prevent phone from sleeping',
'allow the app to prevent the phone from going to sleep.'],
'android.permission.GET_TASKS': ['normal',
'retrieve running apps',
'allow the app to retrieve information about currently and recently running tasks. This may allow the app to discover information about which applications are used on the device.'],
'android.permission.ACCESS_NETWORK_STATE': ['normal',
'view network connections',
'allow the app to view information about network connections such as which networks exist and are connected.']}
```

As we can see the ['normal'] in the screen shot. The permissions act normally and don't steal the user data or access.

- Read external storage
- Access network location
- Write settings
- Battery optimization
- Phones vibrate
- Booting
- Wake lock
- Get tasks running on the mobile.

### 4. What app does (dangerous part)

In the above screenshot we can see ["Dangerous"]. The listed permissions can harm user device or can steal data or access.

- Alert window
- Internet – full access
- Change wifi state
- Read phone state
- Write external storage.

5. The picture tells about the services.

Command: a.get\_services()

```
In [1]: a.get_services()
Out[1]:
['trikita.talalarmo.alarm.AlarmService',
 'com.tiffany.webbtech.core.UpdateService',
 'com.xdandroid.hellodaemon.AbsWorkService$WorkNotificationService',
 'com.xdandroid.hellodaemon.JobSchedulerService',
 'com.xdandroid.hellodaemon.WatchDogService',
 'com.xdandroid.hellodaemon.WatchDogService$WatchDogNotificationService']
```

6. The picture tells about the android version code, name and sdk version of min, max and target.

```
In [4]: a.get_androidversion_code()
Out[4]: '1'

In [5]: a.get_androidversion_name()
Out[5]: '1.2.170920'

In [6]:

In [6]: a.get_min_sdk_version()
Out[6]: '15'

In [7]: a.get_max_sdk_version()

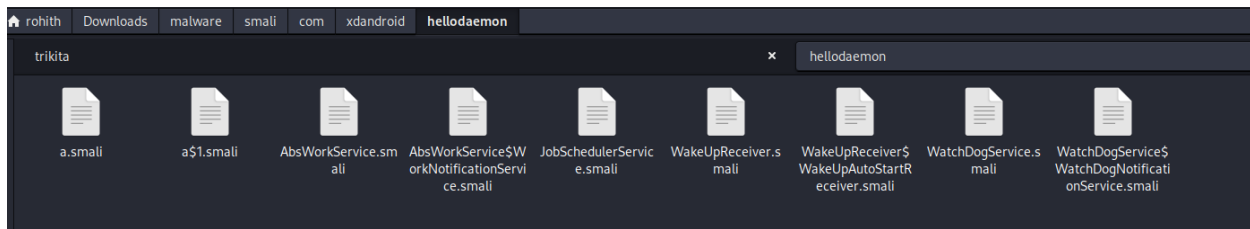
In [8]: a.get_target_sdk_version()
Out[8]: '22'
```

7. Activities performed by the app.

```
In [1]: a.get_activities()
Out[1]:
['trikita.talalarmo.MainActivity',
 'trikita.talalarmo.AgentActivity',
 'trikita.talalarmo.SettingsActivity',
 'trikita.talalarmo.alarm.AlarmActivity',
 'com.google.android.gms.ads.AdActivity',
 'com.tiffany.webbtech.core.WebViewActivity']
```

8. Decompile the .apk file using apktool

Command: apktool d malware.apk



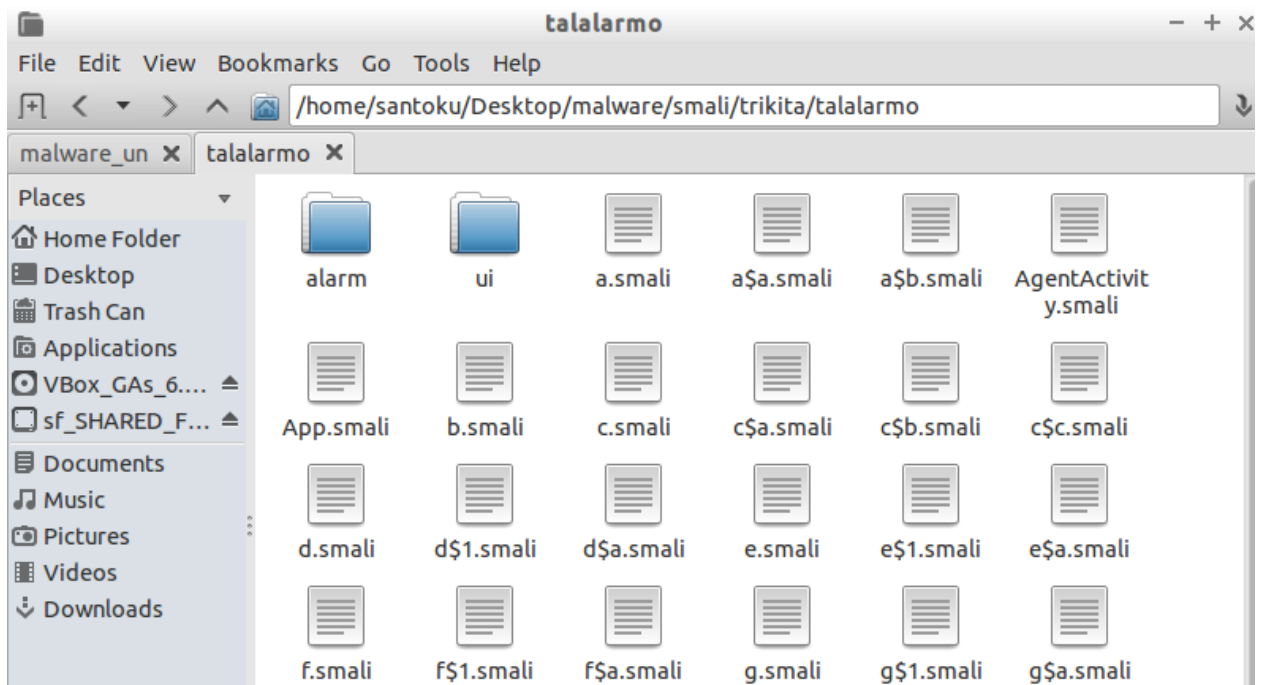
After analyzing the decompiled .apk we can see WatchDogService files which allow malicious activities.

Watchdog is a rogue anti-spyware program from Rogue. FakeVimes family of computer infections.

9. Unzip the malware.apk using unzip command.

```
santoku@santoku-VirtualBox: ~/Downloads
File Edit Tabs Help
santoku@santoku-VirtualBox:~/Downloads$ unzip malware.apk -d malware_un
Archive:  malware.apk
  inflating: malware_un/AndroidManifest.xml
  inflating: malware_un/META-INF/CERT.RSA
  inflating: malware_un/META-INF/CERT.SF
  inflating: malware_un/META-INF/MANIFEST.MF
  inflating: malware_un/META-INF/services/javax.ws.rs.ext.MessageBodyReader
  inflating: malware_un/META-INF/services/javax.ws.rs.ext.MessageBodyWriter
  inflating: malware_un/assets/armeabi-v7a/skysea
  inflating: malware_un/assets/armeabi/skysea
  inflating: malware_un/classes.dex
  extracting: malware_un/res/drawable-hdpi-v4/ic_launcher.png
  extracting: malware_un/res/drawable-mdpi-v4/ic_launcher.png
  extracting: malware_un/res/drawable-xhdpi-v4/ic_launcher.png
  extracting: malware_un/res/drawable-xxhdpi-v4/app_icon.png
  extracting: malware_un/res/drawable-xxhdpi-v4/close_m.png
  extracting: malware_un/res/drawable-xxhdpi-v4/ic_launcher.png
  extracting: malware_un/res/drawable-xxhdpi-v4/ic_launcher.png
  inflating: malware_un/res/drawable/dialog_bg_m.xml
  inflating: malware_un/res/layout/activity_browser.xml
  inflating: malware_un/res/layout/activity_main.xml
  inflating: malware_un/res/layout/rect_dialog.xml
  inflating: malware_un/res/menu/overflow_popup.xml
  extracting: malware_un/res/mipmap-hdpi-v4/ic_launcher.png
```

10. After unzipping, we can see some .smali files which are responsible for triggering malicious activities with the original functionality of the app.



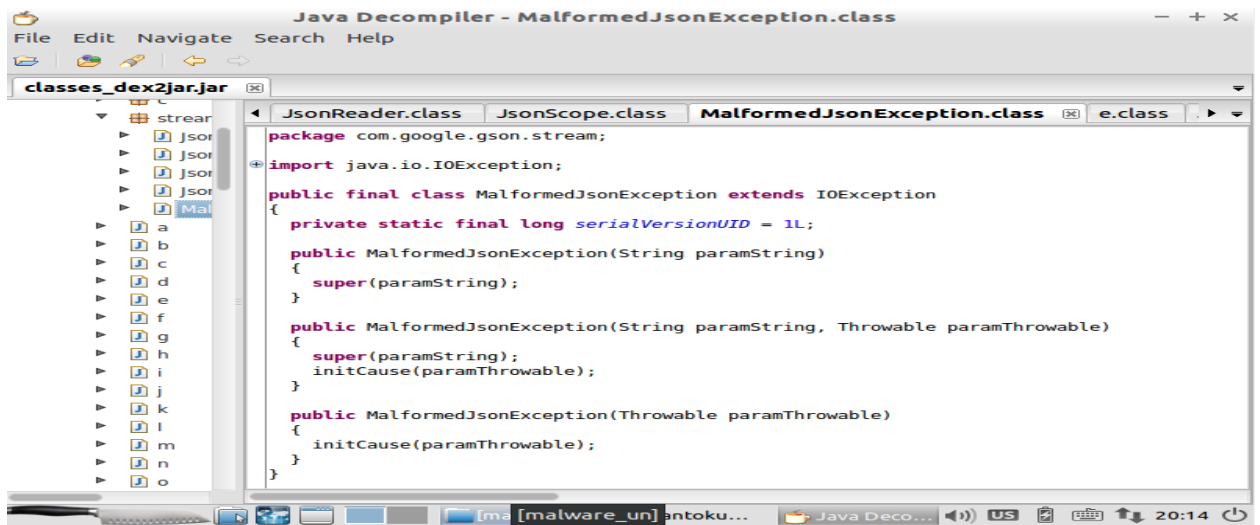
11. Now, we can see classes.dex file in the malware\_uz folder.

We can open the classes.dex2jar.jar file using JD-GUI.

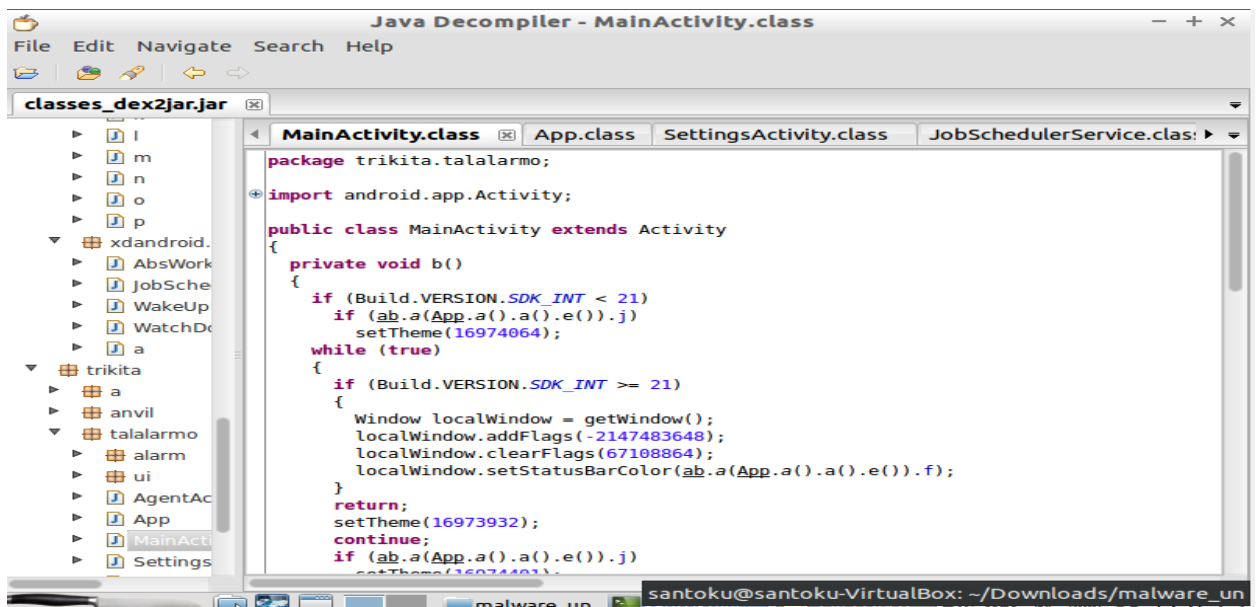
Command: dex2jar classes.dex

```
santoku@santoku-VirtualBox: ~/Downloads/malware_un
File Edit Tabs Help
extracting: malware_un/res/drawable-xxhdpi-v4/close_m.png
extracting: malware_un/res/drawable-xxhdpi-v4/ic_launcher.png
extracting: malware_un/res/drawable-xxhdpi-v4/ic_launcher.png
inflating: malware_un/res/drawable/dialog_bg_m.xml
inflating: malware_un/res/layout/activity_browser.xml
inflating: malware_un/res/layout/activity_main.xml
inflating: malware_un/res/layout/rect_dialog.xml
inflating: malware_un/res/menu/overflow_popup.xml
extracting: malware_un/res/mipmap-hdpi-v4/ic_launcher.png
extracting: malware_un/res/mipmap-mdpi-v4/ic_launcher.png
extracting: malware_un/res/mipmap-xhdpi-v4/ic_launcher.png
extracting: malware_un/res/mipmap-xxhdpi-v4/ic_launcher.png
extracting: malware_un/res/mipmap-xxhdpi-v4/ic_launcher.png
inflating: malware_un/res/xml/preferences.xml
extracting: malware_un/resources.arsc
santoku@santoku-VirtualBox:~/Downloads$ cd malware_uz
bash: cd: malware_uz: No such file or directory
santoku@santoku-VirtualBox:~/Downloads$ cd malware_un
santoku@santoku-VirtualBox:~/Downloads/malware_un$ dex2jar classes.dex
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar classes.dex -> classes_dex2jar.jar
Done.
santoku@santoku-VirtualBox:~/Downloads/malware_un$
```

12. We didn't see any obfuscation in the code. Everything is clear but, some objects and classes are unclear like a,b and so on.



13. We found the MainActivity file was in the .jar file where the program starts.



14. Created control graph of the .apk file using androguard.

```

(sudha_gudla@Kali)-[~/Downloads]
$ androguard decompile -o test3 -f png -i malware.apk --limit "trikita/.*"
[INFO    ] androguard.apk: Starting analysis on AndroidManifest.xml
[INFO    ] androguard.apk: APK file was successfully validated!
[INFO    ] androguard.analysis: Adding DEX file version 35
[INFO    ] androguard.analysis: Reading bytecode took : 0min 01s
[INFO    ] androguard.analysis: End of creating cross references (XREF) run time: 0min 01s
Dump information malware.apk in test3
Create directory test3
Decompilation ... End
Dump Ltrikita/a/a; <init> (Ljava/lang/Enum;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/a; <init> (Ljava/lang/Enum; Ljava/lang/Object;)V ... png ... bytecodes ...
Dump Ltrikita/a/a; toString ()Ljava/lang/String; ... png ... bytecodes ...
Dump Ltrikita/a/c$a; a (Ltrikita/a/c; Ljava/lang/Object; Ltrikita/a/c$b;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/b; <init> (Ljava/lang/String;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/b; a (Ltrikita/a/c; Ljava/lang/Object; Ltrikita/a/c$b;)V ... png ... bytecodes ...
Dump Ltrikita/a/c$1; <init> (Ltrikita/a/c;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c$1; a (Ltrikita/a/c; Ljava/lang/Object; Ltrikita/a/c$b;)V ... png ... bytecodes ...
Dump Ltrikita/a/c$b; a (Ljava/lang/Object;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c$2; <init> (Ltrikita/a/c;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c$2; a (Ljava/lang/Object;)V ... png ... bytecodes ...
Dump Ltrikita/a/c$3; <init> (Ltrikita/a/c; Ltrikita/a/c$a; Ltrikita/a/c$b;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c$3; a (Ljava/lang/Object;)V ... png ... bytecodes ...
Dump Ltrikita/a/c$4; <init> (Ltrikita/a/c; Ljava/lang/Runnable;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c$4; run ()V ... png ... bytecodes ...
Dump Ltrikita/a/c$c; a (Ljava/lang/Object; Ljava/lang/Object;)Ljava/lang/Object; ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c; <init> (Ltrikita/a/c$c; Ljava/lang/Object; [Ltrikita/a/c$a;)V ... png ... source codes ... bytecodes ...
Dump Ltrikita/a/c; a (Ltrikita/a/c;)Ljava/lang/Object; ... png ... bytecodes ...

```

Command: androguard decompile -o outputfolder -f format png -i .apk file --limit "file where you want the graph"

15. The graph of the main activity tells about the classes and objects which are accessible

