



Malicious APK File Analysis

No. 18

Analysing the APK file using Santoku:

An Android app reverse engineering tool called Androguard is built on Python. The raw Android Package files of the application must be broken down and analyzed in order to achieve this. Here we are using the Androguard tool for analyzing the application.

Androguard provides a Python interactive shell via which users may provide several commands to the API.

Here are some commands

```
ravi2308@ravi2308-VirtualBox: /usr/share/androguard
File Edit Tabs Help
ravi2308@ravi2308-VirtualBox:~$ androlyze.py
androlyze.py: command not found
ravi2308@ravi2308-VirtualBox:~$ Androlyze.py
Androlyze.py: command not found
ravi2308@ravi2308-VirtualBox:~$ cd /usr/share/androguard
ravi2308@ravi2308-VirtualBox:/usr/share/androguard$ ./androlyze.py -s
/usr/lib/python2.7/dist-packages/IPython/frontend.py:30: UserWarning: The top-level `frontend` package has been deprecated. All its subpackages have been moved to the top `IPython` level.
  warn("The top-level `frontend` package has been deprecated. ")
Androlyze version 2.0
In [1]: a,d,dx=AnalyzeAPK("/home/ravi2308/Downloads/alaraminfected.apk", decompiler="dad")

In [2]: a.get_activities()
Out[2]:
['com.better.alarm.presenter.AlarmsListActivity',
 'com.better.alarm.presenter.SettingsActivity',
 'com.better.alarm.presenter.HandleSetAlarm',
 'com.better.alarm.alert.AlarmAlertFullScreen',
 'com.better.alarm.presenter.TransparentActivity']

In [3]: a.get_permissions()
Out[3]:
['android.permission.READ_CONTACTS',
 'android.permission.INTERNET',
 'android.permission.ACCESS_FINE_LOCATION',
 'android.permission.READ_PHONE_STATE',
 'android.permission.RECEIVE_SMS',
 'android.permission.RECORD_AUDIO',
```

- `a,d,dx = AnalyzeAPK(<apk_filename>, decompiler="dad")` is the default command for decompiling the apk file
- using `a.get_permissions()` command we get the permissions required for this application

```
ravi2308@ravi2308-VirtualBox: /usr/share/androguard
File Edit Tabs Help
tions can corrupt your system's configuration."],
'com.android.alarm.permission.SET_ALARM': ['normal',
'set alarm in alarm clock',
'Allows the application to set an alarm in an installed alarm clock applicatio
n. Some alarm clock applications may not implement this feature.']]

In [5]: a.get_services()
Out[5]:
['com.better.alarm.background.AlertServiceWrapper',
'org.acra.sender.LegacySenderService',
'org.acra.sender.JobSenderService',
'com.better.alarm.lxhiy.Udnna']

In [6]: a.get_receivers()
Out[6]: ['com.better.alarm.model.AlarmsReceiver', 'com.better.alarm.lxhiy.Ypmmh'
]

In [7]: a.get_androidversion_code()
Out[7]: u'31010'

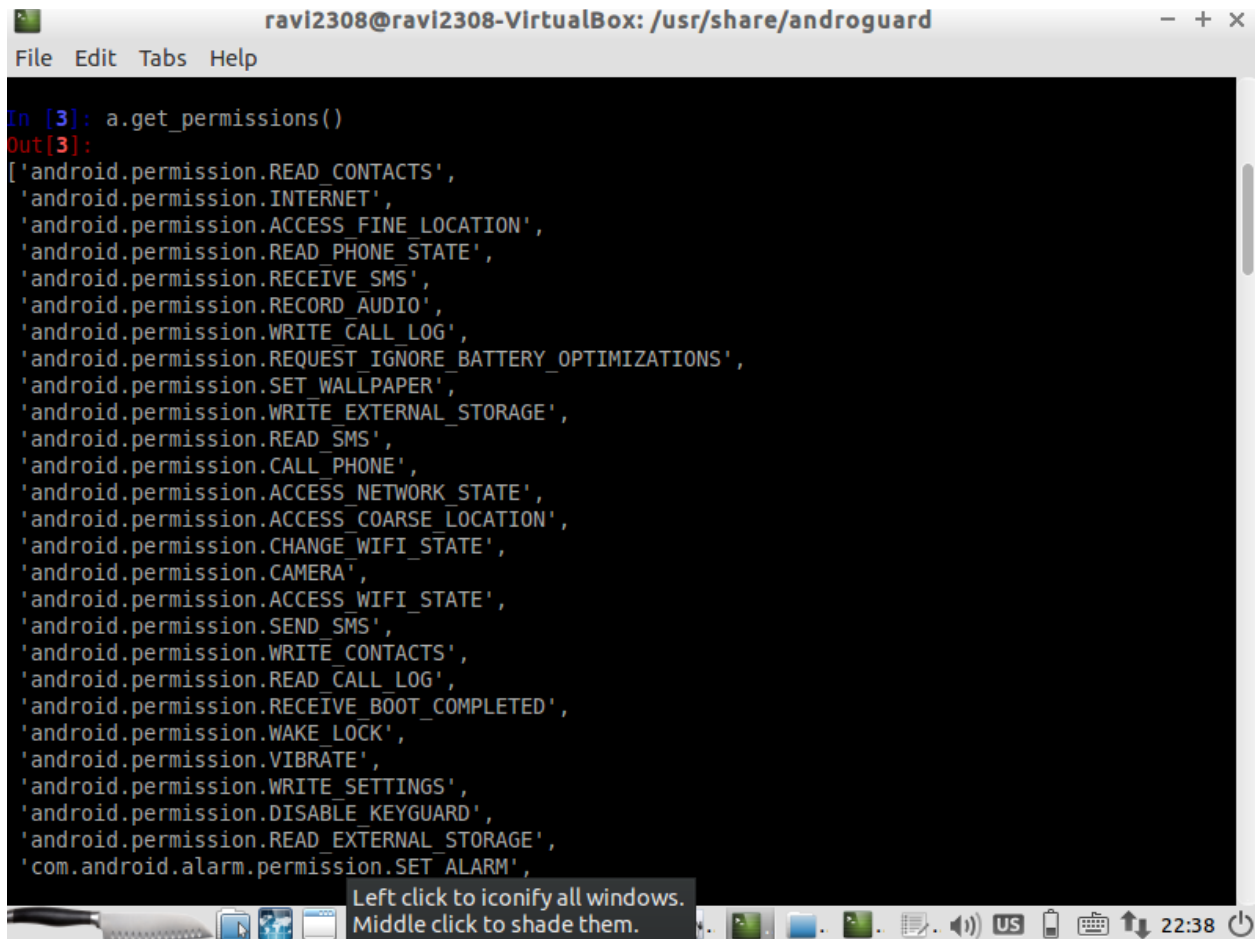
In [8]: a.get_androidversion_name()
Out[8]: u'3.10.10'

In [9]: a.get_min_sdk_version()
Out[9]: u'16'

In [10]: a.get_max_sdk_version()

In [11]: a.get_signature_name()
Out[11]: u'META-INF/SIGNING_.RSA'
```

- Above are the various commands we can use to get details about the application.
- Here the application is trying to get the permissions from the device, once the user gives access to the permissions application gets access to the data related to the given permissions. Below are the permissions required for this application



The screenshot shows a terminal window titled "ravi2308@ravi2308-VirtualBox: /usr/share/androguard". The terminal displays the output of a Python script that calls `a.get_permissions()`. The output is a list of 30 Android permissions, including `android.permission.READ_CONTACTS`, `android.permission.INTERNET`, `android.permission.ACCESS_FINE_LOCATION`, `android.permission.READ_PHONE_STATE`, `android.permission.RECEIVE_SMS`, `android.permission.RECORD_AUDIO`, `android.permission.WRITE_CALL_LOG`, `android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS`, `android.permission.SET_WALLPAPER`, `android.permission.WRITE_EXTERNAL_STORAGE`, `android.permission.READ_SMS`, `android.permission.CALL_PHONE`, `android.permission.ACCESS_NETWORK_STATE`, `android.permission.ACCESS_COARSE_LOCATION`, `android.permission.CHANGE_WIFI_STATE`, `android.permission.CAMERA`, `android.permission.ACCESS_WIFI_STATE`, `android.permission.SEND_SMS`, `android.permission.WRITE_CONTACTS`, `android.permission.READ_CALL_LOG`, `android.permission.RECEIVE_BOOT_COMPLETED`, `android.permission.WAKE_LOCK`, `android.permission.VIBRATE`, `android.permission.WRITE_SETTINGS`, `android.permission.DISABLE_KEYGUARD`, `android.permission.READ_EXTERNAL_STORAGE`, and `com.android.alarm.permission.SET_ALARM`. A tooltip at the bottom of the terminal window reads: "Left click to iconify all windows. Middle click to shade them." The system tray at the bottom shows the time as 22:38 and includes icons for network, volume, and power.

```
ravi2308@ravi2308-VirtualBox: /usr/share/androguard
File Edit Tabs Help

In [3]: a.get_permissions()
Out[3]:
['android.permission.READ_CONTACTS',
'android.permission.INTERNET',
'android.permission.ACCESS_FINE_LOCATION',
'android.permission.READ_PHONE_STATE',
'android.permission.RECEIVE_SMS',
'android.permission.RECORD_AUDIO',
'android.permission.WRITE_CALL_LOG',
'android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS',
'android.permission.SET_WALLPAPER',
'android.permission.WRITE_EXTERNAL_STORAGE',
'android.permission.READ_SMS',
'android.permission.CALL_PHONE',
'android.permission.ACCESS_NETWORK_STATE',
'android.permission.ACCESS_COARSE_LOCATION',
'android.permission.CHANGE_WIFI_STATE',
'android.permission.CAMERA',
'android.permission.ACCESS_WIFI_STATE',
'android.permission.SEND_SMS',
'android.permission.WRITE_CONTACTS',
'android.permission.READ_CALL_LOG',
'android.permission.RECEIVE_BOOT_COMPLETED',
'android.permission.WAKE_LOCK',
'android.permission.VIBRATE',
'android.permission.WRITE_SETTINGS',
'android.permission.DISABLE_KEYGUARD',
'android.permission.READ_EXTERNAL_STORAGE',
'com.android.alarm.permission.SET_ALARM',
```

- Here the application is getting access to contacts, SMS, Wallpaper, Audio records, phone calls, Internet and other data.
- Application is the Multipurpose Internet Mail Extension (MIME) for sending the data.
- MIME is an Internet standard that enhances email message formats to include attachments of music, video, pictures, and application programs in addition to text in character sets other than ASCII.
- RFC822 is the MIME type which is used by the application. RFC822 is an email format which consists of header fields and message body.

```
</uses-permission>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE">
</uses-permission>
<uses-permission android:name="android.permission.USE_FULL_SCREEN_INTENT">
</uses-permission>
-<queries>
  -<intent>
    <action android:name="android.intent.action.SEND"> </action>
    <data android:mimeType="message/rfc822"> </data>
  </intent>
  -<intent>
    <action android:name="android.intent.action.SEND_MULTIPLE"> </action>
    <data android:mimeType="message/rfc822"> </data>
  </intent>
  -<intent>
    <action android:name="android.intent.action.SENDTO"> </action>
  </intent>
</queries>
-<application android:allowBackup="true"
  android:appComponentFactory="androidx.core.app.CoreComponentFactory"
  android:fullBackupOnly="true" android:icon="@7F0E0000"
  android:installLocation="1" android:label="@7F1100D0"
  android:name="com.better.alarm.configuration.AlarmApplication">
  -<activity android:configChanges="0x000000F0" android:exported="true"
```

- There is no secret code embedded in this application