Retrieving APK Files

pentestlab.blog/category/mobile-pentesting/page/5

January 30, 2017

The first step in every android security assessment is to have the .APK file which is the actual application. In the majority of the cases the client is responsible to provide this file especially in a situation where the actual application is not publicly available. However if for whatever reason this is not possible (i.e. client has requested a black box assessment) then it is up to the consultant to obtain this file.

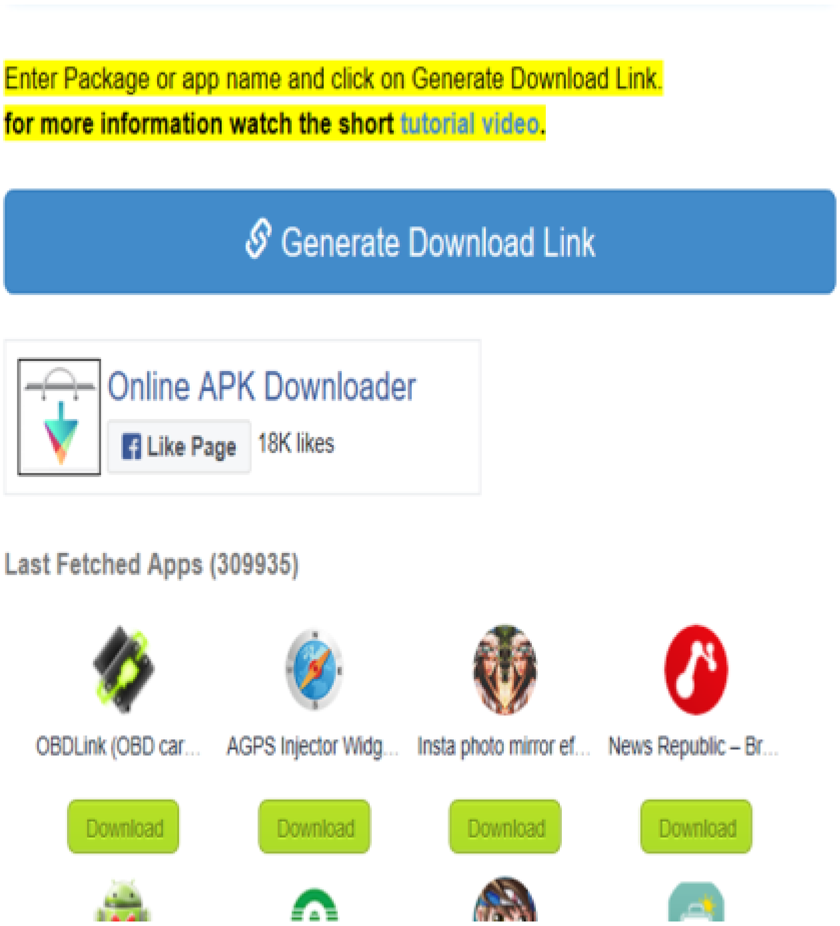
The are three different scenarios of how to retrieve an APK file:

- Client provides the APK file directly
- Application is available on Google Play Store
- Application is already installed on the phone

If the client provides the APK file then everything is ready and the consultant he can start with the assessment. So lets explore the other two scenarios.

Google Play Store

For applications that are available publicly and are included already in the Google Play Store there are various websites that can provide APK files like apkleecher and apkdownloader.



Enter Package or app name and click on Generate Download Link.

for more information watch the short tutorial video.

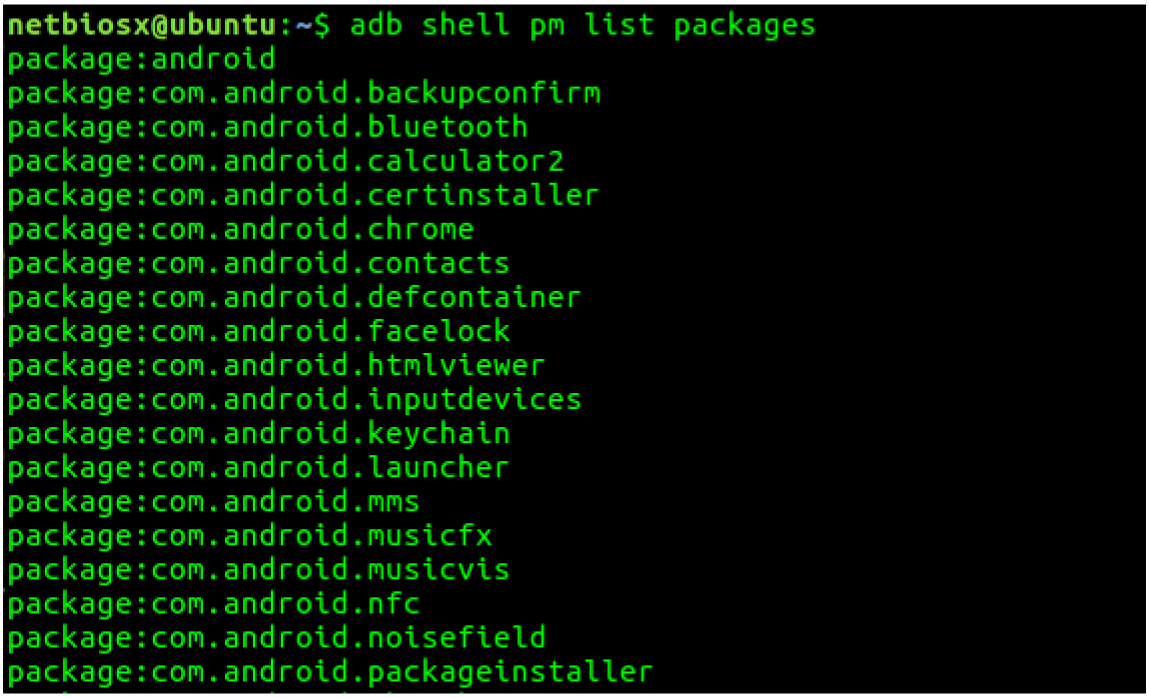
Android Phone

For applications that are already installed on the Android phone the consultant can start the testing immediately by using Burp. However this can cover only the dynamic analysis testing. In order to fully perform the assessment the APK file is essential as well for static analysis of the files that are included in the APK like the manifest file and for reverse engineering the application to investigate further vulnerabilities.

Tools such as Drozer and adb can reveal the location of the APK file on the phone.

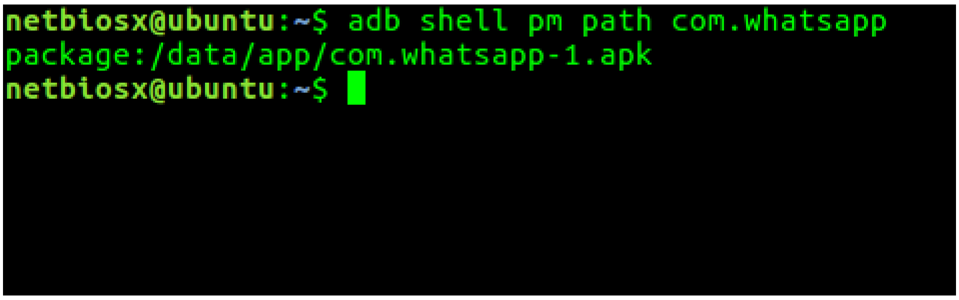
ADB

The first step is to obtain the list of applications that are installed on the phone with the following command:



```
netbiosx@ubuntu:~$ adb shell pm list packages
package:android
package:com.android.backupconfirm
package:com.android.bluetooth
package:com.android.calculator2
package:com.android.certinstaller
package:com.android.chrome
package:com.android.contacts
package:com.android.defcontainer
package:com.android.facelock
package:com.android.htmlviewer
package:com.android.inputdevices
package:com.android.keychain
package:com.android.launcher
package:com.android.mms
package:com.android.musicfx
package:com.android.musicvis
package:com.android.nfc
package:com.android.noisefield
package:com.android.packageinstaller
```

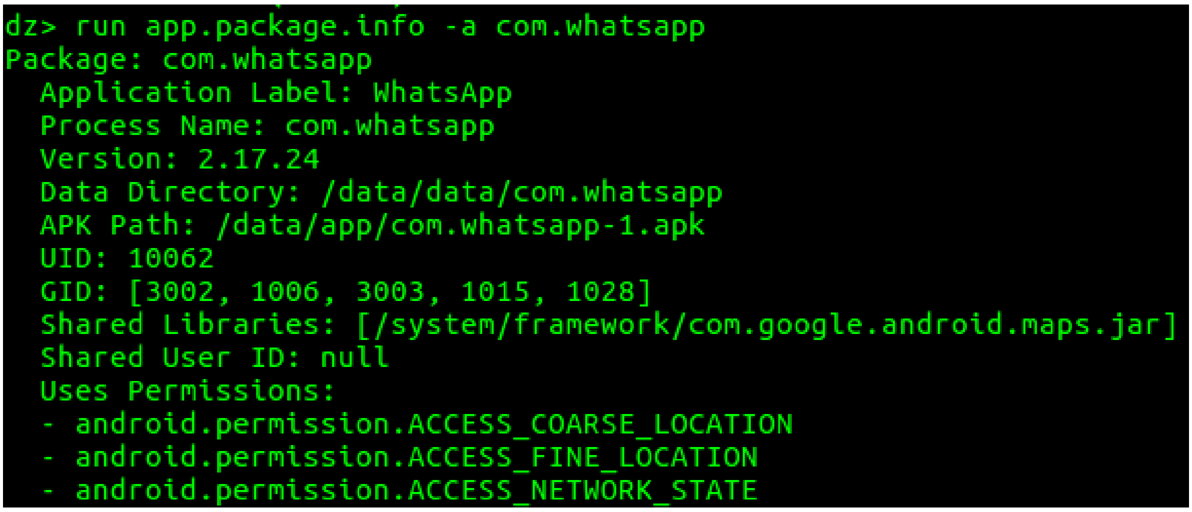
The location of where the .APK file is stored on the device can be discovered with the following:



```
netbiosx@ubuntu:~$ adb shell pm path com.whatsapp
package:/data/app/com.whatsapp-1.apk
netbiosx@ubuntu:~$
```

Drozer

Drozer can also be used to identify APK files on the phones. From the drozer console the following command can be executed to obtain the APK path:



```
dz> run app.package.info -a com.whatsapp
Package: com.whatsapp
  Application Label: WhatsApp
  Process Name: com.whatsapp
  Version: 2.17.24
  Data Directory: /data/data/com.whatsapp
  APK Path: /data/app/com.whatsapp-1.apk
  UID: 10062
  GID: [3002, 1006, 3003, 1015, 1028]
  Shared Libraries: [/system/framework/com.google.android.maps.jar]
  Shared User ID: null
  Uses Permissions:
  - android.permission.ACCESS_COARSE_LOCATION
  - android.permission.ACCESS_FINE_LOCATION
  - android.permission.ACCESS_NETWORK_STATE
```

Conclusion

This article explained various scenarios on how to identify and retrieve APK files on the phone or from the Google Play Store. This is an important step in the process of reverse engineering any mobile application.