CODIGO DEFENSIVO

PRÁCTICA #5

DESARROLLO DE SOFTWARE SEGURO

- Jorge Ibarra Peña
- 20310025
- 7°P
- Desarrollo de Software
- 09/11/2023
- CETI COLOMOS

PRÁCTICA 5: “CONECTAR EQUIPOS A TRAVÉS DE NAT”

Objetivo: Conectar los equipos de Linux y metasploitable en una red NAT para realizar las prácticas de Código Defensivo

¿Qué es una red NAT?

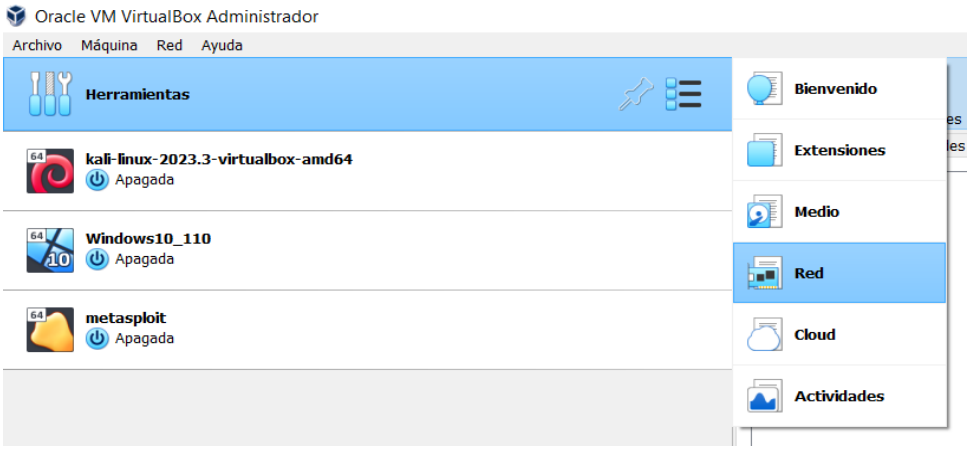
El modo NAT (Network Address Translator), en las máquinas virtuales, es un sistema que crea una subred con una dirección IP diferente a la del router de la máquina anfitriona. Si no se activa este modo de red, nuestra máquina virtual obtendrá una dirección IP que coincidirá con el punto de conexión físico a internet. En otras palabras, sería como si este ordenador estuviese conectado por cable a dicho punto.

¿Cómo funciona la red NAT?

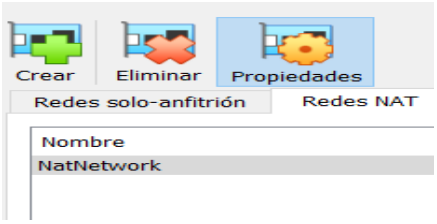
su trabajo consiste en tomar una dirección IP privada y traducirla a una dirección IP pública o viceversa. Se usa cuando necesitamos que nuestros dispositivos en la red (con IP privadas) se comuniquen a través de internet.

Procedimiento:

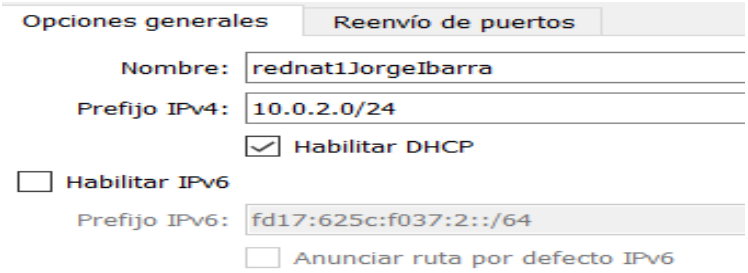
Abrimos nuestra VirtualBox y nos vamos a la sección de herramientas en el apartado de red:



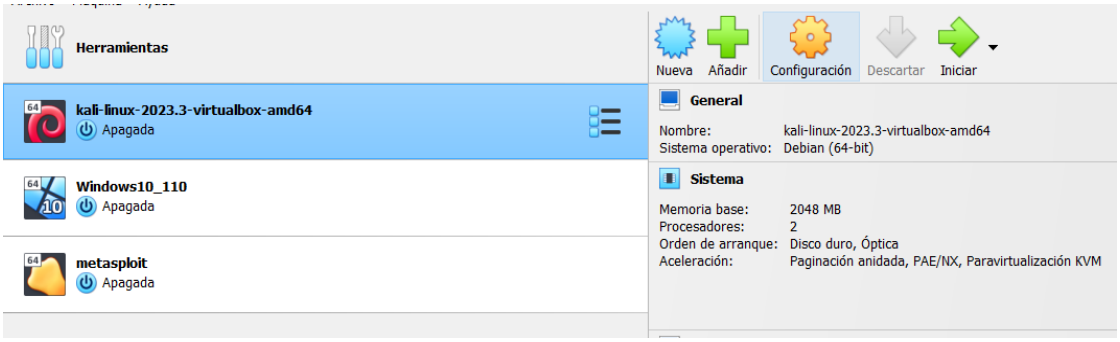
Una vez en red, nos dirigimos a la pestaña de “Redes NAT” y acto seguido presionamos el botón de “Crear”



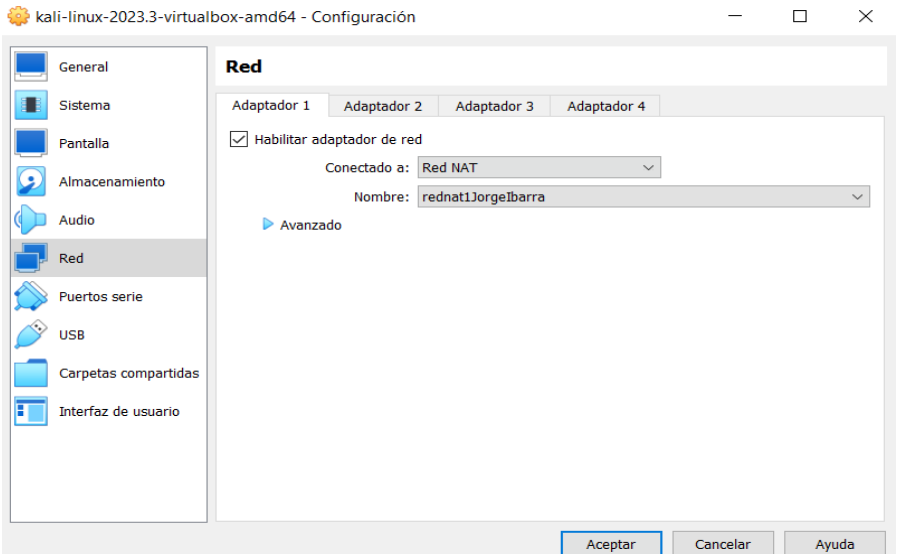
A su vez en este mismo apartado cambiamos el nombre de nuestra red y le pondremos la siguiente:



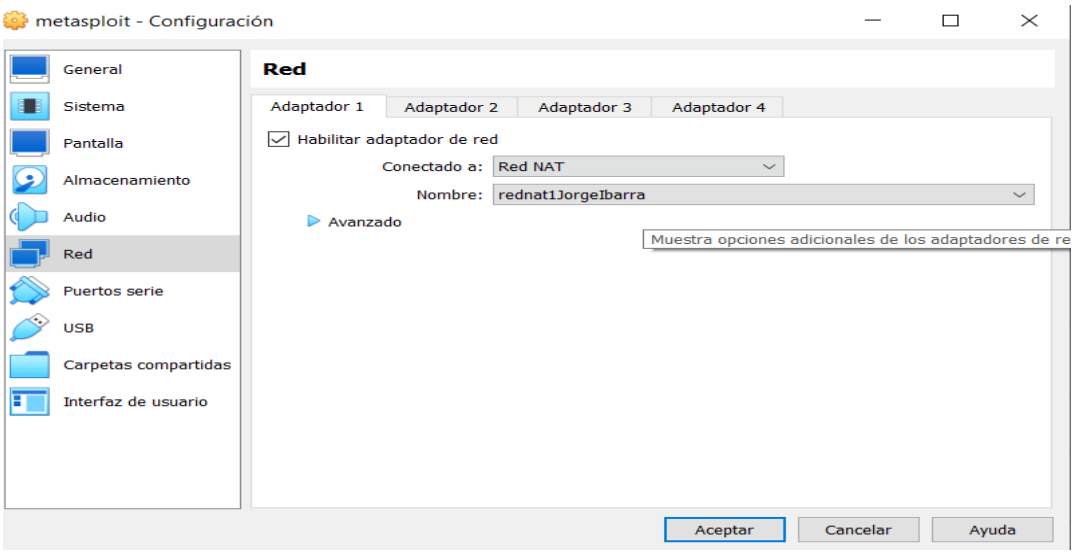
Una vez terminado este proceso, ahora ingresamos las configuraciones, pero ahora a nuestro Kali Linux y nuestro Metasploitable, para eso nos dirigimos a "Configuración" en nuestra máquina virtual



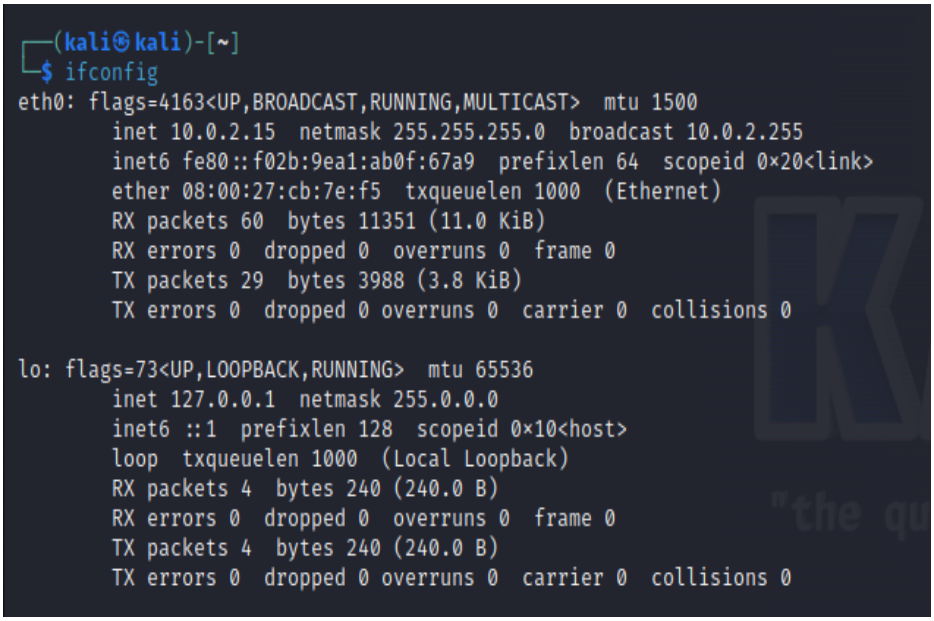
Nos vamos al apartado de red y cambiamos la opción a Red NAT



Y aplicamos la misma configuración, pero en metasploit:



Encendemos nuestras máquinas virtuales y nos vamos al CMD o mejor dicho Terminal del sistema operativo y escribimos “Ifconfig” para conocer nuestra ip



Hacemos lo mismo, pero ahora en Kali Linux, para eso ponemos el mismo comando y ahora ya tenemos las dos IP’s las cuales son:

- Kali Linux: 10.0.2.15
- Metasploitable: 10.0.2.4

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:73:88:f4
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe73:88f4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:49 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8487 (8.2 KB)  TX bytes:13279 (12.9 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:168 errors:0 dropped:0 overruns:0 frame:0
          TX packets:168 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:56553 (55.2 KB)  TX bytes:56553 (55.2 KB)
```

Ahora realizamos el ping para comprobar que realmente la conexión es exitosa, para eso lo probamos primero en Kali Linux con el comando “Ping” y la IP de Metasploit:

```
┌──(kali㉿kali)-[~]
└─$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=2.66 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=5.47 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=3.60 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=3.37 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=3.99 ms
64 bytes from 10.0.2.4: icmp_seq=6 ttl=64 time=4.08 ms
^X64 bytes from 10.0.2.4: icmp_seq=7 ttl=64 time=2.20 ms
^C
— 10.0.2.4 ping statistics —
7 packets transmitted, 7 received, 0% packet loss, time 6012ms
rtt min/avg/max/mdev = 2.204/3.623/5.467/0.981 ms
```

Y finalmente aplicamos el mismo ping, pero ahora con la IP de Kali Linux:

```
sfadmin@metasploitable:~$ ping 10.0.2.15
ING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
4 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.01 ms
4 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=3.63 ms
4 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=2.53 ms
4 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=1.25 ms
4 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=6.92 ms
4 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=6.72 ms
4 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=2.66 ms
4 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=1.72 ms
4 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=4.73 ms
4 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=3.67 ms
4 bytes from 10.0.2.15: icmp_seq=11 ttl=64 time=1.47 ms
4 bytes from 10.0.2.15: icmp_seq=12 ttl=64 time=2.16 ms
4 bytes from 10.0.2.15: icmp_seq=13 ttl=64 time=11.3 ms
4 bytes from 10.0.2.15: icmp_seq=14 ttl=64 time=2.01 ms
4 bytes from 10.0.2.15: icmp_seq=15 ttl=64 time=1.85 ms
4 bytes from 10.0.2.15: icmp_seq=16 ttl=64 time=4.04 ms
4 bytes from 10.0.2.15: icmp_seq=17 ttl=64 time=0.880 ms
4 bytes from 10.0.2.15: icmp_seq=18 ttl=64 time=16.2 ms

-- 10.0.2.15 ping statistics ---
8 packets transmitted, 18 received, 0% packet loss, time 17127ms
tt min/avg/max/mdev = 0.880/4.164/16.283/3.905 ms
sfadmin@metasploitable:~$
```

CONCLUSIÓN

Está práctica ha sido sencilla debido a que esto ya sabíamos como realizarlo y el procedimiento es realmente sencillo ya que solamente es hacer unos pocos clics y conocerte los comandos para ejecutarlos en las máquinas virtuales y así saber si la conexión es exitosa o realmente hay algo que haz hecho mal.