

Firewalls y DMZ

Proyecto de Seguridad y Alta Disponibilidad
Gonzalo Tudela Chavero

ÍNDICE DE CONTENIDOS

ENUNCIADO	1
ESQUEMA DE RED	2
VM UBUNTU SERVER (INTERNAL FIREWALL/ROUTER).....	2
Configuración Ubuntu ens33 y ens36:	2
Configuración forwarding Ubuntu Server:.....	3
Pruebas de conectividad Ubuntu Server:	3
VM CENTOS (DMZ FIREWALL/ROUTER).....	3
Configuración CentOS de ens33:	3
Configuración CentOS de ens36:	3
Configuración de ruta estática para CentOS ens36:	3
Rutas finales para CentOS:.....	4
Configuración forwarding CentOS:.....	4
Pruebas de conectividad CentOS:	4
Instalación de iptables services en CentOS:	4
NAT MASQUERADE:	5
Almacenamos la configuración de forma permanente:	5
VM WINDOWS 10 (DMZ System)	5
Windows 10 RED:.....	5
Windows 10 XAMPP:	6
Windows 10 rutas:	6
Pruebas conectividad Windows 10:	7
VM KALI (RED INTERNA)	8
Kali configuración de red:	8
Pruebas conectividad Kali Linux:	8
ACLARACIONES IPTABLES:	9
IPTABLES – Red Interna:	9
IPTABLES – DMZ:	9
OBJETIVOS IP TABLES:	9
Los usuarios de la red interna podrán utilizar los servicios de los servidores de la DMZ.	9
Desde Internet se podrá acceder a los servicios de la DMZ, pero no se podrá acceder a la red interna de la organización.	10
1. Reglas en CentOS:	10
2. Preparamos la máquina que simulará un host en internet:	10
3. Procedemos a comprobar la conexión al host en internet desde Kali (Red Interna).	10
4. Comprobamos si desde internet (Windows 7) podemos alcanzar la maquina Kali:	10

Acceso a internet para Red Interna:	10
Medidas de seguridad adicionales:	11
Configuración NAT final:	11
Configuración del resto de tablas:	11
Comprobación NMAP desde internet:	12

ÍNDICE DE FIGURAS

Ilustración 1 - Arquitectura de zona desmilitarizada.	1
Ilustración 2 - Esquema de red del proyecto.....	2
Ilustración 3 - Configuración de las interfaces de red del firewall Ubuntu.	2
Ilustración 4 - Configuración del bit de enrutamiento de forma permanente.....	3
Ilustración 5 - CentOS7 configuración ens33.	3
Ilustración 6 - Configuración CentOS7 ens36.....	3
Ilustración 7 - Configuración CentOS7 ruta estática de ens36.	4
Ilustración 8 - CentOS7 rutas.	4
Ilustración 9 - CentOS7 pruebas de conectividad.	4
Ilustración 10 - Configuración de la interfaz de red de DMZ System.	5
Ilustración 11 - Configuración XAMPP.	6
Ilustración 12 - Pruebas conectividad Windows 10 DMZ.	7
Ilustración 13 - Configuración de red para la maquina Kali Linux.	8
Ilustración 14 - Pruebas de conectividad en la maquina Kali Linux (cliente red interna).....	8
Ilustración 15 - Acceso al servicio MySQL desde la maquina Kali (Red Interna).	9
Ilustración 16 - Acceso al servicio WEB desde la maquina Kali (Red Interna).	9
Ilustración 17 - La máquina Kali si puede iniciar conexiones con la maquina en internet.	10
Ilustración 18 - Una maquina en la red 192.168.200.0/24 (internet) no puede iniciar la conexión a la red Interna. ..	10
Ilustración 19 - Consulta desde Kali a internet real.....	10
Ilustración 20 - Tabla NAT final CentOS.....	11
Ilustración 21 - Configuración de la tabla FORWARD.	11
Ilustración 22 - Contenido de /etc/sysconfig/iptables.....	12
Ilustración 23 - Resultado NMAP desde internet.....	12

ENUNCIADO

Se quiere montar una zona desmilitarizada, DMZ, con dos routers firewall sobre IPTables en los sistemas operativos Ubuntu Server y CentOS7.

El esquema de la DMZ será el siguiente:

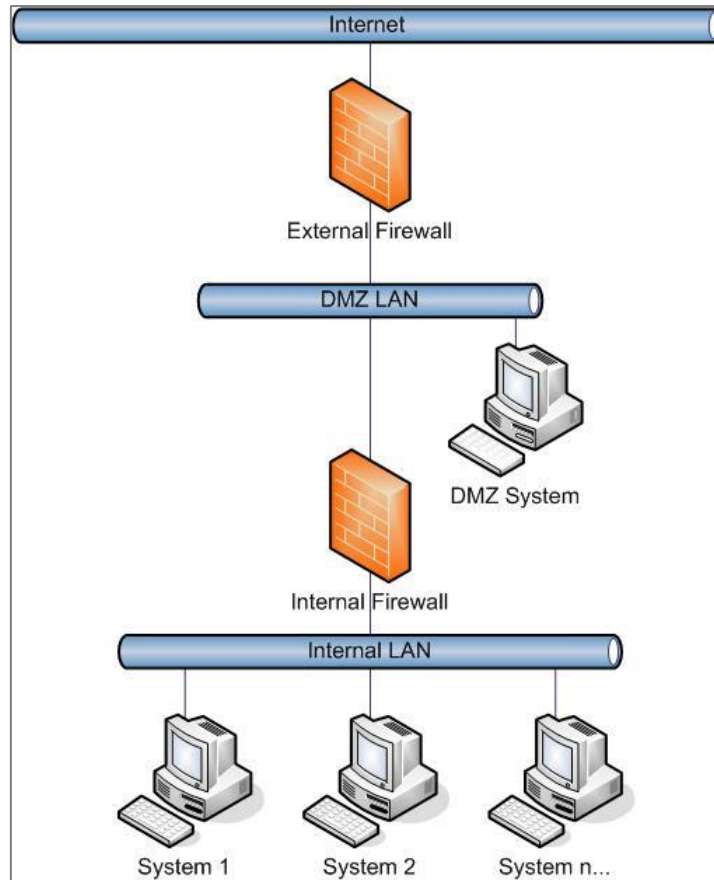


Ilustración 1 - Arquitectura de zona desmilitarizada.

Los servidores que colocarán en la DMZ serán de base de datos (MySQL) y Web.

Los usuarios de la red interna podrán utilizar los servicios de los servidores de la DMZ. Desde Internet se podrá acceder a los servicios de la DMZ, pero no se podrá acceder a la red interna de la organización. Por el contrario, los usuarios de la organización (red interna) sí podrán acceder a Internet, ya que los servidores de correo electrónico de la organización se encuentran en Internet, no es la DMZ. Es necesario que los equipos de la organización puedan actualizarse.

ESQUEMA DE RED

El siguiente esquema de red muestra la previsión que se ha hecho de como se configurará todo el ejercicio, para simplificar el esquema una vez que salimos de VMnet8 se han omitido el resto de topología ya que esta información fuera del objetivo de este proyecto.

A efectos prácticos supondremos que tras la interfaz ENS33 de la maquina CentOS se encuentra el acceso a internet, hay que fijarse en que el dispositivo NAT como la documentación de VMware lo llama, no tiene el rombo de interfaz de red en su cable ya que en si mismo es una interfaz que existe tanto en el host real en forma de adaptador de red y como una boca del switch con la obligatoriedad de ser la primera dirección de la red VMnet8.

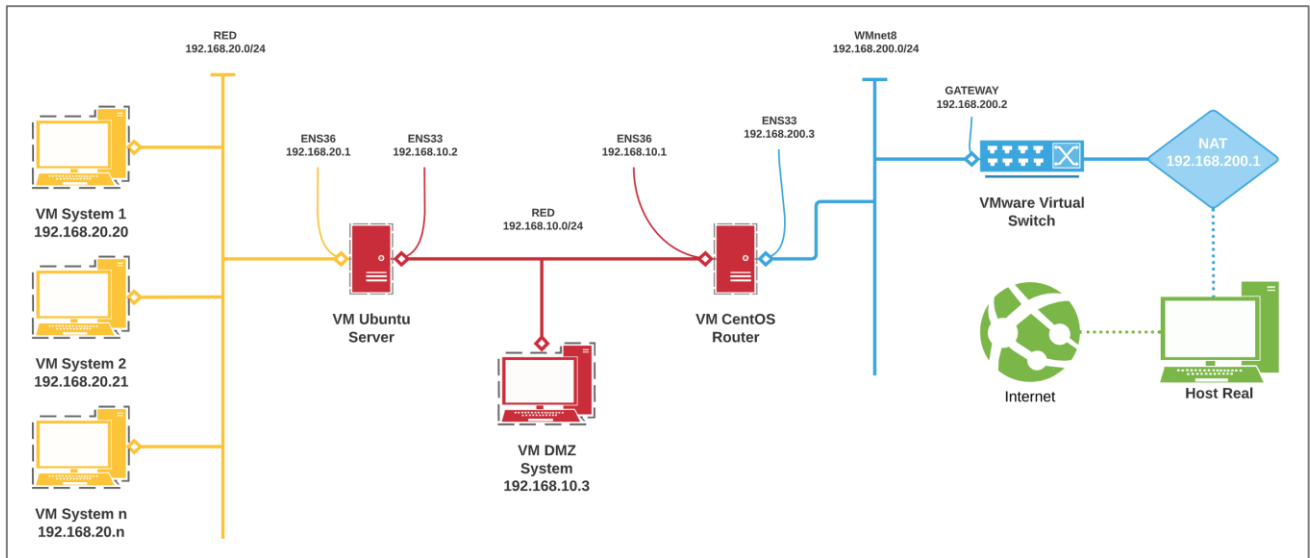


Ilustración 2 - Esquema de red del proyecto.

VM UBUNTU SERVER (INTERNAL FIREWALL/ROUTER)

Creamos esta maquina virtual con 2 interfaces de red en segmentos LAN diferentes que llamaremos según la nomenclatura de la **Ilustración 1 - Arquitectura de zona desmilitarizada**.

Por lo que tendrá su interfaz ens33 en "Segmento DMZ" y ens36 en "Segmento Interno".

Configuración Ubuntu ens33 y ens36:

(Debido a las pruebas realizadas en VMware Workstation la interfaz ens36 paso a llamarse ens37).

```
GNU nano 2.9.3 /etc/netplan/50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: false
      addresses: [192.168.20.1/24]
      gateway4: 192.168.20.1
    ens36:
      dhcp4: false
      addresses: [192.168.10.2/24]
      gateway4: 192.168.10.1
```

Ilustración 3 - Configuración de las interfaces de red del firewall Ubuntu.

Configuración forwarding Ubuntu Server:

Habilitamos en bit de enrutamiento de forma permanente editando `/etc/sysctl.conf` y asignamos a `net.ipv4.ip_forward` el valor 1.

```
GNU nano 2.9.3 /etc/sysctl.conf Modified
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Ilustración 4 - Configuración del bit de enrutamiento de forma permanente.

Esta red no necesita de ninguna ruta estática puesto que para ir a cualquier otro sitio hemos de pasar por la puerta de enlace.

Pruebas de conectividad Ubuntu Server:

(realizadas después de crear el resto de infraestructura)

VM CENTOS (DMZ FIREWALL/ROUTER)

Esta maquina dispondrá de 2 interfaces de red, ens36 configurada en el segmento DMZ y ens33 como NAT sin servicio DHCP. Deshabilitamos su firewall mediante `systemctl disable firewalld` y lo paramos `systemctl stop firewalld`.

Configuración CentOS de ens33:

```
GNU nano 2.3.1 Fichero: ifcfg-ens33
TYPE=Ethernet
BOOTPROTO=none
IPV6INIT=no
NAME=ens33
UUID=105d0ab1-3f14-4cd8-bdaa-93458b561b2a
DEVICE=ens33
ONBOOT=yes
HWADDR=00:0C:29:DD:59:3C
IPADDR=192.168.200.3
PREFIX=24
DEFROUTE=yes
GATEWAY=192.168.200.2
```

Necesitamos que el gateway se añada como ruta por defecto, le pondra la misma métrica que la red en la que está ens33, en el caso de CentOS 7 es 100.

Ilustración 5 - CentOS7 configuración ens33.

Configuración CentOS de ens36:

```
GNU nano 2.3.1 Fichero: ifcfg-ens36
HWADDR=00:0C:29:DD:59:46
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.10.1
PREFIX=24
IPV6INIT=no
NAME=ens36
UUID=c08d86a1-783b-33c3
ONBOOT=yes
DEFROUTE=no
```

En el caso de esta interfaz no necesitamos que se añada una ruta por defecto, ni si quiera hemos configurado un gateway, los host que necesitamos estan en su red y la ruta por defecto al resto la añadió ens33, recordemos que las rutas son compartidas por todas las interfaces. Solo añadiremos una ruta permanente para la red 192.168.20.0/24

Ilustración 6 - Configuración CentOS7 ens36.

Configuración de ruta estática para CentOS ens36:

Es muy importante configurar las rutas permanentes en el archivo de la interfaz a la que afectan, es decir, si configuramos la ruta 192.168.20.0 en el archivo `route-ens33` veremos la ruta, pero no funcionará.

Necesitamos decirle la ruta en el archivo correspondiente a la interfaz a la que afecta, en este caso la ruta para llegar a 192.168.20.0/24 pasa por 192.168.20.2 y la interfaz de salida será ens36 y esto deberá aparecer en el archivo route-ens36, hemos de fijarnos en la métrica 102.

```
GNU nano 2.3.1 Fichero: route-ens36
192.168.20.0/24 via 192.168.10.2 dev ens36 metric 102
```

Ilustración 7 - Configuración CentOS7 ruta estática de ens36.

Rutas finales para CentOS:

El resultado de las rutas de esta máquina es el siguiente:

```
[root@GonCentOS network-scripts]# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.200.2  0.0.0.0         UG    100    0      0 ens33
192.168.10.0     0.0.0.0        255.255.255.0   U     101    0      0 ens36
192.168.20.0     192.168.10.2   255.255.255.0   UG    102    0      0 ens36
192.168.200.0    0.0.0.0        255.255.255.0   U     100    0      0 ens33
[root@GonCentOS network-scripts]#
```

Ilustración 8 - CentOS7 rutas.

Configuración forwarding CentOS:

Configuramos el bit de enrutamiento de forma permanente editando el archivo /etc/sysctl.conf añadiendo la sentencia: `ip.ipv4.ip_forward=1`.

Pruebas de conectividad CentOS:

En la siguiente figura podemos ver como la maquina tiene comunicación con el resto de la red incluso con internet.

Se creo una maquina Kali dentro de la red 192.168.20.0/24 para poder comprobar que llegábamnos a clientes del router Ubuntu.

```
[root@GonCentOS network-scripts]# ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=63 time=0.348 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=63 time=0.336 ms
[root@GonCentOS network-scripts]# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=64 time=0.152 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=64 time=0.166 ms
[root@GonCentOS network-scripts]# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.217 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.173 ms
[root@GonCentOS network-scripts]# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=128 time=0.236 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=128 time=0.172 ms
[root@GonCentOS network-scripts]# ping www.google.es
PING www.google.es (172.217.168.163) 56(84) bytes of data.
64 bytes from mad07s10-in-f3.1e100.net (172.217.168.163): icmp_seq=1 ttl=128 time=25.3 ms
64 bytes from mad07s10-in-f3.1e100.net (172.217.168.163): icmp_seq=2 ttl=128 time=25.3 ms
```

Ilustración 9 - CentOS7 pruebas de conectividad.

Instalación de iptables services en CentOS:

Para su instalación y poder utilizar configuraciones fijas de iptables hemos instalado el paquete iptables-services y lo hemos hecho arrancar automáticamente mediante los siguientes comandos:

```
yum install iptables-services
```

```
systemctl enable iptables
```

```
systemctl start iptables
```

Ahora podremos utilizar un archivo para almacenar la configuración de iptables que se leerá cada vez que arranque el sistema, este está localizado en `/etc/sysconfig/iptables`.

NAT MASQUERADE:

Para asegurar conectividad de las maquinas a internet debemos hacer NAT MASQUERADE en POSTROUTING, ya que las maquinas a partir del router CentOS no conocen la ruta para la red interna o la DMZ ya que son direcciones privadas.

Por esto deberemos hacer que la IP de los paquetes que salgan de esas redes se enmascaren con la IP de la interfaz de salida ens33 en la maquina CentOS, esta interfaz esta en una red privada, la red NAT de VMware Workstation y podríamos pensar que tampoco funcionará pero esta red como vemos en el diagrama de red tiene a su vez otro dispositivo NAT conectado en el switch virtual que hace esta misma operación, por lo que en efecto hasta ese punto nuestras redes están escondidas tras 3 NAT que ocultan las direcciones IP originales si también contamos el que realiza el router real de mi casa..

```
iptables -t nat -A POSTROUTING -s 192.168.20.0/24 -o ens33 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o ens33 -j MASQUERADE
```

Estos comandos permitirán que las maquinas de esas redes de origen (-s) figuren con la IP de salida de ens33 ante la puerta de enlace a internet ya que de otra forma esta puerta de enlace no sabría como llegar a dichas maquinas ya que no conoce su ubicación (están en una red privada).

Almacenamos la configuración de forma permanente:

```
iptables-save > /etc/sysconfig/iptables
```

VM WINDOWS 10 (DMZ System)

Windows 10 RED:

Este sistema tendrá su interfaz de red en el segmento DMZ y dispondrá de servicios MySQL y WEB que proporcionaremos con una solución XAMPP.

Ilustración 10 - Configuración de la interfaz de red de DMZ System.

Windows 10 XAMPP:

Instalamos XAMPP y configuramos los servicios MySQL y Apache de forma permanente cada vez que reinicie la máquina.

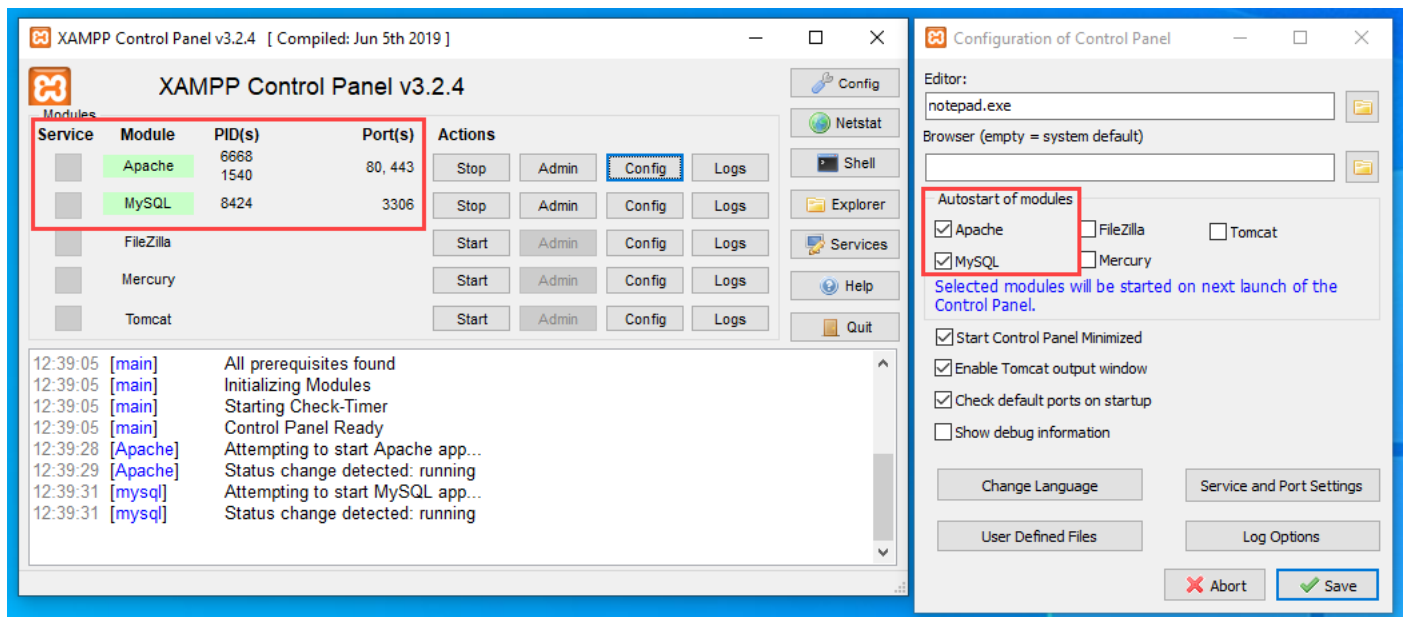


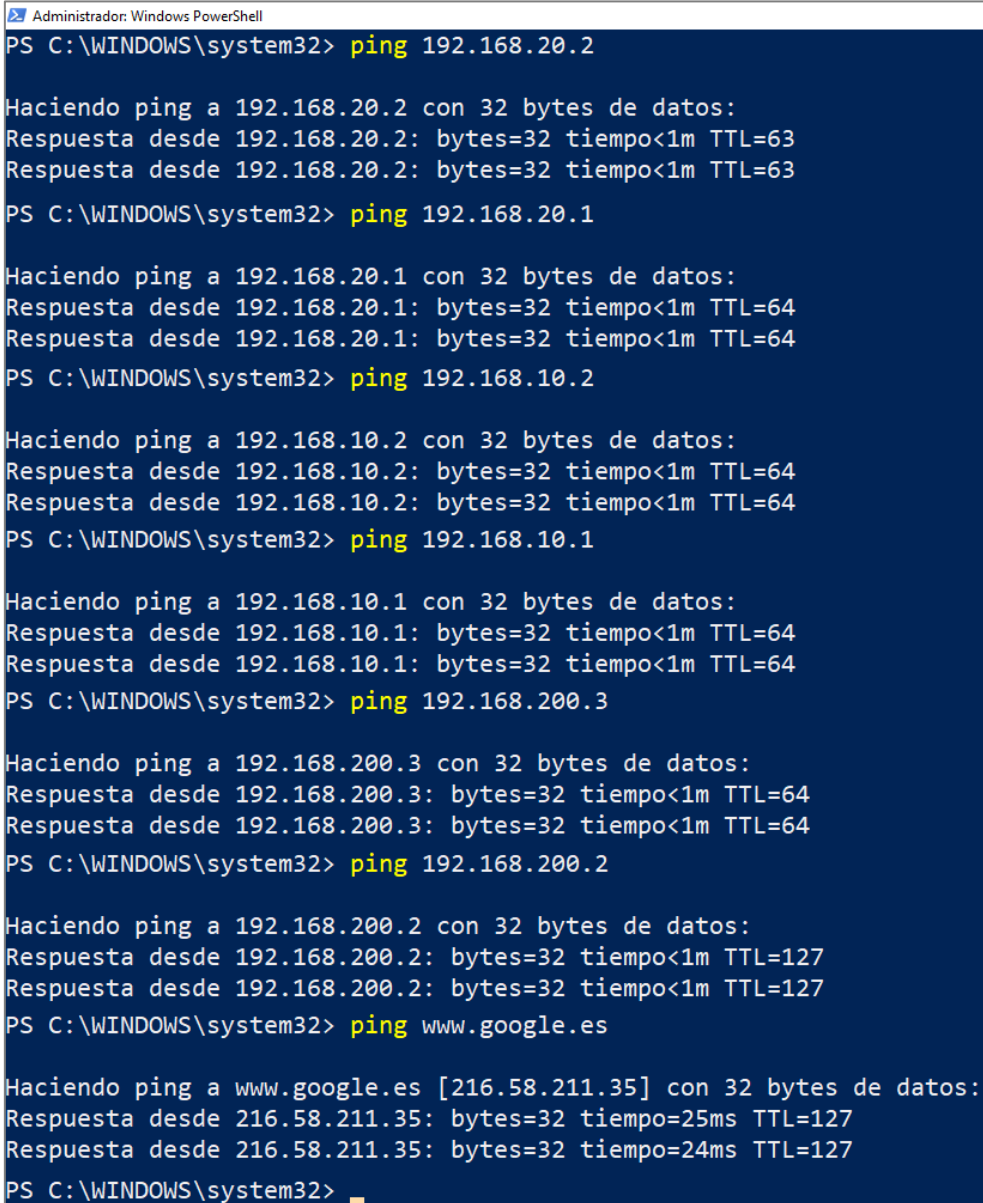
Ilustración 11 - Configuración XAMPP.

Windows 10 rutas:

Añadimos la ruta persistente (-p) para que esta máquina pueda alcanzar la red 192.168.20.0/24 con el siguiente comando:

```
route -p add 192.168.20.0 mask 255.255.255.0 192.168.10.2 if 5
```

Conocemos que el número de interfaz "if" es 5 ya que anteriormente hemos hecho un `route print` donde hemos visto las rutas y las interfaces disponibles.

Pruebas conectividad Windows 10:

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> ping 192.168.20.2

Haciendo ping a 192.168.20.2 con 32 bytes de datos:
Respuesta desde 192.168.20.2: bytes=32 tiempo<1m TTL=63
Respuesta desde 192.168.20.2: bytes=32 tiempo<1m TTL=63
PS C:\WINDOWS\system32> ping 192.168.20.1

Haciendo ping a 192.168.20.1 con 32 bytes de datos:
Respuesta desde 192.168.20.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.20.1: bytes=32 tiempo<1m TTL=64
PS C:\WINDOWS\system32> ping 192.168.10.2

Haciendo ping a 192.168.10.2 con 32 bytes de datos:
Respuesta desde 192.168.10.2: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.2: bytes=32 tiempo<1m TTL=64
PS C:\WINDOWS\system32> ping 192.168.10.1

Haciendo ping a 192.168.10.1 con 32 bytes de datos:
Respuesta desde 192.168.10.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.1: bytes=32 tiempo<1m TTL=64
PS C:\WINDOWS\system32> ping 192.168.200.3

Haciendo ping a 192.168.200.3 con 32 bytes de datos:
Respuesta desde 192.168.200.3: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.200.3: bytes=32 tiempo<1m TTL=64
PS C:\WINDOWS\system32> ping 192.168.200.2

Haciendo ping a 192.168.200.2 con 32 bytes de datos:
Respuesta desde 192.168.200.2: bytes=32 tiempo<1m TTL=127
Respuesta desde 192.168.200.2: bytes=32 tiempo<1m TTL=127
PS C:\WINDOWS\system32> ping www.google.es

Haciendo ping a www.google.es [216.58.211.35] con 32 bytes de datos:
Respuesta desde 216.58.211.35: bytes=32 tiempo=25ms TTL=127
Respuesta desde 216.58.211.35: bytes=32 tiempo=24ms TTL=127
PS C:\WINDOWS\system32>
```

Ilustración 12 - Pruebas conectividad Windows 10 DMZ.

VM KALI (RED INTERNA)

Creamos una maquina virtual con Kali Linux, configuramos su adaptador de red en VMware Workstation como perteneciente al "Segmento Interno" y le asignamos una IP estática dentro de la red 192.168.20.0/24, en este caso 192.168.20.2, para finalizar su puerta de enlace se añadirá a la tabla de rutas y esta será 192.168.20.1

Kali configuración de red:

La configuración del DNS la hacemos editando el archivo `/etc/resolv.conf` y en el introduciremos `nameserver 1.1.1.1` por ejemplo.

Para la configuración de la interfaz de red hemos utilizado la siguiente:

```
GNU nano 4.5 /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
auto eth0
iface eth0 inet static
address 192.168.20.2
gateway 192.168.20.1
```

Ilustración 13 - Configuración de red para la maquina Kali Linux.

Pruebas conectividad Kali Linux:

```
root@lightsaber:~# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=64 time=0.197 ms
root@lightsaber:~# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.172 ms
root@lightsaber:~# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=63 time=0.341 ms
root@lightsaber:~# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.178 ms
root@lightsaber:~# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=127 time=0.434 ms
root@lightsaber:~# ping 192.168.200.3
PING 192.168.200.3 (192.168.200.3) 56(84) bytes of data.
64 bytes from 192.168.200.3: icmp_seq=1 ttl=63 time=0.330 ms
root@lightsaber:~# ping 192.168.200.2
PING 192.168.200.2 (192.168.200.2) 56(84) bytes of data.
64 bytes from 192.168.200.2: icmp_seq=1 ttl=126 time=0.487 ms
root@lightsaber:~# ping www.google.es
PING www.google.es (172.217.168.163) 56(84) bytes of data.
64 bytes from mad07s10-in-f3.1e100.net (172.217.168.163): icmp_seq=1 ttl=126 time=25.5 ms
```

Ilustración 14 - Pruebas de conectividad en la maquina Kali Linux (cliente red interna).

ACLARACIONES IPTABLES:

IPTABLES – Red Interna:

Para las máquinas de la red interna no vamos a establecer reglas de filtrado para el tráfico proveniente de internet ya que deberíamos definir con anterioridad una serie de políticas de uso de estas máquinas, establecer las reglas en las tablas iptables como DROP por defecto e ir abriendo puertos mediante NAT en los casos en los que estas máquinas esperan conexiones para actualizarse, por ejemplo, substituyendo estos con números de puerto privados, es decir por encima de 49152.

También habríamos de estudiar si las máquinas que hacen de router deberían o no contar con reglas en sus tablas INPUT y OUTPUT para regular la comunicación con las máquinas de las redes que conectan o sencillamente limitarse a enrutar, o si alguna de estas máquinas será utilizada para administrar/supervisar la infraestructura...etc.

IPTABLES – DMZ:

En este caso estamos ante 3 dispositivos que realizan NAT hasta que nuestras conexiones se presentan con la IP pública que asigna mi proveedor de internet en estos momentos la 90.94.140.14. cualquier cliente que necesite llegar a la red DMZ desde internet tendrá que pasar por el router físico el cual deberá conocer a quien enviar el tráfico al puerto 80, 443 y 3306 (en este último caso lo cambiaríamos por otro menos obvio).

Para simplificar la aplicación de estas reglas vamos a suponer que internet es la red VMware NAT (192.168.200.0/24)

OBJETIVOS IP TABLES:

Los usuarios de la red interna podrán utilizar los servicios de los servidores de la DMZ.

```
root@lightsaber:~# mysql -u root -h 192.168.10.3 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.4.10-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Ilustración 15 – Acceso al servicio MySQL desde la máquina Kali (Red Interna).

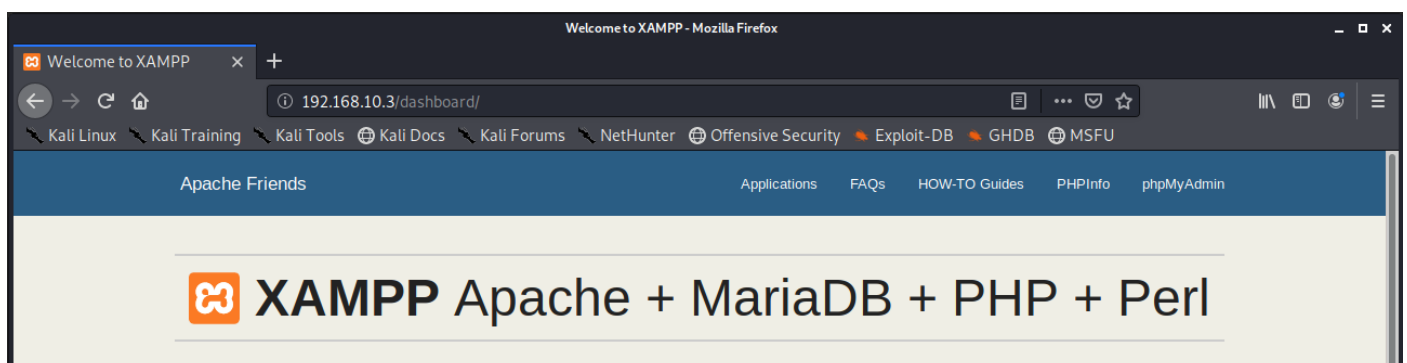


Ilustración 16 – Acceso al servicio WEB desde la máquina Kali (Red Interna).

Desde Internet se podrá acceder a los servicios de la DMZ, pero no se podrá acceder a la red interna de la organización.

1. Reglas en CentOS:

Ya que el estado por defecto es aceptar solo hemos de añadir esta regla, que evita la creación de nuevas conexiones en cualquier protocolo y puerto con origen la red (internet) 192.168.200.0/24 y destino 192.168.20.0/24, pero si permite a los hosts de la red interna salir a internet cuando son ellos los que inician la conexión.

```
iptables -A FORWARD -s 192.168.200.0/24 -d 192.168.20.0/24 -m state --state NEW -j DROP
```

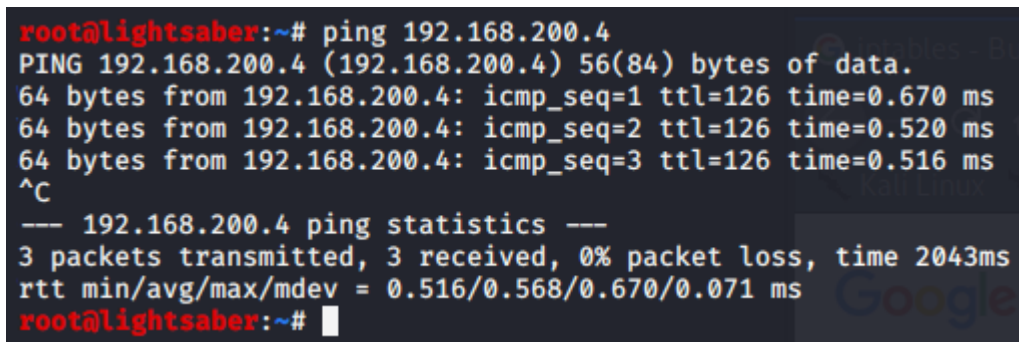
2. Preparamos la máquina que simulará un host en internet:

Para simular esta situación creamos un host Windows 7 (192.168.200.4) en la red Internet (192.168.200.0/24) al que hemos añadido las rutas necesarias para las pruebas:

```
route add 192.168.10.0 mask 255.255.255.0 192.168.200.3 if 11
```

```
route add 192.168.20.0 mask 255.255.255.0 192.168.200.3 if 11
```

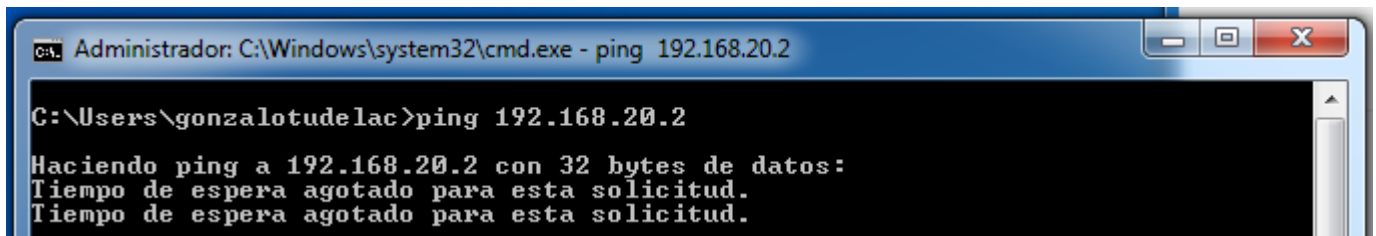
3. Procedemos a comprobar la conexión al host en internet desde Kali (Red Interna).



```
root@lightsaber:~# ping 192.168.200.4
PING 192.168.200.4 (192.168.200.4) 56(84) bytes of data:
64 bytes from 192.168.200.4: icmp_seq=1 ttl=126 time=0.670 ms
64 bytes from 192.168.200.4: icmp_seq=2 ttl=126 time=0.520 ms
64 bytes from 192.168.200.4: icmp_seq=3 ttl=126 time=0.516 ms
^C
--- 192.168.200.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.516/0.568/0.670/0.071 ms
root@lightsaber:~#
```

Ilustración 17 - La máquina Kali si puede iniciar conexiones con la maquina en internet.

4. Comprobamos si desde internet (Windows 7) podemos alcanzar la maquina Kali:



```
Administrador: C:\Windows\system32\cmd.exe - ping 192.168.20.2

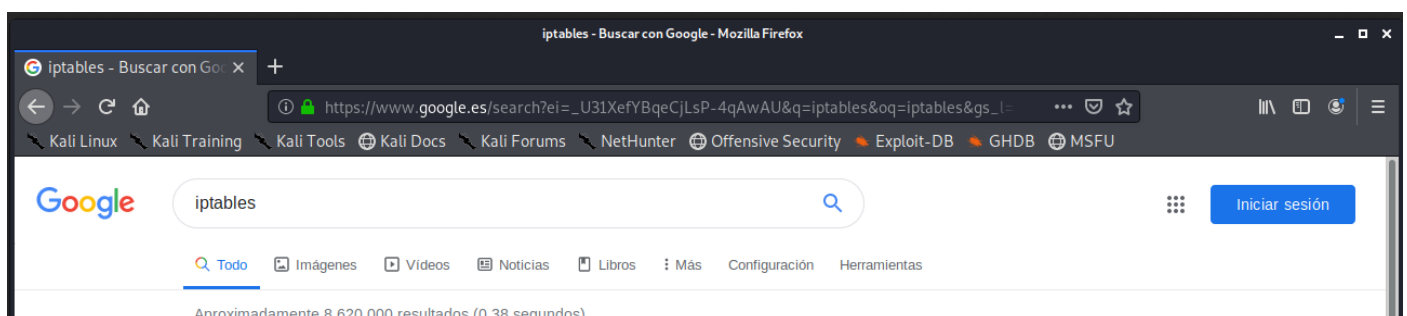
C:\Users\gonzalo tudela>ping 192.168.20.2

Haciendo ping a 192.168.20.2 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
```

Ilustración 18 - Una maquina en la red 192.168.200.0/24 (internet) no puede iniciar la conexión a la red Interna.

Acceso a internet para Red Interna:

Los usuarios de la organización (red interna) sí podrán acceder a Internet, ya que los servidores de correo electrónico de la organización se encuentran en Internet, no en la DMZ.



The screenshot shows a Mozilla Firefox browser window with the address bar displaying a Google search URL for 'iptables'. The search results show approximately 8,620,000 results found in 0.38 seconds. The browser's address bar and search bar are visible, along with the Google logo and search results.

Ilustración 19 - Consulta desde Kali a internet real.

Medidas de seguridad adicionales:

Evitar la exposición de puertos conocidos a internet es una medida de seguridad que vamos a tratar de implementar mediante NAT, configurando el router CentOS para que solo realice forward de conexiones al puerto privado 49152 para el caso de mysql, dicho de otra forma, los usuarios de la red interna se conectarán al sistema DMZ en su servicio MySQL mediante el puerto habitual y las conexiones desde internet deberán hacerlo al 49152.

Si el paquete tiene el destino 192.168.10.3:49152 lo enviamos al servidor MySQL al puerto 3306.

```
iptables -t nat -A PREROUTING -d 192.168.10.3 -p tcp --dport 49152 -j DNAT --to 192.168.10.3:3306
```

Si el paquete tiene el destino 192.168.10.3:3306 lo enviamos al puerto 445 que está filtrado junto con otros tantos.

```
iptables -t nat -A PREROUTING -d 192.168.10.3 -p tcp --dport 3306 -j DNAT --to 192.168.10.3:445
```

Filtrar los puertos que no queremos que se acepten conexiones (tenemos por defecto ACCEPT).

```
iptables -A FORWARD -s 192.168.200.0/24 -d 192.168.10.3 -p tcp --match multiport -dports 135,139,445,5357 -j DROP
```

Configuración NAT final:

```
iptables -t nat -L -n --line-numbers
```

```
Chain PREROUTING (policy ACCEPT)
num  target      prot opt source      destination
1    DNAT        tcp  --  0.0.0.0/0    192.168.10.3    tcp dpt:49152 to:192.168.10.3:3306
2    DNAT        tcp  --  0.0.0.0/0    192.168.10.3    tcp dpt:3306 to:192.168.10.3:445

Chain INPUT (policy ACCEPT)
num  target      prot opt source      destination

Chain OUTPUT (policy ACCEPT)
num  target      prot opt source      destination

Chain POSTROUTING (policy ACCEPT)
num  target      prot opt source      destination
1    MASQUERADE  all  --  192.168.10.0/24  0.0.0.0/0
2    MASQUERADE  all  --  192.168.20.0/24  0.0.0.0/0
[root@GonCentOS gonzalo]#
```

Ilustración 20 - Tabla NAT final CentOS.

Configuración del resto de tablas:

```
Chain INPUT (policy ACCEPT)
num  target      prot opt source      destination

Chain FORWARD (policy ACCEPT)
num  target      prot opt source      destination
1    DROP        tcp  --  192.168.200.0/24  192.168.10.3    multiport dports 135,139,445,5357

Chain OUTPUT (policy ACCEPT)
num  target      prot opt source      destination
[root@GonCentOS gonzalo]#
```

Ilustración 21 - Configuración de la tabla FORWARD.

Finalmente guardamos la configuración para que sea permanente con:

```
iptables-save > /etc/sysconfig/iptables
```

```
[root@GonCent0S gonzalo]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.21 on Sun Dec 15 01:16:09 2019
*filter
:INPUT ACCEPT [225:63610]
:FORWARD ACCEPT [2317:174684]
:OUTPUT ACCEPT [2:143]
-A FORWARD -s 192.168.200.0/24 -d 192.168.10.3/32 -p tcp -m multiport --dports 135,139,445,5357 -j DROP
COMMIT
# Completed on Sun Dec 15 01:16:09 2019
# Generated by iptables-save v1.4.21 on Sun Dec 15 01:16:09 2019
*nat
:PREROUTING ACCEPT [1118:50101]
:INPUT ACCEPT [11:1051]
:OUTPUT ACCEPT [6:429]
:POSTROUTING ACCEPT [1077:47521]
-A PREROUTING -d 192.168.10.3/32 -p tcp -m tcp --dport 49152 -j DNAT --to-destination 192.168.10.3:3306
-A PREROUTING -d 192.168.10.3/32 -p tcp -m tcp --dport 3306 -j DNAT --to-destination 192.168.10.3:445
-A POSTROUTING -s 192.168.10.0/24 -o ens33 -j MASQUERADE
-A POSTROUTING -s 192.168.20.0/24 -o ens33 -j MASQUERADE
COMMIT
# Completed on Sun Dec 15 01:16:09 2019
[root@GonCent0S gonzalo]#
```

Ilustración 22 - Contenido de /etc/sysconfig/iptables

Comprobación NMAP desde internet:

Comprobamos los puertos desde la maquina en internet Windows 7 mediante nmap:

```
nmap 192.168.10.3 -sS
```

```
C:\Users\gonzalotudelac>nmap 192.168.10.3 -sS
Starting Nmap 7.80 < https://nmap.org > at 2019-12-14 23:13 Hora estándar romanc
e
Nmap scan report for 192.168.10.3
Host is up (0.00039s latency).
Not shown: 992 closed ports
PORT      STATE      SERVICE
80/tcp    open       http
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
443/tcp   open       https
445/tcp   filtered  microsoft-ds
3306/tcp  filtered  mysql
5357/tcp  filtered  wsddapi
49152/tcp open       unknown

Nmap done: 1 IP address (1 host up) scanned in 8.40 seconds
C:\Users\gonzalotudelac>_
```

Ilustración 23 - Resultado NMAP desde internet.

Por desgracia NMAP revisa ese puerto en su escaneo por defecto, para evitar este resultado deberíamos encontrar algún puerto que no este en este escaneo y reconfigurar los pasos anteriores para que al menos en un escaneo inicial no se muestre el puerto abierto.