REDES DE ORDENADORES

PRÁCTICA II: FIREWALL CON IPTABLES

CONFIGURACIÓN DE RED LOCAL Y DMZ

SEBASTIÁN GUTIÉRREZ Y JORGE MARTÍNEZ
INGENIERÍA DEL SOFTWARE, 2ºB
U-TAD

Índice General

1. El Problema a Resolver **4**

1.1. Software utilizado en la práctica 4

2. El Diagrama de Red **5**

2.1. Explicación del Diagrama 5

3. El Firewall **6**

3.1. Reglas con iptables 6

3.2. Script firewall-setup.sh 8

4. El Cliente (Red Local) **10**

4.1. Script client-setup.sh 11

5. El Servidor (DMZ) **12**

5.1. Script server-setup.sh 12

6. Comprobaciones de Funcionamiento **14**

6.1. Acceso a Servidor desde Red Local e Internet 14

6.2. Acceso a Red Local desde Servidor e Internet 17

6.3. Acceso desde Internet a Red Local 19

7. Conclusión de la práctica y posible implementación **20**

Índice de Figuras

1. El Problema a Resolver **4**

2. El Diagrama de Red **5**

2.1. Diagrama de Red 5

3. El Firewall **6**

3.1 Reglas con iptables 6

3.1.1 Preparación de iptables 6

3.1.2 Cierre de Firewall y preparación de DNAT y SNAT 6

3.1.3 Configuración de puertos 53, 80 y 443 7

3.1.4 Configuración recíproca de puertos 53, 80 y 443 7

3.1.5 Permisos de acceso desde Internet a la DMZ por puerto 80 7

3.1.6 Permisos de acceso desde Red Local a la DMZ por puertos 80 y 22 7

3.2. Script firewall-setup.sh 8

3.2.1 Mensaje introductorio al script firewall-setup.sh 8

3.2.2 Comprobación de permisos sudo en el script firewall-setup.sh 9

3.2.3 Configuración de Default Gateway en firewall-setup.sh 9

3.2.4 Activación de IP Forwarding en el script firewall-setup.sh 9

3.2.5 Configuración de tarjetas de red en firewall-setup.sh 10

4. El Cliente (Red Local) **10**

4.1. Mensaje introductorio al script client-setup.sh 11

4.2. Configuración de tarjeta de red en client-setup.sh 11

4.3. Configuración de default gateway en client-setup.sh 12

5. El Servidor (DMZ) **12**

5.1. Configuración de tarjeta de red en server-setup.sh 12

5.2. Actualización apt e instalación de servidor SSH y Apache2 en server-setup.sh 13

5.3. Configuración de tarjeta de red en server-setup.sh 13

5.4. Configuración de default gateway en server-setup.sh 13

6. Comprobaciones de Funcionamiento **14**

6.1 Acceso a Servidor desde Red Local e Internet 14

6.1.1 Acceso a la DMZ por puerto 80 desde Internet (equipo Windows) 14

6.1.2 Acceso a la DMZ por puerto 80 desde Red Local 15

6.1.3 Acceso a la DMZ por puerto 22 desde Red Local 16
6.1.4 Configuración de PuTTY para acceder desde Internet a la DMZ via SSH 16
6.1.5 Prueba de acceso a DMZ por puerto 22 desde Internet (Windows) 17
6.2 Acceso a Red Local desde Servidor e Internet 17
6.2.1 Prueba de acceso a la Red Local desde la DMZ por puerto 22 17
6.2.2 Configuración de PuTTY para acceder desde Internet a la Red Local via SSH 18
6.2.3 Acceso a la Red Local desde Internet (Windows) por puerto 22 18
6.3 Acceso a Internet desde Red Local 19
6.3.1 Acceso a Internet HTTPS (puerto 443) desde Red Local 19
6.3.2 Acceso a Internet HTTP (puerto 80) desde Red Local 19
7. Conclusión de la práctica y posible implementación 20

1. El problema a Resolver

En esta práctica, el objetivo es diseñar una infraestructura de seguridad mediante un firewall basado en Linux para una empresa ficticia. Dicha empresa necesita una red con las siguientes características:

- Red local (30.0.0.0/8): los equipos en esta red sólo han de poder navegar por Internet (http y https), y poder acceder a la web de la intranet de la empresa.
- Red DMZ: en esta red, debe haber un servidor web Linux que alberga la página web. A este servidor sólo se puede acceder vía SSH desde la red local, y sólo puede suministrar la página web a peticiones externas o de la red local.

Todos los scripts y documentación necesaria son accesibles públicamente en el siguiente repositorio de GitHub: https://github.com/jorgemhdev/practica-firewall-iptables

Los perfiles de GitHub de los integrantes del grupo son:

- Sebastián: https://github.com/Willyfsg
- Jorge: https://github.com/jorgemhdev/

1.1. Software utilizado en la práctica

Para realizar la práctica, hemos virtualizado todo el entorno mediante VMware Workstation, utilizando tres máquinas virtuales. Tanto el cliente como el servidor se ejecutan sobre máquinas con Ubuntu 20, y el Firewall es una máquina con Kali Linux.

Tanto en Ubuntu como en Kali hemos utilizado GNU Nano como editor de texto para realizar los scripts, y Mozilla Firefox como navegador para realizar pruebas de conectividad http y https. Hemos usado OpenSSH y Apache2 en el servidor para dotarlo de funcionalidad SSH y web, respectivamente.

Para realizar el diagrama de red, el cual se abarcará en el siguiente capítulo, utilizamos Adobe Photoshop.

2. Diagrama de Red

Antes de indagar en los detalles técnicos de la práctica, es preciso mostrar un diagrama que permita visualizar de forma rápida y clara en qué consiste la misma. La siguiente figura representa la arquitectura de red que hemos diseñado para la práctica:

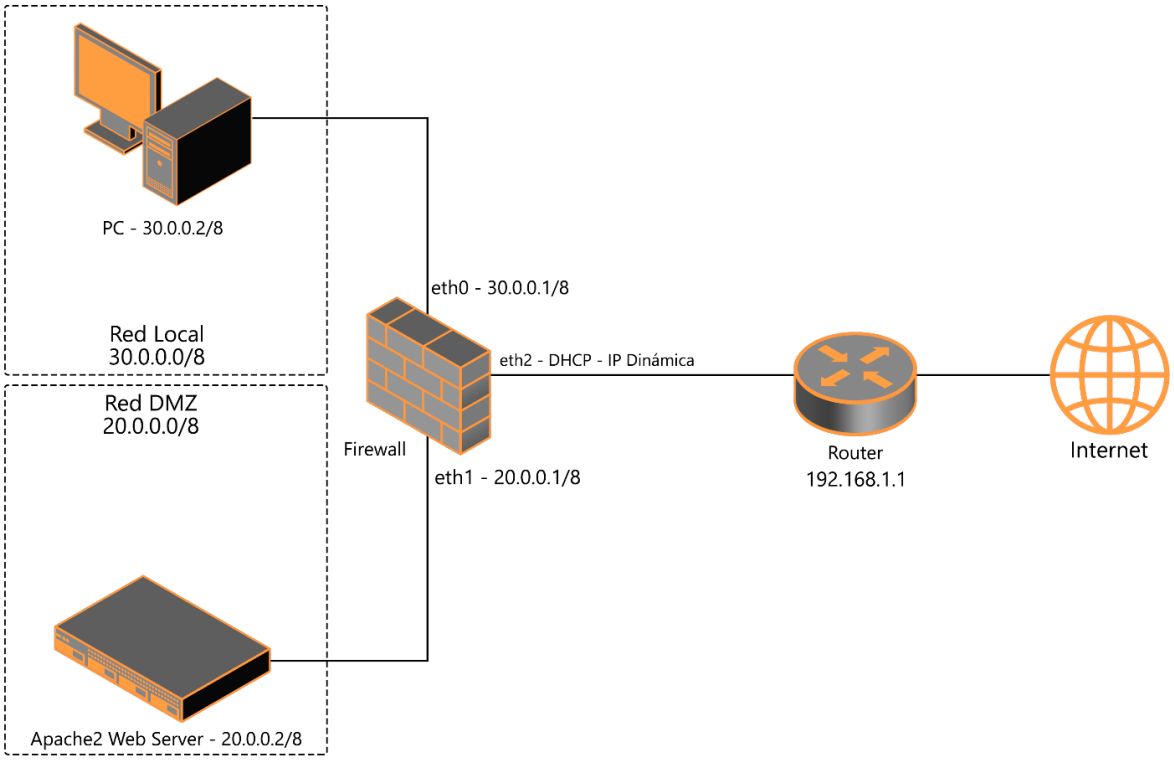

Fig. 2.1: Diagrama de Red

2.1. Explicación del Diagrama

En nuestro diagrama, el router tiene la dirección IP 192.168.1.1 y a su vez realiza la función de Gateway a Internet. El Firewall está conectado al Router, y es el punto intermedio entre la Red Local y la Red DMZ. Su IP es asignada dinámicamente por el servidor DHCP integrado en el Router.

El Firewall a su vez tiene dos interfaces de red cuyas IP son 30.0.0.1/8 y 20.0.0.1/8. A ellas se conectarán el PC de la Red Local y el Servidor de la DMZ, respectivamente.

En la Red Local (30.0.0.0/8) se encuentra un PC cuya IP es 30.0.0.2/8, y en la DMZ (20.0.0.0/8) se encuentra un servidor Apache2 cuya IP es 20.0.0.2/8.

Al ser el Firewall el punto medio entre las dos redes e Internet, podemos configurarlo para permitir y administrar el tráfico entre redes. Para ello, crearemos reglas usando iptables.

3. El Firewall

El Firewall es el dispositivo que utilizamos para crear reglas que nos permiten controlar el tráfico entre las redes de nuestra arquitectura e Internet. En esta práctica, el Firewall es una máquina virtual ejecutando Kali Linux a la que se han asignado tres interfaces de red: una para conectarse al Router, y otras dos para las dos redes.

El Firewall funciona con unas reglas definidas mediante **iptables** que dotan de la seguridad necesaria a la red de la empresa. Estas reglas previenen accesos no autorizados y contribuyen al correcto funcionamiento de la infraestructura.

3.1. Reglas con iptables

Las reglas que utilizamos para configurar el Firewall se aplican al ejecutar el script *firewall-setup.sh*, que se abarcará en la sección 3.2, y serán expuestas a continuación.

Antes de configurar las reglas, hemos de eliminar cualquier regla que pudiese haber anteriormente. También habilitamos el pre routing y el post routing.

```
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
```

Fig. 3.1.1: preparación de iptables

A continuación, cerraremos todo el acceso por defecto y habilitaremos DNAT y SNAT:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
```

Fig. 3.1.2: cierre de Firewall y activación de DNAT y SNAT

Una vez se ha cerrado todo el Firewall, vamos a permitir el tráfico únicamente por los puertos 80, 443 y 53 (http, https y DNS, respectivamente):

```
iptables -A FORWARD -i eth0 -o eth2 -p tcp --dport 80 -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -o eth2 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i eth0 -o eth2 -p udp --dport 53 -j ACCEPT
```

Fig. 3.1.3: configuración de puertos 53, 80 y 443

Añadimos reciprocidad a todas las reglas anteriores:

```
iptables -A FORWARD -i eth2 -o eth0 -p tcp -m state --state ESTABLISHED,RELATED
-j ACCEPT

iptables -A FORWARD -i eth2 -o eth0 -p udp -m state --state ESTABLISHED,RELATED
-j ACCEPT

iptables -t nat -A POSTROUTING -s 30.0.0.0/8 -o eth2 -j MASQUERADE
```

Fig. 3.1.4: configuración recíproca de puertos 53, 80 y 443

Permitimos acceso desde Internet a la DMZ (20.0.0.0/8) mediante HTTP (80):

```
iptables -A FORWARD -i eth2 -o eth1 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i eth2 -o eth1 -p tcp -m state --state ESTABLISHED,RELATED
-j ACCEPT

iptables -t nat -A PREROUTING -i eth2 -p tcp --dport 80 -j DNAT --to 20.0.0.2:80
```

Fig. 3.1.5: permisos de acceso desde Internet a la DMZ por puerto 80 (HTTP)

Permitimos el acceso a la DMZ desde la Red Local (30.0.0.0/8) mediante HTTP (80) y SSH (22):

```
iptables -A FORWARD -s 30.0.0.0/8 -d 20.0.0.0/8 -p tcp --dport=80 -j ACCEPT

iptables -A FORWARD -s 30.0.0.0/8 -d 20.0.0.0/8 -p tcp --dport=22 -j ACCEPT

iptables -A FORWARD -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Fig. 3.1.6: permisos de acceso desde Red Local a la DMZ por puertos 80 (HTTP) y 22 (SSH)

Por ultimo, muestra la tabla de iptables para verificar que las reglas se han configurado de forma satisfactoria:

```
iptables -L
```

3.2. Script *firewall-setup.sh*

Tal y como se ha comentado anteriormente, estas reglas se aplican cuando se ejecuta el script de configuración *firewall-setup.sh.* En esta sección se verá el funcionamiento de dicho script

Inicialmente, se establece a bash como el intérprete del script, y se imprime por pantalla un pequeño mensaje introductorio al script:

```
#!/bin/bash
# IMPORTANTE: este script se debe ejecutar con permisos sudo.
delaytime=1 #variable delaytime; determina el tiempo de los sleep
echo "------------------------------------------------"
echo " "
echo "Script de configuracion de Firewall para DMZ, v.0.1"
echo "Por Sebastian Gutierrez y Jorge Martinez"
echo "INSO 2B, U-Tad"
echo " "
echo "------------------------------------------------"
sleep $delaytime
echo " "
```

Fig. 3.2.1: mensaje introductorio al script firewall-setup.sh

A continuación, se comprueba si el script se está ejecutando como sudo, requisito que resulta esencial para la configuración del Firewall. Para ello, comprueba el User ID del usuario que ejecutó el script: si es 0 es root, por lo que permite la ejecución; si no lo es, entonces sale del script y pide al usuario que lo ejecute como root:

```
if [ "$EUID" -ne 0 ]; then
      echo " "
      echo "ERROR: Permisos sudo no detectados."
      echo "Por favor, ejecuta el script como root."
      echo " "
```

```
        echo "-------------------------------------------------"
        exit 1
fi
```

Fig. 3.2.2: comprobación de permisos sudo en el script firewall-setup.sh

Tras comprobar los permisos sudo, pasamos a configurar el Default Gateway. En este caso, el Router es el Gateway del Firewall, por lo que asignamos la IP 192.168.1.1 a tal función:

```
route add default gw 192.168.1.1
/etc/init.d/networking restart
def_gateway=$(/sbin/ip route | awk '/default/ { print $3 }')
echo "El default gateway configurado es: $def_gateway"
sleep $delaytime
```

Fig. 3.2.3: configuración de Default Gateway en firewall-setup.sh

La variable **def_gateway** alberga el valor de la **Gateway** configurada, y se utiliza para comprobar que se ha configurado de forma satisfactoria.

A continuación, **activamos IP Forwarding** para que el **tráfico** pueda atravesar el Firewall. Para ello, añadimos la línea **"net.ipv4.ip_forward = 1" al archivo /etc/sysctl.conf** mediante un ***echo >>*** (se usa >> para no sobreescribir el archivo y simplemente añadir la línea al final del mismo). Así, el cambio persiste reinicios.

```
echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf #cambio permanente en el archivo sysctl.conf.
sudo sysctl -p
/etc/init.d/networking restart
ipfwd_status=$(cat /proc/sys/net/ipv4/ip_forward)
if [ "$ipfwd_status" != "1" ]; then
        echo "Error activando IP Forward."
        exit 1
fi
sleep $delaytime
```

Fig. 3.2.4: activación de IP Forwarding en el script firewall-setup.sh

La variable ***ipfwd_status*** contiene el valor (0 o 1) de IP Forward; a continuación, se comprueba si la misma tiene valor 1, y si no lo tiene, detiene la ejecución del script.

A continuación, se asignan las IP adecuadas a las interfaces de red del Firewall. **La interfaz *eth0*** conectará con la **Red Local**, ***eth1*** con la **DMZ** y ***eth2*** con el **Router**. Las dos primeras tendrán IP estáticas **(30.0.0.1/8 y 20.0.0.1/8**, respectivamente) y la tercera tendrá una IP dinámica suministrada por **DHCP**.

```
ifconfig eth0 30.0.0.1/8

ifconfig eth1 20.0.0.1/8

dhclient eth2

sleep $delaytime

eth0_info=$(ip a | grep inet | grep eth0)

eth1_info=$(ip a | grep inet | grep eth1)

eth2_info=$(ip a | grep inet | grep eth2)

echo "- eth0: $eth0_info"

echo "- eth1: $eth2_info"

echo "- eth2: $eth1_info"

sleep $delaytime
```

Fig. 3.2.5: configuración de tarjetas de red en firewall-setup.sh

Las variables ***eth0_info, eth1_info*** **y** ***eth2_info*** **contienen las IP e información sobre las interfaces de red** recientemente configuradas, y sus valores son mostrados por pantalla para verificar que su configuración ha resultado satisfactoria.

A continuación, se configuran las reglas de Firewall mediante los comandos expuestos en la sección 3.1, concluyendo así la configuración del Firewall.

4. El Cliente (Red Local)

En esta práctica, una máquina virtual con Ubuntu 20 realiza la función de cliente en la Red Local (20.0.0.0/8). Esta máquina (IP 20.0.0.2/8) simula ser un equipo de la empresa con conexión a Internet y con permisos para administrar vía SSH el servidor localizado en la DMZ.

4.1. Script *client-setup.sh*

Este script se ejecuta como sudo en el cliente y lo configura automáticamente acorde a la arquitectura diseñada para la empresa.

Como el script anterior, lo primero que hace es comprobar si se está ejecutando con permisos de superusuario (UID 0). Si no los tiene, detiene su ejecución:

```
if [ "$EUID" -ne 0 ]; then
      echo " "
      echo "ERROR: Permisos sudo no detectados."
      echo "Por favor, ejecuta el script como root."
      echo " "
      echo "--------------------------------------------------"
      exit 1
fi
```

Fig. 4.1: mensaje introductorio al script client-setup.sh

Una vez se ha comprobado que el script se ejecuta con los permisos pertinentes, se configura la IP de la máquina:

```
ip a flush dev ens33
ip a add 30.0.0.2/8 dev ens33

ens33_info=$(ip a | grep inet | grep ens33)
echo "$ens33_info"

sleep $delaytime
```

Fig. 4.2: configuración de tarjeta de red en client-setup.sh

Primero descartamos cualquier IP que pudiese tener anteriormente, a continuación le asignamos la IP deseada y por último mostramos por pantalla el contenido de la variable ens33_info, que contiene información sobre la interfaz que acaba de ser configurada.

Por último, se configura el Default Gateway del cliente. En este caso, hemos de dirigir el tráfico del cliente al Firewall (30.0.0.1/8), por lo que lo configuraremos así:

```
ip route add default via 30.0.0.1
echo " "
route -n
sleep $delaytime
```

Fig. 4.3: configuración de default gateway en client-setup.sh

La variable delaytime, utilizada en scripts anteriores, contiene el valor en segundos de las pausas entre configuración y configuración. Así se puede monitorizar por encima el progreso de cada script.

5. El Servidor (DMZ)

El servidor, al igual que el cliente, es una máquina virtual con Ubuntu 20 situada en la DMZ (20.0.0.0/8). En esta máquina se han instalado apache2 y openssh-server para convertirla en servidor web administrado mediante SSH.

5.1. Script *server-setup.sh*

Este script se ejecuta como sudo en el servidor y configura automáticamente la máquina virtual acorde a la arquitectura diseñada para la empresa.

Al igual que en los scripts anteriores, se realiza una comprobación de permisos sudo al empezar:

```
if [ "$EUID" -ne 0 ]; then
      echo " "
      echo "ERROR: Permisos sudo no detectados."
      echo "Por favor, ejecuta el script como root."
      echo " "
      echo "--------------------------------------------------"
      exit 1
fi
```

Fig. 5.1: configuración de tarjeta de red en server-setup.sh

A continuación, instalamos net-tools, apache2 y openssh-server, actualizando el sistema antes:

```
apt update -y && apt upgrade -y
sleep $delaytime

apt install net-tools -y
apt install apache2 -y
sleep $delaytime

apt install openssh-server -y
sleep $delaytime
```

Fig. 5.2: actualización apt e instalación de servidor SSH y Apache2 en server-setup.sh

Después, asignamos la IP 20.0.0.2/8 al servidor, descartando primero cualquier IP que pudiese tener ya asignada:

```
ip a flush dev ens33
ip a add 20.0.0.2/8 dev ens33
ens33_info=$(ip a | grep inet | grep ens33)
echo "$ens33_info"
sleep $delaytime
```

Fig. 5.3: configuración de tarjeta de red en server-setup.sh

Al igual que en los scripts anteriores, se muestra por pantalla el contenido de la variable ens33_info, que contiene información sobre la interfaz que se acaba de configurar.

Por último, configuramos el Default Gateway del servidor, que en este caso será el Firewall (20.0.0.1/8):

```
ip route add default via 20.0.0.1
route -n
sleep $delaytime
```

Fig. 5.4: configuración de default gateway en server-setup.sh

6. Comprobaciones de Funcionamiento

6.1. Acceso a Servidor desde Red Local e Internet

Dada la configuración de nuestro Firewall, debemos poder acceder al servidor de nuestra DMZ por el puerto 80 tanto desde Internet como desde la LAN. Además, se ha de poder acceder por SSH (puerto 22) desde la LAN, para poder administrarlo.

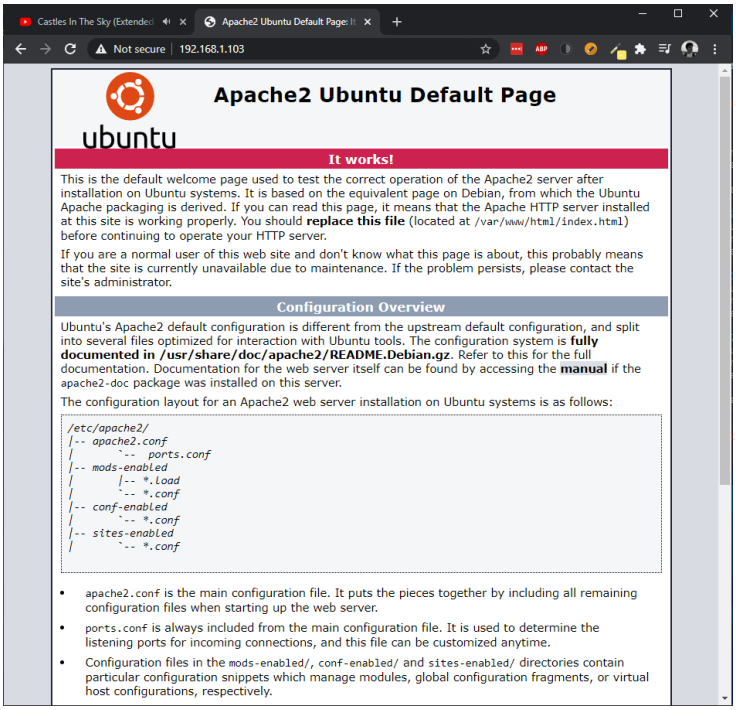

Fig. 6.1.1: prueba de acceso a la DMZ por puerto 80 desde Internet (simulado desde equipo Windows)

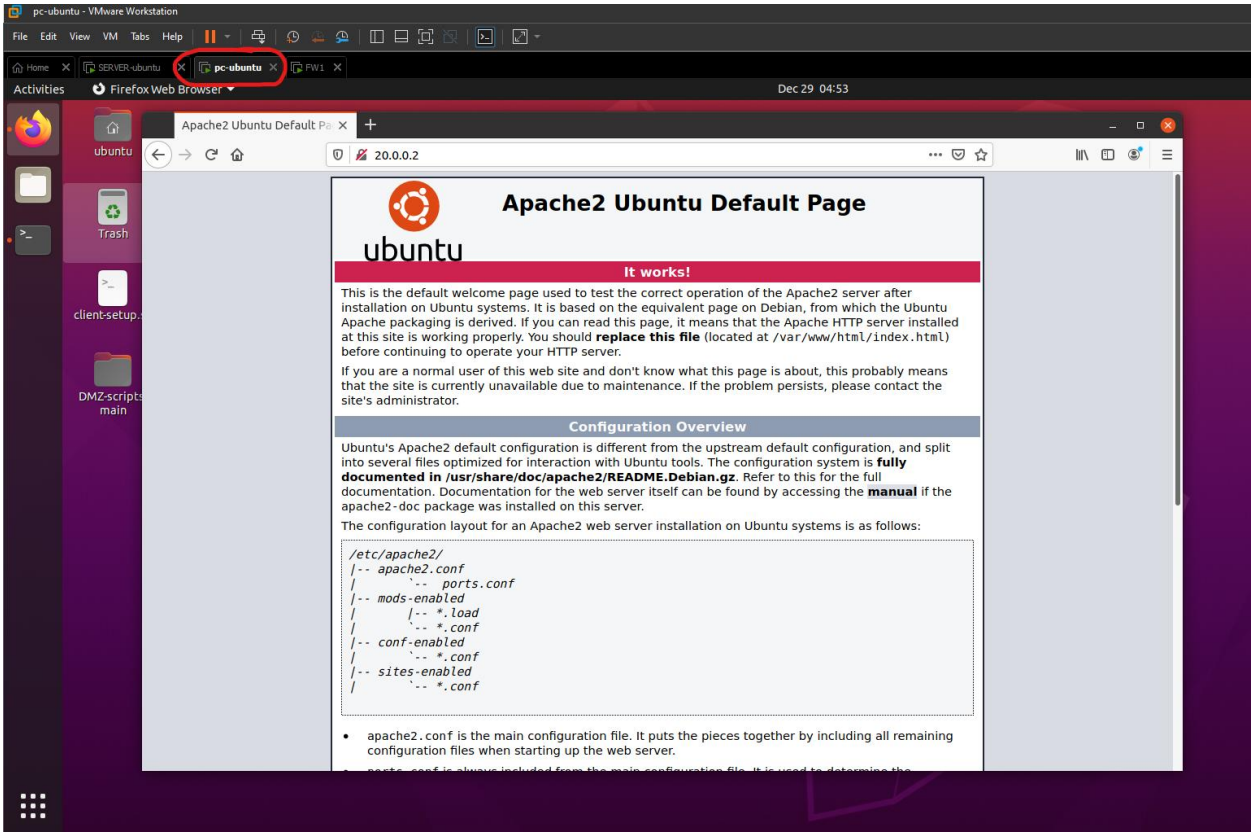

Fig. 6.1.2: prueba de acceso a la DMZ por puerto 80 desde Red Local

Como se puede ver, ambos accesos por el puerto 80 son posibles. Ahora, comprobaremos que sólo se puede acceder desde la Red Local (30.0.0.0/8) por SSH al servidor (20.0.0.2/8).

Intentamos acceder desde la Red Local (máquina “pc-ubuntu”, IP 30.0.0.2/8):

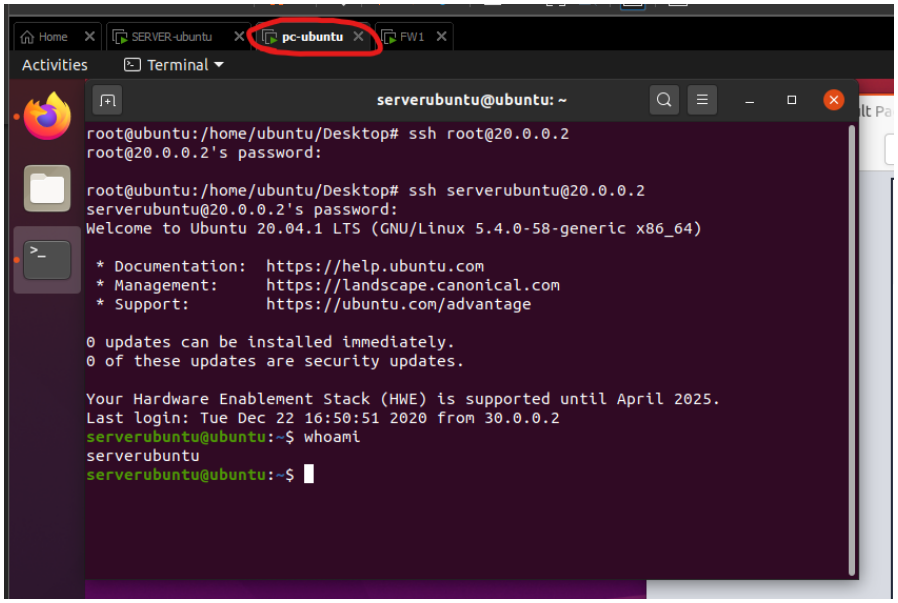

Fig. 6.1.3: prueba de acceso a la DMZ por puerto 22 desde Red Local

Ahora intentamos el mismo acceso desde Windows (simulando Internet), con PuTTY:

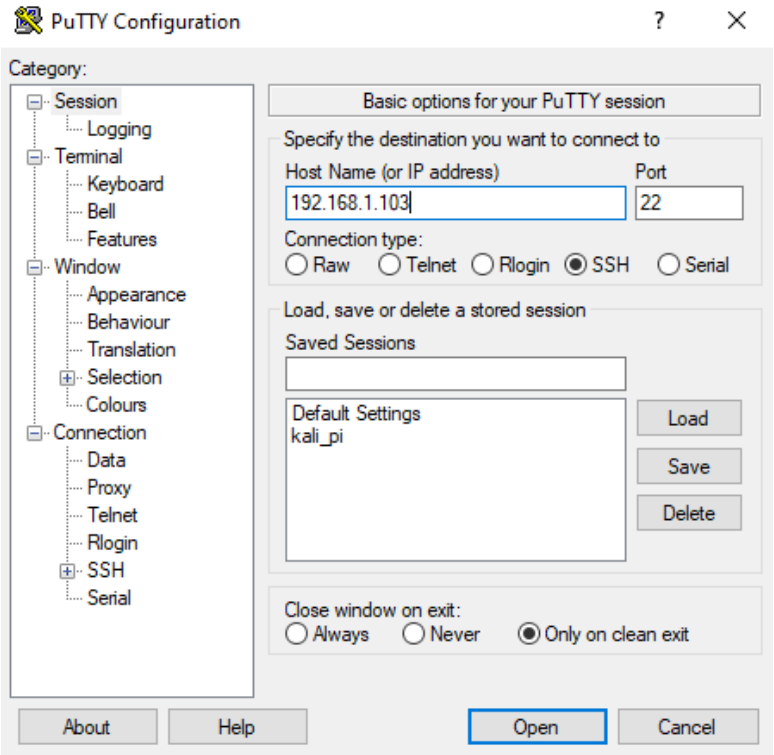

Fig. 6.1.4: configuración de PuTTY para probar el acceso desde Internet a la DMZ por puerto 22

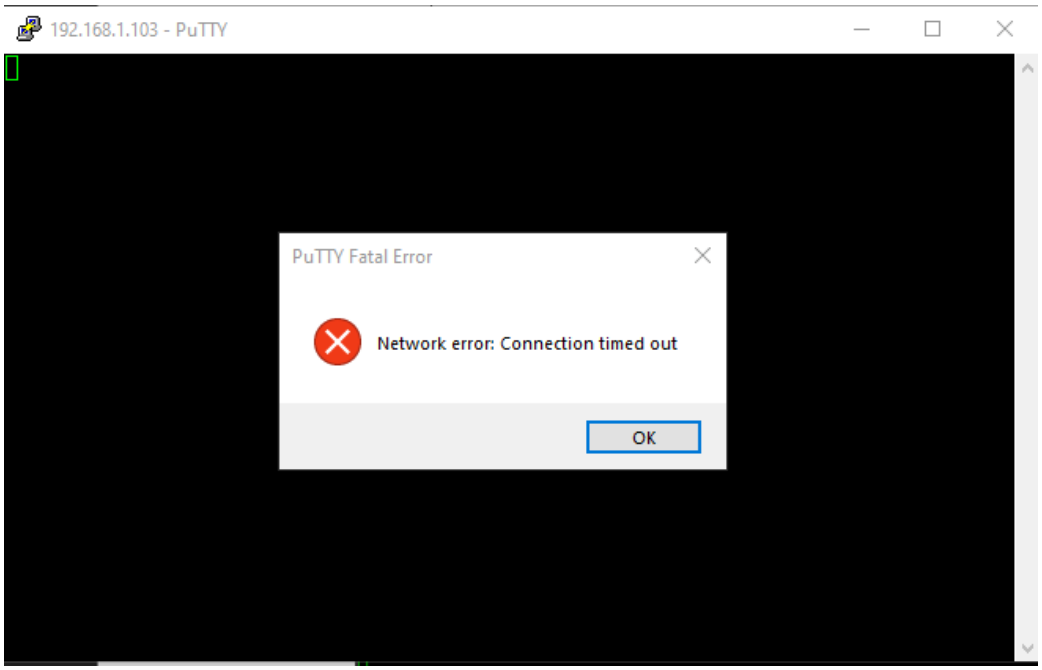

Fig. 6.1.5: prueba fallida de acceso a DMZ por puerto 22 desde Internet (simulación con equipo Windows)

Como se puede ver, obtenemos acceso desde la máquina Ubuntu de la Red Local (30.0.0.2/8), pero no desde Windows: el Firewall está funcionando de la forma deseada.

6.2. Acceso a Red Local desde Servidor e Internet

Otra de las condiciones del Firewall es que no puede haber accesos ilícitos a la Red Local, ni desde internet ni desde el Servidor.

Prueba de acceso a Red Local desde Servidor:

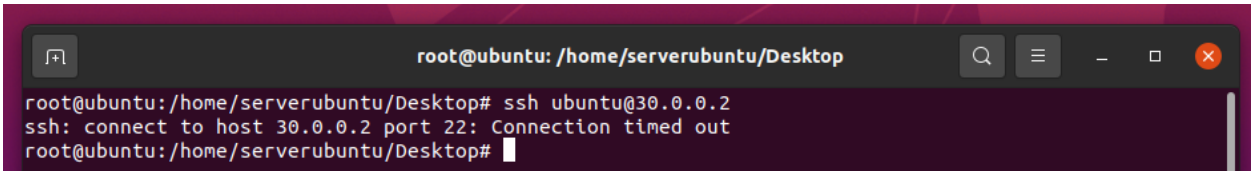

Fig. 6.2.1: prueba fallida de acceso a la Red Local desde la DMZ por puerto 22

Prueba de acceso a Red Local desde Internet (simulado desde Windows):

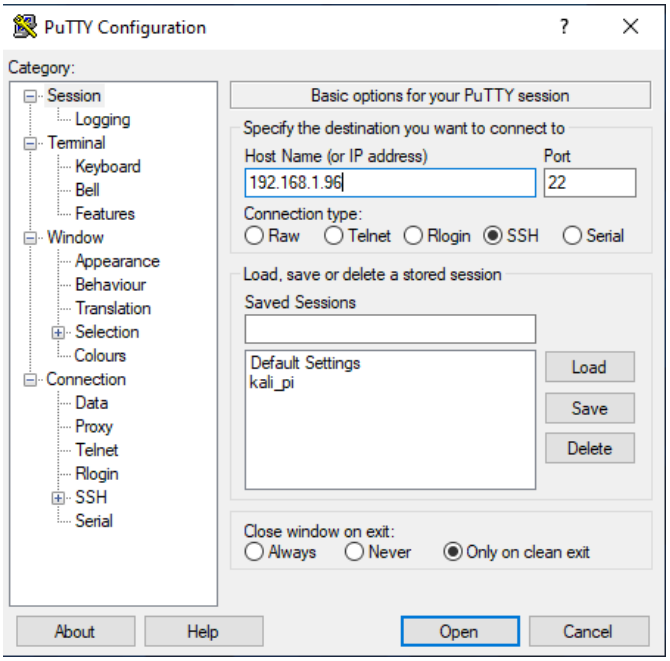

Fig. 6.2.2: configuración de PuTTY para probar el acceso desde Internet a la Red Local por puerto 22

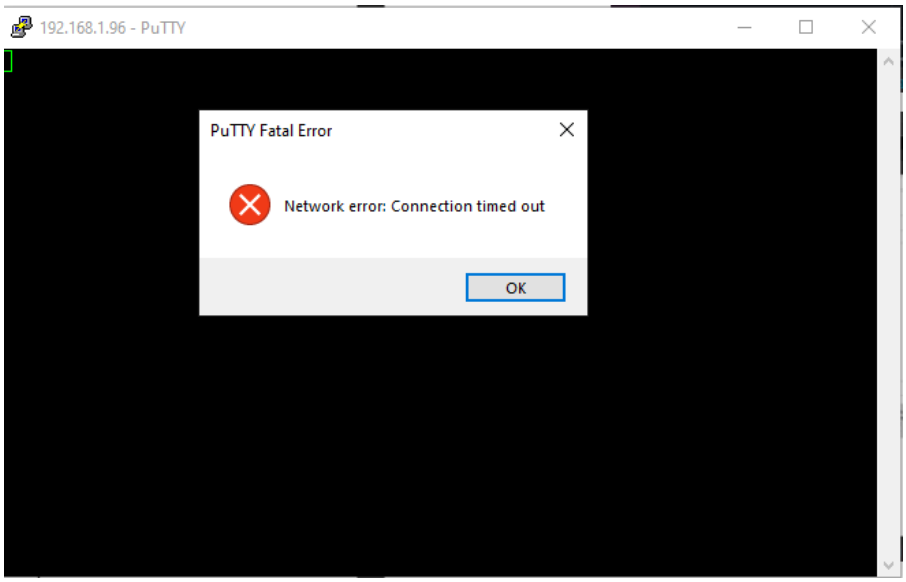

Fig. 6.2.3: prueba fallida de acceso a la Red Local desde Internet por puerto 22

Como se puede ver, en el hipotético caso de que un atacante pudiese obtener acceso SSH al servidor, no podría comunicarse con la Red Local (Fig. 6.2.1). También se puede observar que, desde Internet, no se

puede acceder por SSH a ningún equipo de la Red Local (Fig. 6.2.2 y 6.2.3), pues el Firewall no permite ninguna conexión entrante por dicho puerto.

6.3. Acceso a Internet desde Red Local

Otro de los requisitos del Firewall es permitir la navegación por Internet (DNS, http y https; puertos 53, 80 y 443, respectivamente) de dispositivos de la Red Local.

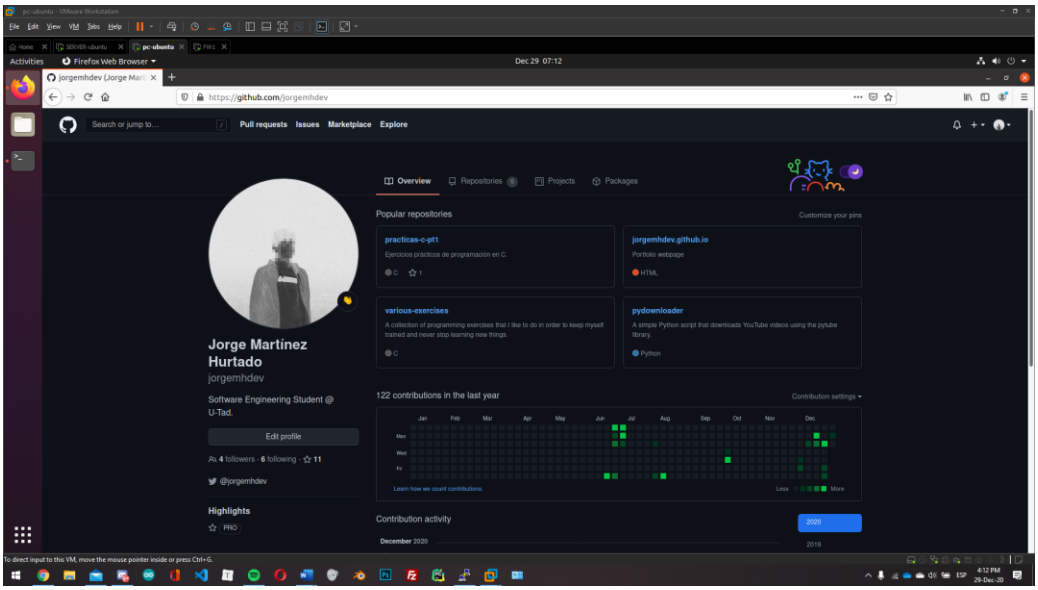

Fig. 6.3.1: prueba de acceso a Internet HTTPS (puerto 443) desde Red Local

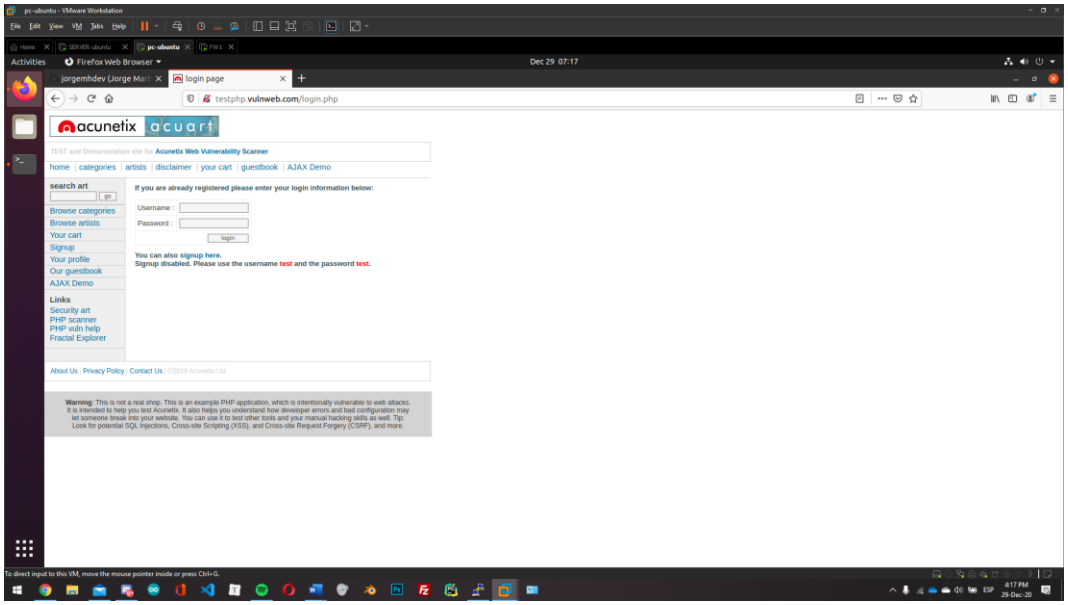

Fig. 6.3.2: prueba de acceso a Internet HTTP (puerto 80) desde Red Local

Tal y como se puede observar, tanto el tráfico HTTP como el tráfico HTTPS está permitido; los nombres de dominios han sido resueltos mediante DNS, lo que indica que el puerto 53 también ha sido configurado de forma correcta.

7. Conclusión de la Práctica

Mediante esta práctica, hemos podido comprobar la importancia de contar con un Firewall en una red a la hora de dotar a la misma de seguridad, así como la importancia a la hora de planificar todos los aspectos de la arquitectura de red desde el principio. Probablemente lo que más nos ha impresionado es el hecho de que un equipo no dedicado a tareas de red se pueda utilizar como Firewall, así como su forma de configurarlo. Gracias a esta práctica, hemos podido asentar conceptos vistos en clase anteriormente como la transmisión de información dentro de una red o los protocolos usados para la misma.

Obviamente, la práctica es una prueba de concepto de una arquitectura real; la maqueta funciona, pero en un escenario real no se usarían máquinas virtuales, sino hardware dedicado. Además, si tuviésemos que implementar esta arquitectura en un escenario real, probablemente añadiríamos un punto de acceso inalámbrico de forma que se pudiesen conectar dispositivos como *smartphones*, portátiles o *tablets.* Este punto de acceso inalámbrico supondría una tercera red con las mismas reglas que la Red Local.

Utilizaríamos un firewall Cisco Firepower cuya serie dependería de la envergadura de la empresa (siendo el Firepower 1000 para una pequeña empresa como la de la práctica y el Firepower 9000 para Data Center), así como un punto de acceso inalámbrico como el RUCKUS R320, que soporta redes 5G y cuyo rango de cobertura es amplio.