Sistemas Telemáticos para Medios Audiovisuales
Práctica 6: Cortafuegos *(firewalls)*

GSyC
Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación

Diciembre de 2018

1. Escenario para la configuración de un firewall

En el fichero `lab-firewall.tgz` se encuentran los ficheros de configuración del escenario de red que se muestra en la figura 1. En esta figura se representa un conjunto de subredes y máquinas (`pc1`, `pc2`, `pc4`, `pc5`, `r1`, `r2` y `firewall`) que pertenecen a una determinada empresa y su conexión a Internet a través de la máquina `firewall`. La empresa tiene definidas un conjunto de subredes de ámbito privado:

- 10.0.0.0/24: r1(eth1), pc1, pc2
- 10.0.1.0/24: firewall(eth0), r1(eth0), r2(eth0)
- 10.0.2.0/24: r2(eth1), pc3

Adicionalmente, la empresa tiene las máquinas pc4 y pc5 que se encuentran en una subred pública: 100.0.0.0/24. Estas máquinas proporcionan servicios básicos de la empresa: servidor de HTTP y servidor de fecha y hora. A este tipo configuración, donde la empresa tiene una o varias subredes públicas para ofrecer servicios a Internet se le denomina zona desmilitarizada o DMZ (DeMilitarized Zone).

Todas las máquinas de la empresa se conectan a Internet a través de la máquina `firewall` y la subred 100.0.1.0/24.

En este escenario, se considera que Internet está formado por las siguientes máquinas: `r3`, `r4`, `r5`, `pc6` y `pc7` que se encuentran conectadas a las siguientes subredes públicas:

- 100.0.1.0/24: r3(eth0)
- 100.0.2.0/24: r3(eth1), r5(eth2)
- 100.0.3.0/24: r3(eth2), r4(eth2)
- 100.0.4.0/24: r4(eth1), r5(eth0)
- 100.0.5.0/24: r4(eth0), pc6
- 100.0.6.0/24: r5(eth1), pc7

Arranca de una en una todas las máquinas de la figura.

En esta práctica se configurará la máquina `firewall` para que actúe como traductor de direcciones y como cortafuegos. Habrá que definir varias reglas utilizando `iptables`. Por este motivo, es recomendable guardar dichas reglas en un fichero *script de shell*.

Para hacer un *script de shell* crea un fichero de texto de nombre, por ejemplo, fw.sh, editándolo con `mcedit` en la forma:

```
mcedit fw.sh
```

La primera línea del fichero debe ser `#!/bin/sh` y las siguientes líneas serán la definición de las reglas para `iptables` tal y como se escribirían en el terminal:

```
#!/bin/sh

# Esto es un comentario

iptables -t filter -F
...
```

Una vez creado el *script* debes darle permisos de ejecución con la orden:

```
chmod 755 fw.sh
```

Y por último, para ejecutarlo, debes escribir:

```
./fw.sh
```

En los siguientes apartados realiza *scripts* en el `firewall` que implementen la siguiente configuración.

Para esta práctica se hará uso de la herramienta `nc` (ya estudiada en la asignatura Arquitectura de Internet) que permite arrancar aplicaciones TCP y UDP en modo cliente/servidor. Consulta el anexo de la sección 4 para ver cómo se utilizan.

2. Traducción de direcciones y puertos en el firewall: tabla nat

2.1. Cliente en la red privada, servidor externo

1. Configura un *script* `fw1.sh` en el *firewall* para que primero borre las reglas que hubiera configuradas previamente en la tabla `nat` y reinicie los contadores de dicha tabla, y a continuación realice la traducción de direcciones en el tráfico saliente de las redes privadas (SNAT) y en su correspondiente tráfico de respuesta. Explica para qué subredes has tenido has tenido que realizar la configuración de SNAT.

2.1.1. ICMP

Ejecuta el *script* `fw1.sh` de 2.1.

1. Ejecuta el siguiente comando en `pc1`:

   ```
   pc1:~# ping -c 2 100.0.5.60
   ```

   Y realiza una captura `iptables-01.cap` en `r3`. Explica las direcciones IP que se usan en la captura.

2. Explica qué significa el resultado de la ejecución del siguiente comando en `firewall`:

   ```
   firewall:~# iptables -t nat -L -v -n
   ```

   Qué regla/s está/n cumpliendo los paquetes ICMP *echo request* e ICMP *echo response* y cuántas veces se cumple/n.

2.1.2. UDP

Ejecuta el *script* `fw1.sh` de 2.1 para que reinicie los contadores de paquetes de iptables.

1. Ejecuta `nc` en modo servidor UDP en `pc6` y `nc` en modo cliente UDP en `pc2`. Simultáneamente realiza una captura en `r3` (`iptables-02.cap`) y consulta la información `ip_conntrack` de `firewall` con el comando:

   `firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack`.

   Escribe 5 líneas en el terminal de `pc2` para que se las envíe a `pc6` (cada línea, es decir, cada vez que pulsas una cadena de caracteres y `<Enter>` se envía un paquete UDP nuevo). Observa el número de paquetes enviados en la información que muestra `ip_conntrack`. Escribe una línea en `pc6` para que se la envíe a `pc2`. Observa nuevamente el número de paquetes en `ip_conntrack`.

   Interrumpe la captura y las ejecuciones de `nc`, explica la captura y cómo ésta se relaciona con la información que has visto en `ip_conntrack`.

2. Explica lo que muestra el comando en `firewall`:

   ```
   firewall:~# iptables -t nat -L -v -n
   ```

   Qué regla/s están cumpliendo los paquetes y cuántas veces se cumple/n.

3. Vuelve a repetir la misma prueba anterior pero iniciando el servidor UDP en `pc6` y el cliente UDP en `pc3`. Escribe 5 líneas en el terminal de `pc3` para que se las envíe a `pc6`. Observa la información de `ip_conntrack`. Escribe una línea en `pc6` para que se la envíe a `pc3`. Observa la información en `ip_conntrack`.

   Explica lo que muestra el comando en `firewall`:

   ```
   firewall:~# iptables -t nat -L -v -n
   ```

4. Primero inicia una captura en `r3` (`iptables-03.cap`) para capturar todo el tráfico que atraviese este router y otra captura en `r1-eth0` (`iptables-04.cap`).

   Ejecuta una aplicación servidor UDP escuchando en el puerto 7777 en `pc7` con el comando `nc`:

   ```
   pc7:~# nc -u -l -p 7777
   ```

   Ejecuta en `pc1` una aplicación cliente UDP que utilice localmente el puerto 6666 y que se comunique con ese servidor anterior de la siguiente manera:

   ```
   pc1:~# nc -u -p 6666 100.0.6.70 7777
   ```

   Y ejecuta en `pc2` una aplicación cliente UDP que utilice localmente el puerto 6666 y que se comunique con ese servidor anterior de la siguiente manera:

   ```
   pc2:~# nc -u -p 6666 100.0.6.70 7777
   ```

   Escribe una cadena de caracteres a través de la entrada estándar de `pc1` y pulsa `<Enter>`. A continuación introduce una cadena de caracteres a través de la entrada estándar de `pc2` y pulsa `<Enter>`. Interrumpe las dos capturas y explica qué ocurre con la traducción de direcciones y puertos en los paquetes que envía `pc1` a `pc7` y en los que envía `pc2` a `pc7`. [1]

[1] Con el envío desde pc2, el servidor en pc7 ya no está escuchando en el puerto 7777 porque el servidor UDP de nc sólo

2.1.3. TCP

Ejecuta el *script* `fw1.sh` de 2.1 para que reinicie los contadores de paquetes de iptables.

1. Para este apartado vamos a usar `nc` en modo TCP.

   Primero inicia una captura en `r3` (`iptables-05.cap`) para capturar todo el tráfico que atraviese este router.

   Ejecuta una aplicación servidor TCP escuchando en el puerto 7777 en `pc6` con el comando `nc`:

   ```
   pc6:~# nc -l -p 7777
   ```

   Y ejecuta en `pc1` una aplicación cliente TCP que se comunique con el servidor anterior de la siguiente manera:

   ```
   pc1:~# nc 100.0.5.60 7777
   ```

   No introduzcas nada por la entrada estándar, ni en `pc1` ni en `pc6`.

   Simultáneamente consulta `ip_conntrack` del `firewall` cada medio segundo. Explica el número de paquetes que se han observado en cada sentido, razonando la respuesta.

2. Introduce una palabra en la entrada estándar de `pc1`, pulsa `<Enter>` y explica razonadamente lo que observas en `ip_conntrack`.

3. Realiza un Ctrl+C en el terminal de `pc1` para interrumpir la ejecución de `nc`. Interrumpe la captura en `r3` y contrasta lo que observas en la captura con lo que muestra `ip_conntrack`.

2.2. Servidor en la red privada, cliente externo

Aunque en una red como la que aparece en la figura, lo habitual es colocar los servidores accesibles desde el exterior en la zona DMZ, para ver cómo funciona DNAT, vamos permitir que haya servidores accesibles desde el exterior en la red privada interna.

2.2.1. UDP

Realiza un nuevo *script* de iptables `fw2.sh` en `firewall` que primero borre las reglas que hubiera configuradas previamente en la tabla `nat`, reinicie los contadores de dicha tabla, y a continuación, realice la siguiente traducción de direcciones:

- El tráfico de entrada al firewall destinado al puerto UDP 5001 debe ser redirigido a `pc1`, puerto 5001.
- El tráfico de entrada al firewall destinado al puerto UDP 5002 debe ser redirigido a `pc2`, puerto 5001.

1. Explica el nuevo *script*.
2. Lanza `nc` en modo servidor UDP en `pc1` y `pc2`, escuchando en ambos casos en el puerto 5001.Lanza `nc` en modo cliente UDP en `pc6` y `pc7` de tal forma que el tráfico generado en `pc6` lo reciba `pc1` y el tráfico generado en `pc2` lo reciba `pc7`. Explica cómo has arrancado los dos clientes `nc` en `pc6` y `pc7`.

puede atender a un cliente simultáneamente. Por tanto, con el envío desde pc2, pc7 enviará un error ICMP. Para este apartado este hecho no es importante, sólo queremos analizar que ocurre con la traducción de direcciones IP y puertos que ocurre en firewall.

3. Explica el resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.

4. Explica el resultado de ejecutar en `firewall` indicando el número de reglas que se han cumplido:

```
firewall:~# iptables -t nat -L -v -n
```

2.2.2. TCP

Añade la siguiente configuración de traducción de direcciones al *script* `fw2.sh` de iptables de `firewall`:

- El tráfico de entrada al firewall destinado al puerto TCP 80 debe ser redirigido a `pc3`, puerto 80.

1. Explica las modificaciones del *script*.

2. Lanza `nc` en modo servidor TCP en `pc3` escuchando en el puerto 80. Lanza `nc` en modo cliente TCP en `pc6` de tal forma que el tráfico generado en `pc6` lo reciba `pc3` Explica cómo has arrancado el cliente de `nc` en `pc6`.

3. Explica el resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.

4. Explica el resultado de ejecutar en `firewall` indicando el número de reglas que se han cumplido:

```
firewall:~# iptables -t nat -L -v -n
```

3. Filtrado en el firewall: tabla filter

1. Crea un *script* en el `firewall` `fw3.sh` partiendo de la configuración de traducción de direcciones IP y puertos realizada en `fw1.sh` que añada la siguiente configuración:

   a) Reiniciar la tabla `filter`: borrar su contenido y reiniciar sus contadores.

   b) Fijar las políticas por defecto de las cadenas de la tabla `filter`, haciendo que por defecto se descarte todo el tráfico en el `firewall` excepto los paquetes de salida.

   c) Permitir el tráfico de entrada dirigido a las aplicaciones que se están ejecutando en `firewall` únicamente si este tráfico tiene su origen en las subredes privadas de la empresa.

   d) Permitir todo el tráfico saliente desde las subredes privadas hacia Internet y el tráfico de respuesta al saliente. Ten en cuenta que como has partido del *script* `fw1.sh`, en dicho *script* ya tenías las reglas de la tabla `nat` de modificación de la dirección IP de origen de los paquetes que reenvía el `firewall` y los paquetes del tráfico entrante de respuesta al saliente.

   e) Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas:

   - un servidor *echo* instalado en `pc4` (UDP, puerto 7). El servidor de *echo* es un servidor que al enviarle una cadena de caracteres, devuelve la misma cadena que se le ha enviado. Para comprobar el acceso a este servidor puedes utilizar el programa `nc`:

     ```
     nc -u 100.0.0.40 7
     ```

   - un servidor *daytime* instalado en `pc5` (UDP, puerto 13). El servidor *daytime* es un servidor que al enviarle algo, devuelve la fecha y hora de la máquina donde está instalado. Para comprobar el acceso a este servidor puedes utilizar el programa `nc`:

```
nc -u 100.0.0.50 13
```

f) Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma:

- Conexión de `telnet` (TCP, puerto 23) desde `pc1` a `pc5`. La conexión de `telnet` permite a un usuario conectarse de forma remota a otra máquina. Para poder probar esta comunicación, desde `pc1` ejecuta:
  ```
  telnet 100.0.0.50
  ```
  podrás entrar de forma remota en `pc5` utilizando usuario: `root`, clave: `root`.
- Conexión al servidor de *echo* (TCP, puerto 7) de `pc4` en `pc1`.

g) Desde la zona DMZ no se puede iniciar ninguna comunicación con la red privada, ni con el `firewall`.

3.1. Pruebas de la configuración del firewall

A continuación se dan algunas pautas para probar cada una de las restricciones de `fw3.sh`:

1. Permitir el tráfico de entrada en la máquina `firewall` únicamente desde las subredes privadas de la empresa.

   Pruebas

   a) Si se arranca una aplicación servidor (TCP o UDP) en la máquina `firewall` sólo podrá aceptar tráfico de un cliente que envíe mensajes desde una de las máquinas de las subredes privadas. Indica qué reglas se ejecutan cuando por ejemplo se arranca un servidor UDP en `firewall` y se arranca un cliente UDP en `pc1` que se comunique con dicho servidor. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

   b) No podrá aceptar tráfico desde aplicaciones cliente lanzadas en otras subredes diferentes. Indica qué reglas se ejecutan cuando por ejemplo se arranca un servidor UDP en `firewall` y arrancas un cliente UDP en `pc6` que se comunique con dicho servidor. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

2. Permitir todo el tráfico saliente desde las subredes privadas hacia Internet, modificando la dirección IP de origen de los paquetes que reenvía el `firewall`, y el tráfico entrante de respuesta al saliente.

   Pruebas

   a) Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de Internet y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de las subredes internas, el tráfico debe poder enviarse del cliente al servidor y del servidor al cliente, observando que el tráfico que sale del firewall con destino a la máquina de Internet no tiene como dirección IP origen la dirección de la máquina que pertenece a la subred privada, sino que lleva la dirección 100.0.1.100. Ejecuta la misma prueba que en el apartado 2.1.3 e indica que reglas de fw3.sh se han aplicado. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

b) Si se arranca una aplicación cliente en `pc4` o `pc5` para comunicarse con el servidor que se haya arrancado en una de las máquinas de Internet, el `firewall` no debería permitir reenviar ese tráfico hacia Internet. Indica que reglas de fw3.sh se han aplicado en este caso para rechazar los paquetes.

3. Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas:

   - un servidor *echo* instalado en `pc4` (UDP, puerto 7).

     Pruebas

     a) Desde una máquina de Internet se debería poder acceder a ese servidor de *echo* de `pc4`:

     ```
     nc -u 100.0.0.40 7
     ```

     Indica qué reglas de `fw3.sh` se han aplicado en este caso. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

     b) Si se prueba lo mismo arrancando el comando anterior desde `pc3` no debería comunicarse. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes.Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

   - un servidor *daytime* instalado en `pc5` (UDP, puerto 13). El servidor *daytime* es un servidor que al enviarle algo, devuelve la fecha y hora de la máquina donde está instalado.

     Pruebas

     a) Desde una máquina de Internet se debería poder obtener la hora de `pc5`:

     ```
     nc -u 100.0.0.50 13
     ```

     Pulsar enter en el terminal de `nc` y debería obtenerse la hora que le envía `pc5`. Indica qué reglas de `fw3.sh` se han aplicado en este caso. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

     b) No se debe permitir otro tipo de tráfico desde Internet a DMZ. Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de DMZ y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de Internet, el tráfico no debería poder enviarse del cliente al servidor. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

4. Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma:

   a) Conexión de `telnet` (TCP, puerto 23) desde `pc1` a `pc5`. La conexión de `telnet` permite a un usuario conectarse de forma remota a otra máquina.

   Pruebas

   1) Desde `pc1` ejecuta el cliente de `telnet`:

   ```
   telnet 100.0.0.50
   ```

podrás entrar de forma remota en `pc5` utilizando usuario: `root`, clave: `root`. Indica qué reglas de `fw3.sh` se han aplicado en este caso. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

2) Si se prueba lo mismo arrancando el cliente de `telnet` desde `pc2` o `pc3` o cualquier máquina de Internet no debería permitir la conexión. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

b) Conexión al servidor de *echo* (TCP, puerto 7) de `pc4` en `pc1`.
Si se arranca cualquier otra aplicación servidor (TCP o UDP) en una de las máquinas de la DMZ y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de las subredes privadas, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

Pruebas

1) Desde `pc1` se debería poder conectarse al servidor de *echo* de `pc4`:
```
nc 100.0.0.40 7
```
Indica qué reglas de `fw3.sh` se han aplicado en este caso.

2) Si se prueba lo mismo arrancando `nc` desde `pc2` o `pc3` no debería conectarse. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

5. Desde la zona DMZ no se puede iniciar ninguna comunicación con la red privada, ni con el `firewall`.

Pruebas

a) Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de las subredes privadas y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes.

b) Si se arranca una aplicación servidor (TCP o UDP) en el `firewall` y se arranca una aplicación cliente para que se comunique con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor. Indica qué reglas de `fw3.sh` se han aplicado en este caso para rechazar los paquetes.

4. Anexo - Generar tráfico con nc

En esta práctica utilizaremos la aplicación `nc` para intercambiar tráfico a través de aplicaciones cliente/servidor en TCP y UDP.

Al arrancar la aplicación que funciona como servidor, ésta se quedará esperando a recibir mensajes de otras aplicaciones que funcionan como clientes.

Al arrancar la aplicación que funciona como cliente, ésta tomará la iniciativa de enviar el primer mensaje a la aplicación servidor que ya tiene que estar preparada para recibir mensajes de los clientes. Por este motivo, **es necesario arrancar primero la aplicación que funciona como servidor** y posteriormente arrancar la aplicación que funciona como cliente.

`nc` puede ser lanzado como servidor o como cliente TCP o UDP en cualquier máquina. Una aplicación `nc` lanzada como cliente se comunicará con otra lanzada como servidor y viceversa. Una vez arrancada, `nc` permite al usuario escribir líneas de texto a través de la entrada estándar. Cada vez que se pulsa `Enter`, la línea de texto es enviada por la red a la máquina remota, la cuál mostrará la línea recibida.

4.1. Tráfico UDP

4.1.1. Aplicación servidor UDP

Para arrancar una aplicación que funciona como servidor utilizando el protocolo UDP ejecutaremos la siguiente instrucción:

```
nc -u -l -p <Pto-Loc>
```

Donde `<Pto-Loc>` es el número de puerto local UDP en el que la aplicación servidor está esperando recibir datagramas UDP de los clientes.

Por ejemplo, si queremos arrancar una aplicación servidor UDP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente instrucción:

```
pc1:~# nc -u -l -p 7777
```

4.1.2. Aplicación cliente UDP

Para arrancar una aplicación que funciona como cliente utilizando el protocolo UDP ejecutaremos la siguiente instrucción:

```
nc -u -p <Pto-Loc> <IP-dest> <Pto-dest>
```

Donde:

- `<Pto-Loc>` es el número de puerto local UDP en el que la aplicación cliente está esperando recibir los datagramas UDP que vengan del servidor.
- `<IP-dest>` es la dirección IP de la máquina donde se está ejecutando la aplicación servidor de UDP.
- `<Pto-dest>` es el número de puerto UDP en el que la aplicación servidor está esperando recibir datagramas UDP de los clientes.

Por ejemplo, si queremos arrancar una aplicación cliente UDP que espere recibir datagramas UDP en el puerto 6666 y que envíe datagramas UDP a la dirección IP 200.0.0.1 y puerto 7777 (donde se encuentra esperando recibir datagramas UDP la aplicación servidor) utilizaremos la siguiente instrucción:

```
pc2:~# nc -u -p 6666 200.0.0.1 7777
```

4.1.3. Envío de datos UDP

Como en UDP no hay establecimiento de conexión, **para que se genere tráfico entre el cliente y el servidor UDP es necesario escribir algo (y darle a enter) para que se envíen mensajes UDP**.

Es necesario que primero se escriba en el terminal del cliente para que se lo envíe al servidor. Después de que el cliente haya enviado una primera línea de texto al servidor, todo lo que escribamos a través de la entrada estándar de un extremo será enviado al otro extremo como datagramas UDP: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.

Para interrumpir la ejecución de estas aplicaciones utilizaremos `Ctrl+C`.

4.2. Tráfico TCP

4.2.1. Aplicación servidor TCP

Para arrancar una aplicación que funciona como servidor utilizando el protocolo TCP ejecutaremos la siguiente instrucción:

```
nc -l -p <Pto-Loc>
```

Donde `<Pto-Loc>` es el número de puerto local TCP en el que la aplicación servidor está esperando recibir peticiones de inicio de conexión TCP de los clientes.

Por ejemplo, si queremos arrancar una aplicación servidor TCP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente instrucción:

```
pc1:~# nc -l -p 7777
```

4.2.2. Aplicación cliente TCP

Para arrancar una aplicación que funciona como cliente utilizando el protocolo TCP ejecutaremos la siguiente instrucción:

```
nc -p <Pto-Loc> <IP-dest> <Pto-dest>
```

Donde:

- `<Pto-Loc>` es el número de puerto local TCP desde el que la aplicación cliente establecerá la conexión TCP con el servidor.
- `<IP-dest>` es la dirección IP de la máquina donde se está ejecutando la aplicación servidor TCP.
- `<Pto-dest>` es el número de puerto TCP en el que la aplicación servidor está esperando recibir peticiones de conexiones TCP de los clientes.

Por ejemplo, si queremos arrancar una aplicación cliente TCP que utilice el puerto origen 6666 para establecer una conexión TCP con un servidor TCP que escuche en el puerto destino 7777 de la máquina 200.0.0.1, utilizaremos la siguiente instrucción:

```
pc2:~# nc -p 6666 200.0.0.1 7777
```

Una vez establecida la conexión entre ambos, el cliente y el servidor podrán intercambiar segmentos TCP en ambos sentidos.

4.2.3. Envío de datos TCP

Una vez iniciadas las aplicaciones servidor TCP y cliente TCP, todo lo que escribamos a través de la entrada estándar de un extremo será enviado al otro extremo como segmentos TCP: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.

Para interrumpir la ejecución de estas aplicaciones utilizaremos `Ctrl+C`.

5. Entrega de la práctica

Es necesario entregar la siguiente documentación:

- Memoria en formato pdf donde se explique razonadamente el diseño y la configuración de cada uno de los apartados de este enunciado, así como las pruebas realizadas para comprobar cada característica del cortafuegos pedida.
- Scripts de configuración de `iptables`:
  - `fw1.sh`
  - `fw2.sh`
  - `fw3.sh`

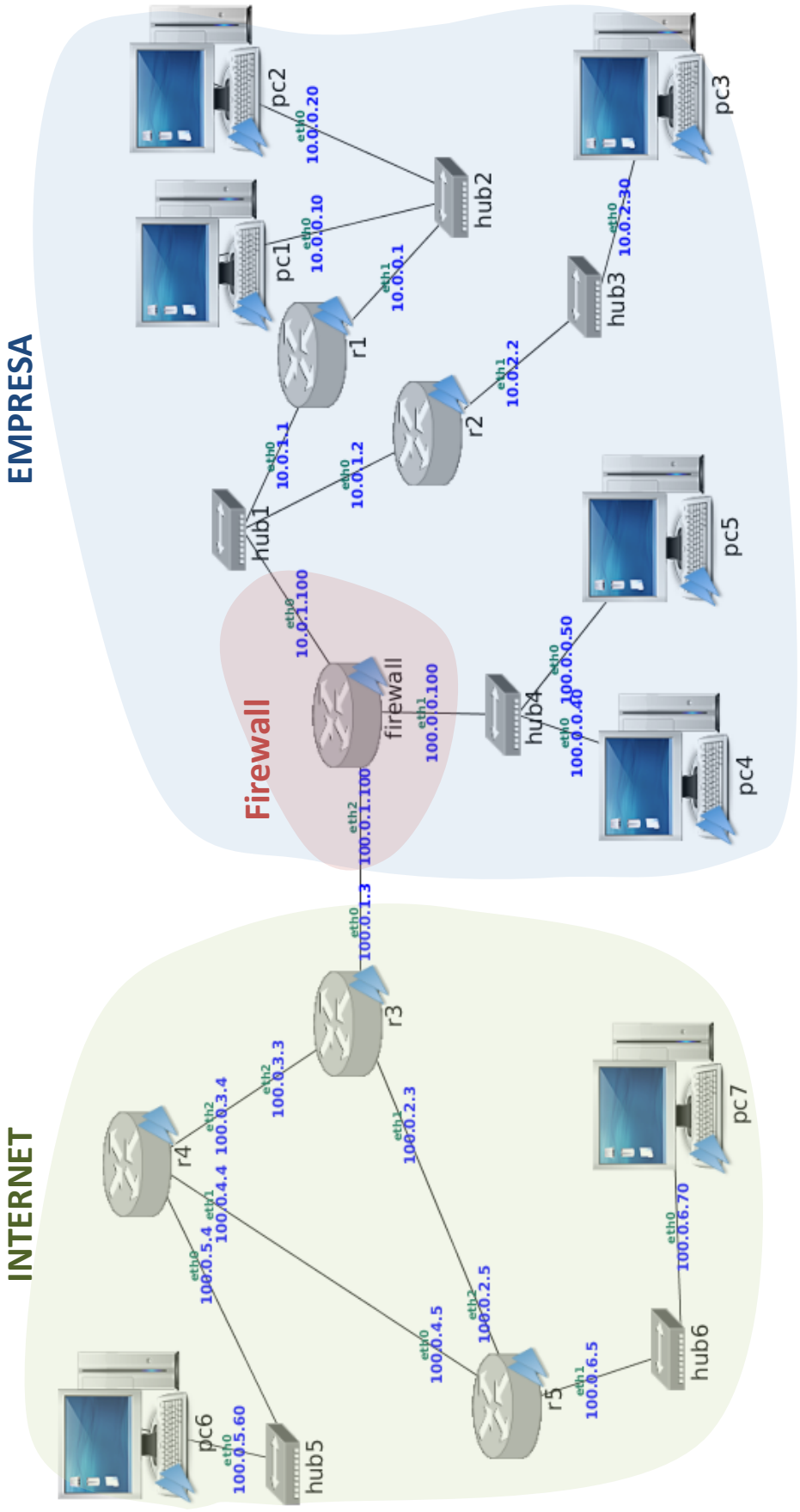

Figura 1: Escenario de red para los ejercicios de configuración de firewall