Planificación y Gestión de Redes de Ordenadores
Práctica 2: Red frontera - *(firewalls)*

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

Septiembre de 2014

1. Escenario para la configuración de un firewall

En la figura 1 se muestra una red empresarial (`pc1`, `pc2`, `pc3`,`pc4`, `pc5`, `r1`, `r2` y `firewall`) formada por una zona con direccionamiento privado y una zona DMZ con direccionamiento público. La empresa se conecta a Internet a través de un *router* `firewall` que será necesario configurarlo atendiendo a los requisitos impuestos por la empresa.

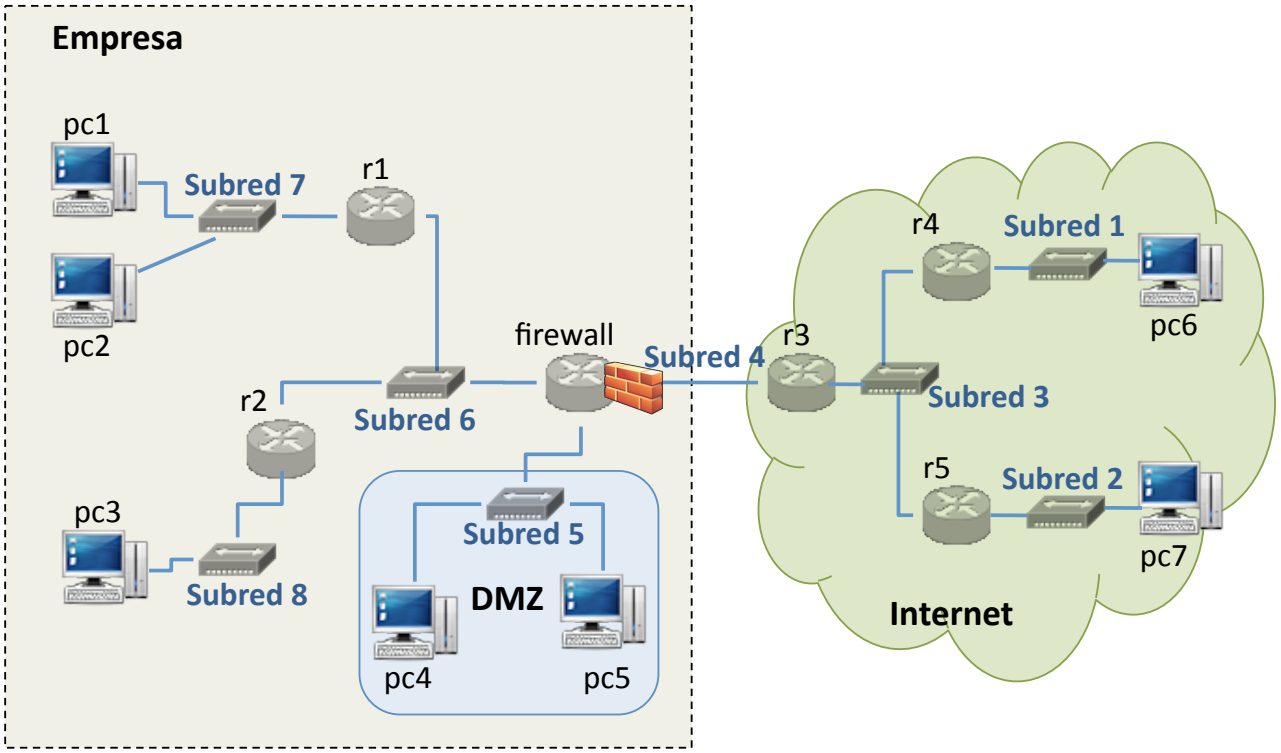

Figura 1: Escenario de red para los ejercicios de configuración de firewall

Deberás primeramente asignar direcciones IP a las máquinas que aparecen en la figura, utilizando las subredes que tenías asignadas en la práctica 1: subred 1, subred 2, subred 3, subred 4 y subred 5. Para las subredes de ámbito privado, calcula las direcciones IP teniendo en cuenta la siguiente fórmula:

- Subred 6: `10.6.X.0/24`
- Subred 7: `10.7.X.0/24`

- Subred 8: `10.8.X.0/24`

Donde `X` es el 2º byte más significativo de las subredes que tienes asignadas en la práctica 1. Por ejemplo, si en la práctica 1 tenías asignadas las direcciones 15.22.0.0/24, X=22 y las subredes que debes utilizar para el direccionamiento privado son:

- Subred 6: `10.6.22.0/24`
- Subred 7: `10.7.22.0/24`
- Subred 8: `10.8.22.0/24`

La empresa tiene las máquinas pc4 y pc5 que se encuentran en una subred pública: subred5. Estas máquinas proporcionan servicios básicos de la empresa, como por ejemplo un servidor de fecha y hora. A este tipo de 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`. El `firewall` deberá aplicar reglas de traducción de direcciones para tráfico de las subredes privadas, no siendo necesaria la traducción de direcciones para el tráfico de la zona DMZ.

En este escenario, se considera que Internet está formado por las siguientes máquinas: `r3`, `r4`, `r5`, `pc6` y `pc7`.

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

Configura las direcciones IP en cada una de las máquinas, asignándoles una dirección IP válida en la subred a la que pertenecen. Configura las rutas que sean necesarias en cada uno de los *routers* para que todas las máquinas de las subredes privadas se puedan comunicar entre ellas y todas las máquinas de las subredes públicas se puedan comunicar entre ellas. Hasta que no se configuren las reglas NAT en el `firewall` no se podrán comunicar las máquinas de las subredes privadas con las de Internet. **El *router* `r3` sólo puede tener rutas a las subredes públicas: subred 1, subred 2, subred 3, subred 4 y subred 5. No puedes configurarle una ruta por defecto**.

Incluye en la memoria una imagen del escenario de NetGUI donde se muestren las direcciones IP que has configurado y los nombres de las interfaces (eth0, eth1, etc).

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 esta práctica se hará uso de la herramienta `nc` 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

Comprueba que no funciona un `ping` desde las máquinas internas de las redes privadas (`pc1`, `pc2` y `pc3`) a destinos de Internet como `pc6` o `pc7`.

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 que realizar la configuración de SNAT. Incluye el *script* `fw1.sh` en la memoria y explícalo.

2.1.1. ICMP

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

1. Realiza una captura de tráfico en `r3` (`iptables-01.cap`). Ejecuta un `ping` desde `pc1` a `pc6` con la opción que permite enviar sólo 2 paquetes ICMP *echo request* (`-c 2`).

   Interrumpe la captura de tráfico. 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. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.

3. Consulta la información de seguimiento de conexiones del módulo `ip_conntrack` del `firewall` y explica el resultado.

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` (con cada línea, es decir cada vez que pulsas una cadena de caracteres y `<Enter>`, se envía un paquete UDP nuevo). Observa el estado de `ip_conntrack`. Escribe una línea en `pc6` para que se la envíe a `pc2`. Observa el estado de `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 contenido de la tabla `nat` del `firewall`. Indica qué regla/s están cumpliendo los paquetes y cuántas veces se cumple/n. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.

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 el estado de `ip_conntrack`. Escribe una línea en `pc6` para que se la envíe a `pc3`. Observa el estado de `ip_conntrack`.

   Interrumpe las ejecuciones de `nc`, explica lo que muestra el contenido de la tabla `nat` del `firewall`. Indica qué regla/s están cumpliendo los paquetes y cuántas veces se cumple/n. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.

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

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

   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 <dirIP_de_pc7> 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 <dirIP_de_pc7> 7777
```

Consulta la información de `ip_conntrack` en `firewall`, dado que todavía no se han enviado datos, no debería aparecer nada.

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. [1]

5. Consulta la tabla `nat` del `firewall` y explica cuántas veces se han cumplido las reglas de traducción de direcciones.

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`.

   Y ejecuta en `pc1` una aplicación cliente TCP que se comunique con el servidor anterior.

   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`.

4. Consulta la tabla `nat` del `firewall` y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.

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.

[1] Con el envío desde `pc1`, el servidor en `pc7` atiende a `pc1` y ya no puede atender a más clientes ya que `nc` sólo permite atender a un cliente simultáneamente. Por tanto, cuando envía posteriormente `pc2`, `pc7` ya no está escuchando en el puerto 7777, ya que `pc7` está atendiendo a `pc1`. Por este motivo, 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` con cada una de las comunicaciones que lo atraviesan.

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` y 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. Inicia una captura de tráfico en `r3` (`iptables-06.cap`). 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`.
3. Escribe una línea en cada uno de los terminales involucrados (`pc1`, `pc2`, `pc6` y `pc7`). Interrumpe los clientes y servidor con Ctrl+C. Interrumpe la captura de tráfico. Explica el resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
4. Consulta la tabla `nat` del `firewall` y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.
5. Relaciona los resultados de la captura de tráfico con la información extraída de `ip_conntrack` y la tabla `nat` del `firewall`.

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. Inicia una captura de tráfico en `r3` (`iptables-07.cap`). 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. Interrumpe el cliente y el servidor con Ctrl+C. Interrumpe la captura de tráfico. Explica el resultado observado en `ip_conntrack` y la traducción de direcciones IP y puertos realizada.
4. Consulta la tabla `nat` del `firewall` y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla `nat` y cuantos paquetes las han cumplido.
5. Relaciona los resultados de la captura de tráfico con la información extraída de `ip_conntrack` y la tabla `nat` del `firewall`.

3. Filtrado en el firewall: tabla filter

3.1. Introducción: Servidores echo, daytime, telnet

En linux se pueden manejar un conjunto de servicios a través del demonio `inetd`: *Internet "super-server"*. Este demonio proporciona el acceso a esos servicios, en particular, en la práctica utilizaremos lo servicios: *echo, daytime* y *telnet*.

Para activar los servicios que queremos utilizar, editaremos el fichero `/etc/inetd.conf`. En este fichero encontraremos comentados algunos de los servicios que queremos usar en la práctica. Para usarlos, habrá que descomentar dichas líneas. Si no encontramos esas líneas comentadas, escribiremos la configuración en ese fichero tal y como se indica a continuación:

- **Daytime**: Es un servidor que escucha conexiones TCP en el puerto `daytime` (en `/etc/services` podemos ver que el puerto `daytime` está asociado al puerto 13). Cuando un proceso se conecta a este número de puerto, el servidor *daytime* devuelve la hora actual de la máquina. Para activar el servidor *daytime* es necesario que en la máquina donde queremos ese servidor, en su fichero `/etc/inetd.conf` se encuentre la siguiente línea:

  ```
  daytime stream tcp nowait root internal
  ```

- **Echo**: Es un servidor que espera paquetes UDP en el puerto `echo` (en `/etc/services` podemos ver que el puerto `echo` está asociado al puerto 7). Cuando un proceso envía una cadena de caracteres al servidor *echo*, este servidor devuelve al cliente esa misma cadena. Para activar el servidor *echo* es necesario que en la máquina donde queremos ese servidor, en su fichero `/etc/inetd.conf` se encuentre la siguiente línea:

  ```
  echo dgram udp nowait root internal
  ```

  También se puede lanzar este mismo servicio a través del protocolo TCP, para ello, la línea del fichero `/etc/inetd.conf` debe ser:

  ```
  echo stream tcp nowait root internal
  ```

- **Telnet**: Es un servidor que escucha conexiones TCP en el puerto `telnet` (en `/etc/services` podemos ver que el puerto `telnet` está asociado al puerto 23). Cuando un proceso se conecta a este número de puerto, se establece una conexión remota entre la máquina cliente y la máquina servidor en la que el cliente puede ejecutar comandos en el servidor. Requiere una fase de autenticación. Para activar el servidor *telnet* es necesario que en la máquina donde queremos ese servidor, en su fichero `/etc/inetd.conf` se encuentre la siguiente línea:

  ```
  telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
  ```

  Para poder permitir el acceso remoto como usuario `root` en el servidor de `telnet` es necesario comentar, en la máquina donde se va a lanzar el servidor, la siguiente línea en el fichero: `/etc/pam.d/login`

  ```
  # auth [success=ok ignore=ignore user_unknown=ignore default=die] pam_securetty.so
  ```

  La línea es un comentario porque comienza con el carácter ''#''.

Una vez configurado el servicio que queremos arrancar dentro del fichero `/etc/inetd.conf` es necesario rearrancar el demonio `inetd` en la máquina donde queremos configurar los servicios para que se cargue la configuración de ese fichero. Para ello deberás ejecutar:

```
/etc/init.d/inetd restart
```

Puedes comprobar qué servicios están activos ejecutando:

```
netstat -atun
```

3.2. Configuración de las reglas de filtrado en el firewall

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 y su correspondiente tráfico de salida:

   - un servidor *echo* instalado en `pc4` (UDP, puerto 7). Debes configurar `inetd` en `pc4` para que arranque este servidor. Utiliza `nc` para probar la comunicación como cliente desde una máquina de Internet y el tráfico de respuesta.
   - un servidor *daytime* instalado en `pc5` (UDP, puerto 13). Debes configurar `inetd` en `pc5` para que arranque este servidor. Utiliza `nc` para probar la comunicación como cliente desde una máquina de Internet y el tráfico de respuesta.

   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`. Debes configurar `inetd` en `pc5` para que arranque este servidor.
     Para poder probar esta comunicación, desde `pc1` ejecuta:
     ```
     telnet <dir_IP_pc5>
     ```
     Podrás entrar de forma remota en `pc5` utilizando usuario: `root`, clave: `root`.
   - Conexión al servidor de *echo* (TCP, puerto 7) desde `pc1` a `pc4`. Debes configurar `inetd` en `pc4` para que arranque este servidor. Utiliza `nc` para probar la comunicación como cliente desde `pc1`.

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

   Incluye el *script* `fw3.sh` en la memoria y explícalo.

3.3. Pruebas de la configuración del firewall

Para poder comprobar qué reglas se están aplicando a cada caso que pruebas, añade a cada regla otra regla con las misma condiciones y acción `LOG` de forma que quede una anotación en el fichero de *log* cada vez que se cumpla cada condición.

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. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables. Por ejemplo arranca un servidor UDP en `firewall` y arranca un cliente UDP en `pc1` que se comunique con dicho servidor (escribe alguna línea en cada uno de los terminales para que haya tráfico UDP).

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

b) No podrá aceptar tráfico desde aplicaciones cliente lanzadas en otras subredes diferentes. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables. Por ejemplo, arranca un servidor UDP en `firewall` y arranca un cliente UDP en `pc6` que se comunique con dicho servidor (escribe alguna línea en cada uno de los terminales para que haya tráfico UDP).

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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 pública del `firewall` de la interfaz que le conecta con Internet. Ejecuta la misma prueba que en el apartado 2.1.3. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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. Asegúrate de que antes de lanzar cliente y servidor has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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`. Ejecuta el siguiente comando desde una máquina de Internet:

```
nc -u <dir_IP_pc4> 7
```

Asegúrate de que antes de lanzar el cliente desde una máquina de Internet has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

b) Si se prueba lo mismo arrancando el comando anterior desde `pc3` y se manda una cadena de caracteres, no se debería obtener respuesta.

Asegúrate de que antes de lanzar el cliente de `pc3` has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

- 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`. Ejecuta el siguiente comando desde una máquina de Internet:

```
nc -u <dir_IP_pc5> 13
```

Pulsa < *Enter* > en el terminal de `nc` y debería obtenerse la hora que le envía `pc5`.

Asegúrate de que antes de lanzar el cliente en `pc5` has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan..

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 ni del servidor al cliente.

Haz una prueba para este tipo de tráfico y explica que prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado `fw3.sh` para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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) Asegúrate de que antes de lanzar el cliente en `pc1` has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`. Desde `pc1` ejecuta el cliente de `telnet`:

```
telnet <dir_IP_pc5>
```

podrás entrar de forma remota en `pc5` utilizando usuario: `root`, clave: `root`.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces.

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.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

b) Conexión al servidor de *echo* (TCP, puerto 7) desde `pc1` a `pc4`.

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) Asegúrate de que antes de lanzar el cliente en `pc1` has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`. Desde `pc1` se debería poder conectarse al servidor de *echo* de `pc4`:

```
nc <dir_IP_pc4> 7
```

Explica en la memoria:
- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

2) Si se prueba lo mismo arrancando `nc` desde `pc2` o `pc3` no debería conectarse.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`.

Explica en la memoria:
- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`.

Explica en la memoria:
- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.

- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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 ni del servidor al cliente.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado `fw3.sh` para que reinicie los contadores de `iptables`.

Explica en la memoria:

- las reglas en las tablas `nat` y `filter` que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas `nat` y `filter` y el número de veces que se ejecutan.

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. Normas de entrega

Es necesario entregar la siguiente documentación:

- Memoria en formato pdf donde se explique razonadamente la configuración de cada uno de los apartados de este enunciado.
- Capturas de tráfico dentro de un único fichero `capturas-iptables.tgz`:
  - `iptables-01.cap`, `iptables-02.cap`, `iptables-03.cap`, `iptables-04.cap`, `iptables-05.cap`, `iptables-06.cap`, `iptables-07.cap`.
- El fichero `netgui.nkp` del escenario realizados en NetGUI.

La fecha límite de entrega de las prácticas será el día del examen publicado en el calendario oficial de exámenes de la ETSIT. No obstante, es recomendable que vayas realizando las prácticas gradualmente y las vayas subiendo al sitio moodle de la asignatura.

La entrega se realizará a través del enlace "Entrega p2" que se muestra en el moodle de la asignatura.

Además, el alumno deberá entregar la memoria en papel, en el despacho 112 del edificio Aulario III.