Prácticas de laboratorio de Telemática II

Práctica 5

Departamento de Ingeniería Telemática

(ENTEL)

Mónica Aguilar
Juanjo Alins
Oscar Esparza
Jose L. Muñoz
Marcos Postigo
Antoni X. Valverde

La composición de este manual de prácticas ha sido realizada con el programa LaTeX 2ε.

Barcelona a 25 de Noviembre de 2002

• Revision 1.6 filter.lyx

Edición electrónica de este documento. Este manual de prácticas se puede conseguir en http://labtel2.upc.es

Índice de las prácticas

Índice de figuras

PRÁCTICA 5

Filtrado de paquetes y NAT

5.1. Introducción

Tanto el filtrado de paquetes como la traducción de direcciones o *Network Address Translation* (NAT) se realizan en los sistemas *unix-like* a nivel de núcleo. Para máquinas con núcleos Linux 2.6 (como las que tenemos en el laboratorio) el filtrado y la traducción de direcciones se realizan mediante la herramienta *iptables* que se ejecuta en el espacio de usuario. Esta herramienta es el interfaz del usuario con el núcleo para especificar que paquetes se deben filtrar o traducir.

5.2. El filtrado de paquetes

El método más utilizado para la protección de las redes son los *firewalls* o cortafuegos. Un *firewall* es un medio que sirve para regular el acceso a una red de computadoras. Para ello, consulta la información identificativa asociada a la comunicación procedente del exterior, el *firewall* decide entonces permitir o no la comunicación de acuerdo a una política de seguridad que ha sido configurada previamente. Usualmente el *firewall* es el *router* que nos comunica con otras redes, con la particularidad de que es capaz de "filtrar paquetes". El filtro de paquetes es un *software* que examina la cabecera de los paquetes de datos según van pasando y básicamente decide si descartar o aceptar el paquete en cuestión. El filtrado de paquetes sirve para realizar las siguientes funciones:

- Control y seguridad: cuando un *router* es lo único entre Internet y nuestra red interna (*intranet*), es indispensable poder permitir ciertos tipos de tráfico, y restringir otros.
- Vigilancia: algunas veces, un *host* mal configurado de la intranet puede emitir paquetes que contengan información "delicada" al mundo exterior, el filtrado de paquetes puede servir para avisarnos en caso de que ocurra algo anormal.

5.3. La traducción de direcciones (NAT)

Normalmente, los paquetes viajan desde su origen a su destino a través de varios *routers*. Como se vió en la practica 4 ninguno de estos *routers* altera realmente el paquete: simplemente lo envía al siguiente *router*. Si uno de estos *routers* hiciera NAT, podría alterar el origen o destino del paquete según pasa a través suyo. Como puede imaginar, ésta no es

la función para la que se diseñó el sistema, y por tanto NAT es siempre un tanto enrevesado. Normalmente, el *router* que realiza NAT recordará cómo modificó el paquete, para poder realizar la acción inversa con el paquete de respuesta, de manera que todo funcione como se espera.

El uso más extendido de NAT es conseguir una reducción del tamaño de los rangos de direcciones IP públicas, y utilizar en su lugar los rangos de direccionamiento privados, transformando en el *router* de salida de la red mediante NAT las direcciones privadas a unas pocas direcciones públicas, este esquema permite además ejecutar cambios del direccionamiento público de forma sencilla y centralizada. Por ejemplo, si en nuestra red no realizamos NAT y cambiamos de proveedor o ISP (*Internet Service Provider*), tendremos que cambiar de rango de direccionamiento público, teniendo que reconfigurar nuestras máquinas. Si se realiza NAT, la configuración de los *hosts* se puede mantener, y sólo deberemos cambiar unas cuantas reglas del *router* que realiza NAT.

NAT se puede dividir de varias formas, en particular una de ellas es dividirlo según se modifiquen parámetros del destino o parámetros del origen:

- NAT por origen o *Source* NAT (SNAT) se produce cuando el *router* altera el origen del paquete, esto es, cambia algún parámetro del lugar de donde viene. SNAT siempre se realiza después del encaminamiento, justo antes de que el paquete salga del *router*.
- NAT por destino o *Destination* NAT (DNAT) se produce cuando el *router* altera el destino del paquete, esto es, cambia algún parámetro del lugar a donde va. DNAT siempre se realiza antes del encaminamiento, cuando el paquete entra al *router*.

5.4. Cómo pasan los paquetes por el núcleo

El núcleo empieza con varias listas de reglas, estas listas se llaman “cadenas” y son las siguientes: PREROUTING, INPUT, OUTPUT, FORWARD y POSTROUTING (tal y como se puede ver en la Figura 5.1).

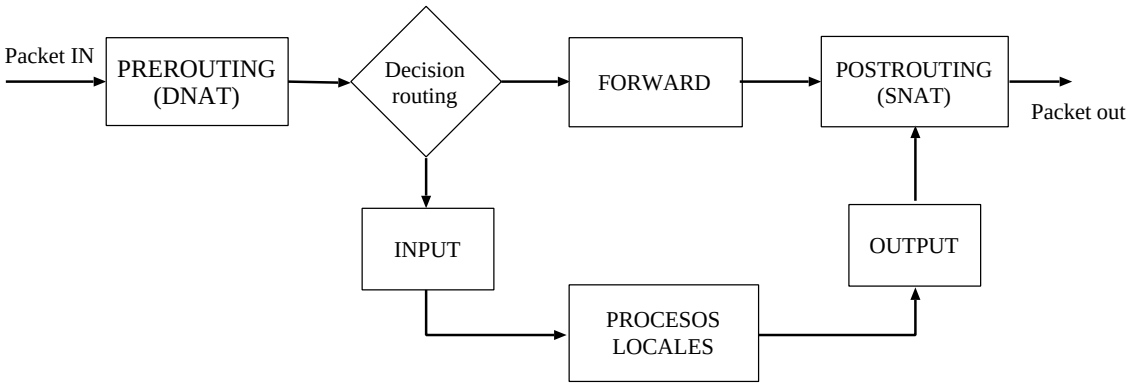

Figura 5.1: Cadenas por defecto del nucleo Linux 2.6.

A continuación se muestran los posibles itinerarios de un paquete por nuestro sistema Linux 2.6:

- Cuando un paquete entra por un interfaz (como por ejemplo la tarjeta Ethernet), entra en la cadena PREROUTING.
- A continuación, el núcleo realiza el encaminamiento de dicho paquete.

- Si el paquete está destinado a esta máquina, entra en la cadena INPUT.
- Si el paquete no está destinado a esta maquina, se comprueba si la máquina puede hacer de *router*, es decir, si está activada la capacidad de reenvío (*forward*). Si esta capacidad no está activada, o aunque esté activada, si no se puede encontrar en las tablas de enrutamiento el siguiente salto (*router*) a donde enviar el paquete, éste es descartado. Si está activado el reenvío y existe una ruta de salida para el paquete, entonces éste pasa a la cadena FORWARD.
- Los procesos que se ejecutan en la máquina también pueden enviar paquetes a la red. Estos paquetes pasan por la cadena OUTPUT.
- Finalmente el paquete antes de salir por un determinado interfaz pasa por la cadena POSTROUTING.

Cuando un paquete alcanza una cadena del diagrama, se examina esa cadena para decidir la suerte del paquete:

- Si la cadena dice que hay que realizar NAT, se realiza cambiando en ese instante la dirección origen o destino correspondiente.
- Si la cadena dice que hay que descartar el paquete, se elimina en ese mismo instante.
- Si la cadena dice que hay aceptarlo, el paquete continúa su camino por el diagrama de la Figura 5.1.

Una cadena no es más que una lista de reglas, donde cada regla dice lo siguiente

```
Si el paquete se parece a esto,
entonces esto otro es lo que hay que hacer con él.
```

Si la regla no se ajusta al paquete, entonces se consulta la siguiente regla en la lista. Al final, si no hay más reglas por consultar, el núcleo mira la política de la cadena para decidir qué hacer. En un sistema consciente de la seguridad, esta política suele decirle al núcleo que descarte el paquete.

5.5. Uso de *iptables* para filtrado de paquetes

Las reglas de filtrado se introducen en tres cadenas:

- En la cadena INPUT para los paquetes destinados a la máquina que llegan a través de los diferentes interfaces.
- En la cadena OUTPUT para los paquetes que salen por un cierto interfaz.
- En la cadena FORWARD para los paquetes que entran por un interfaz y salen por otro (o el mismo!).

La herramienta *iptables* dispone de una página de manual bastante detallada (man iptables), por si necesita más detalles. En las siguientes secciones se muestran las operaciones más comunes que se precisan para la realización de las prácticas.

5.5.1. Operaciones sobre una regla

Cada regla especifica un conjunto de “condiciones” que debe cumplir el paquete, y qué hacer si se ajusta a ellas, es decir, un “objetivo”. Las operaciones más comunes sobre las reglas que forman una cadena son:

1. Añadir una nueva regla a una cadena (**-A**).
2. Insertar una nueva regla en alguna posición de la cadena (**-I**).
3. Mover una regla a otra posición dentro de una cadena (**-R**).
4. Borrar una regla de un sitio en particular de una cadena (**-D**).
5. Borrar la primera regla que coincida con los parámetros dados en una cadena (**-D**).

Por ejemplo, podríamos querer descartar todos los paquetes ICMP que viniesen de la dirección de *loopback* (127.0.0.1). En este caso nuestras condiciones son que el protocolo sea ICMP y que la dirección de origen del paquete sea 127.0.0.1[1]. Nuestro objetivo será DROP (descartar el paquete).

Para probar este ejemplo se pueden ejecutar las siguientes instrucciones:

```
telem2-x# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.2 ms
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
telem2-x# iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
telem2-x# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
telem2-x#
```

En este ejemplo se puede observar que el primer *ping* tuvo éxito. Despues se añadió (**-A**) a la cadena INPUT, una nueva regla que especifica que los paquetes que vengan de 127.0.0.1 (**-s 127.0.0.1**) con protocolo ICMP (**-p icmp**) deben saltar a la cadena DROP (**-j DROP**). La cadena DROP es la encargada de descartar el paquete. Luego probamos nuestra regla, usando el segundo ping. Habrá una pausa antes de que el programa se canse de esperar por una respuesta que nunca llegará.

Podemos borrar la regla de dos maneras. Primero, como sabemos que es la única regla en la cadena, podemos usar un borrado por número:

```
telem2-x# iptables -D INPUT 1
```

Para borrar la regla número uno de la cadena INPUT. La segunda manera es repetir la orden **-A**, pero cambiando **-A** por **-D**. Es útil cuando se tiene una compleja cadena de reglas y no queremos estar contándolas para averiguar que es la regla 37 la que queremos eliminar. En este caso, usaríamos:

```
telem2-x# iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

La sintaxis de **-D** debe tener exactamente las mismas opciones que la orden **-A** (o **-I**, o **-R**). Si hay varias reglas idénticas en la misma cadena, sólo se borrará la primera.

[1] Recuerde que la dirección IP 127.0.0.1 corresponde a la interfaz de *loopback*, de la que se dispone incluso aunque no tengamos una conexión de red real.

Para las especificaciones del filtrado hemos usado **`-p`** para especificar el protocolo, y **`-s`** para la dirección de origen, pero podemos usar otras opciones para especificar las condiciones de los paquetes según se muestra en la Tabla .

Condición
Dirección o direcciones IP de origen
Dirección o direcciones IP de destino
Protocolo
Interfaz
Fragmentos IP
Opciones de protocolo

Tabla 5.1: Condiciones de filtrado

Las direcciones IP de origen se especifican con **`-s`**, **`--source`**, o **`--src`**.

Las direcciones IP de destino se especifican con **`-d`**, **`--destination`**, o **`--dst`**.

Ambas se pueden especificar de cuatro maneras:

1. Nombre: tal como “telem2-9” o www.upc.es.
2. Dirección IP con notación decimal separada por puntos como 147.83.40.29.
3. Un grupo de direcciones mediante una dirección y una máscara indicando el número de bits de red como 147.83.40.0/24.
4. Un grupo de direcciones mediante una dirección y una máscara en notación decimal separada por puntos 147.83.40.0 / 255.255.255.0.

Para especificar una inversión se precede el argumento de la opción por el símbolo **`!`**. Por ejemplo, **`-s ! telem2-9`** coincide con cualquier paquete que no venga de telem2-9.

Para especificar un protocolo se utiliza el indicador **`-p`** o **`--protocol`**. El protocolo se pone usualmente con un nombre simbólico como TCP, UDP o ICMP (no importa si se pone en mayúscula o minúscula).

Para especificar la Interfaz se utilizan las opciones **`-i`** o **`--in-interface`** y **`-o --out-interface`** especifican el nombre de una interfaz con la que coincidir. Los paquetes que pasan por la cadena INPUT no tienen un interfaz de salida, con lo que nunca se activará una regla de esta cadena que use **`-o`**. De forma similar, los paquetes que atraviesan OUTPUT no tienen interfaz de salida, de manera que ninguna regla que use **`-i`** en esta cadena funcionará.

Sólo los paquetes que pasan por la cadena FORWARD tienen a la vez interfaz de entrada y de salida.

Para especificar fragmentos se utiliza la opción **`-f`** o **`--fragment`**. Por ejemplo, la siguiente regla descartará cualquier fragmento dirigido a 192.168.1.1:

```
telem2-x# iptables -A OUTPUT -f -d 192.168.1.1 -j DROP
```

Algunos protocolos ofrecen especificar con más detalle las condiciones del paquete. En la actualidad estos protocolos son TCP, UDP e ICMP. Para estos protocolos podrá especificar nuevas condiciones en la línea de comandos tras la opción **`-p`**.

Condiciones para TCP

`--tcp-flags` permite filtrar dependiendo de ciertos indicadores de TCP. La primera cadena es la máscara: una lista de los indicadores que desea examinar. La segunda cadena indica cuales deben estar activos. Por ejemplo

```
telem2-x# iptables -A INPUT --protocol tcp --tcp-flags ALL SYN,ACK -j DROP
```

Esto indica que deben ser examinados todos los indicadores (ALL es sinónimo de SYN, ACK, FIN, RST, URG, PSH), pero sólo deben estar activos SYN y ACK.

`--syn` es equivalente a –tcp-flags SYN,RST,ACK SYN.

`--source-port` o **`--sport`** especifica el puerto o rango de puertos TCP. Estos pueden estar representados por su nombre, tal como viene en /etc/services, o por su número. Los rangos pueden ser dos nombres de puerto separados por "**`-`**", un puerto seguido de "**`-`**" para especificar un puerto mayor o igual al indicado, o un puerto precedido de "**`-`**" para especificar un puerto menor o igual al indicado.

`--destination-port` o **`--dport`** son lo mismo que lo anterior, sólo que especifican el puerto de destino, en lugar del de origen.

`--tcp-option` se ajusta a paquetes con una opción TCP igual a ese número. Un paquete que no tenga una cabecera TCP completa, será descartado automáticamente si se intenta examinar sus opciones TCP.

Condiciones para UDP

Proporciona las opciones **`--source-port`**, **`--sport`**, **`--destination-port`** y **`--dport`** con los mismos detalles que los indicados para TCP.

Condiciones para ICMP

`--icmp-type` es la única condición proporcionada, y va seguida de un tipo ICMP ya sea en forma simbólica como "host-unreachable", de forma numérica como 3 , o un tipo numérico y un código 3/3. Se puede ver una lista de los nombres de los tipos icmp disponibles usando

```
telem2-x# iptables -p icmp --help.
```

Ahora que sabemos qué condiciones podemos examinar en un paquete IP, necesitamos una manera de decir qué hacer con los paquetes que se ajustan a nuestras pruebas. A esto se le denomina objetivo (*target*) de una regla.

5.5.2. Operaciones sobre una cadena

Como se ha comentado, una cadena especifica qué hacer cuando se ha detectado una coincidencia con las condiciones expuestas. Hay dos objetivos implementados en el sistema por defecto:

- DROP para descartar un paquete.
- ACCEPT para aceptar un paquete

Si una regla coincide con un paquete, y su objetivo es alguno de estos dos, no se consultarán más reglas: la suerte del paquete ha sido decidida.

Además de las cadenas (objetivos) implementados en *iptables* (INPUT, FORWARD, OUTPUT, DROP y ACCEPT) el usuario puede definir otras cadenas para ser utilizadas como objetivo. Por covención las cadenas definidas por el usuario se suelen nombrar en minúsculas para distinguirlas. Cuando un paquete coincide con una regla cuyo objetivo es una cadena definida por el usuario, el paquete empieza a atravesar esa otra cadena. Si en ella no se decide la suerte del paquete, una vez llegado el final de la cadena, se vuelve al mismo punto desde el que se saltó, a la siguiente regla de la antigua cadena.

Las siguientes son las operaciones que se pueden realizar sobre una cadena

1. Crear una nueva cadena (**`-N`**).
2. Borrar una cadena vacía (**`-X`**).
3. Cambiar la política de una cadena de uso interno (**`-P`**).
4. Listar las reglas de una cadena (**`-L`**).
5. Vaciar de reglas una cadena (**`-F`**).
6. Poner a cero los contadores de paquetes y bytes de todas las reglas de una cadena (**`-Z`**).

5.6. Uso de *iptables* para la traducción de direcciones

En Linux las reglas NAT que le dicen al núcleo qué conexiones cambiar, y cómo hacerlo, también se crean utilizando la herramienta *iptables* usando la opción **`-t nat`**.

Las reglas de NAT se introducen en tres cadenas:

- En la cadena PREROUTING para realizar DNAT en los paquetes que llegan a través de los diferentes interfaces al *router*.
- En la cadena OUTPUT para realizar DNAT de los paquetes generados en la propia máquina.
- En la cadena POSTROUTING para realizar SNAT.

En cada uno de los puntos anteriores, cuando un paquete pasa miramos la conexión a la que está asociado. Si es una conexión nueva, comprobamos la cadena correspondiente en la tabla de NAT para ver qué hacer con ella. La respuesta que obtenemos se aplicará a cualquier paquete posterior de esa conexión.

SNAT (Cambio de Origen)

El SNAT se especifica en una determinada regla de la cadena POSTROUTING indicando **`-j SNAT`** y la opción **`--to`** seguida de una dirección IP, un rango de direcciones IP, un puerto o rango de puertos (los puertos sólo son posibles para paquetes de los protocolos UDP y TCP).

Esto significa que cualquier otro servicio de la máquina Linux como el encaminamiento o el filtrado de paquetes verá el paquete sin cambiar y que se deberá utilizar si se precisa la opción **`-o`** para especificar el interfaz de salida.

A continuación se muestran algunos ejemplos para cambiar el origen de los paquetes salientes por el interfaz eth0.

```
telem2-x# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4
```

Para cambiar la dirección origen a la dirección 1.2.3.4.

```
telem2-x# iptables -t nat -A POSTROUTING -o eth0 \
          -j SNAT --to 1.2.3.4-1.2.3.6
```

Para cambiar la dirección de origen a una de las siguientes: 1.2.3.4, 1.2.3.5 o 1.2.3.6

```
telem2-x# iptables -t nat -A POSTROUTING -p tcp -o eth0 \
          -j SNAT --to 1.2.3.4:1-1023
```

Para cambiar la dirección de origen por 1.2.3.4 y a uno de los puertos del rango 1-1023

DNAT (Cambio de destino)

El DNAT se especifica en una determinada regla de la cadena PREROUTING para los paquetes que entran al *router* indicando **`-j DNAT`** y la opción **`--to`**. Esto significa que cualquier otro servicio de la máquina Linux como el encaminamiento o el filtrado de paquetes verá el paquete con su destino final (después de realizar NAT) y que se deberá utilizar si se precisa la opción **`-i`** para especificar la interfaz de entrada.

Para alterar el destino de los paquetes generados en una máquina que hace NAT se debe usar la cadena OUTPUT (esto suele ser más inusual).

A continuación se muestran algunos ejemplos para cambiar el destino de los paquetes entrantes por el interfaz eth1.

```
telem2-x# iptables -t nat -A PREROUTING -i eth1 -j DNAT --to 5.6.7.8
```

Para cambiar la dirección destino a la dirección 5.6.7.8.

```
telem2-x# iptables -t nat -A PREROUTING -i eth1 -j DNAT  \
          --to 5.6.7.8-5.6.7.10
```

Para cambiar la dirección destino a una de las siguientes: 5.6.7.8, 5.6.7.9 o 5.6.7.10.

```
telem2-x# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 -j DNAT \
          --to 5.6.7.8:8080
```

Para cambiar la dirección de destino del tráfico web (puerto 80) por la dirección 5.6.7.8 y el puerto 8080.

```
telem2-x# iptables -t nat -A OUTPUT -d 1.2.3.4 -j DNAT --to 127.0.0.1
```

Para redirigir los paquetes locales que van a 1.2.3.4 hacia el dispositivo *loopback*.

Hay un caso especializado de DNAT llamado redirección: es una simple conveniencia que es exactamente lo mismo que hacer DNAT, pero con la dirección de la interfaz de entrada. Por ejemplo el siguiente comando envía el tráfico que entra dirigido al puerto web (80) al puerto 8080.

```
telem2-x# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 \
          -j REDIRECT --to-port 8080
```

5.7. Ejercicios

Ejercicio 1

En este ejercicio se pretende que se familiarice con los conceptos básicos de filtrado y traducción de direcciones. Para ello se utiliza la estructura de la Figura 5.2 .

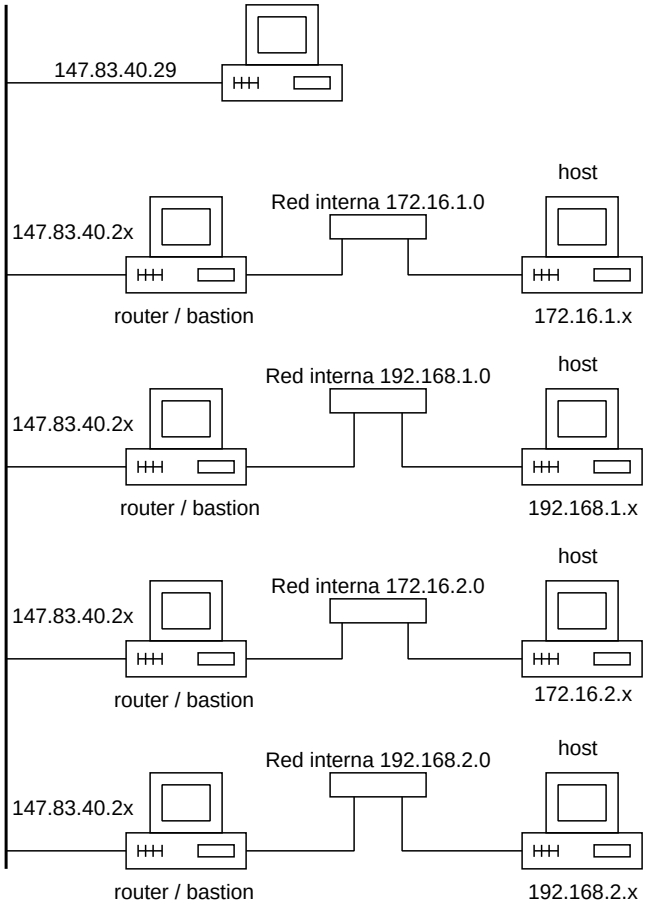

Figura 5.2: Estructura para el ejercicio 1

1. Cada máquina tendrá una tabla de rutas donde estarán configuradas sus rutas directas y una ruta por defecto. La ruta por defecto para los *hosts* será a través de su *router*, mientras que para los *router* la ruta por defecto será a través de 147.83.40.29 (telem2-9). Configure dichas rutas.

2. Configure sus tablas de filtrado para no permitir ningún tipo de tráfico ICMP entrante. Con este filtrado responda a las siguientes preguntas:

a) Si la otra máquina de su subred le envía un ping a la suya ¿se transmitirá un paquete ICMP *echo-request*? ¿y un *echo-reply*? Describa lo que ocurre en este caso.

b) Si realiza un ping desde su máquina a la otra máquina de su subred ¿se transmitirá un paquete ICMP *echo-request*? ¿y un *echo-reply*? Describa lo que ocurre en este caso.

3. Borre la configuración anterior y vuelva a configurar en sus tablas de filtrado para que usted pueda realizar un `ping` a una máquina remota, pero su máquina no responda ninguna petición de `ping` externo. Responda a las preguntas del apartado anterior.

   a) El problema de los esquemas de filtrado anteriores es que hay que configurar las tablas de filtrado en cada una de las máquinas, haciendo que la administración de la red sea muy compleja. La solución más utilizada es "confiar" la seguridad al router de la red, ya que todas las comunicaciones fluyen a través de él y se puede aplicar un control centralizado a las mismas facilitando la administración. Cuando el *router* realiza funciones de filtrado o *firewall* se le conoce con el nombre de bastión de la red.
   Configure su subred con un bastión, para ello elimine las entradas en las tablas de filtrado del *host* de su subred y añada las entradas necesarias en su bastión.
   Conéctese a telem2-9 (pídale al profesor el *login* y *password*) y compruebe que su configuración funciona correctamente.

      1) Desde el *host* de su subred, realice un `ping` a telem2-9 ¿funciona? ¿Es un problema de filtrado o de direccionamiento?

      2) Para solucionar el problema anterior configure el bastión para que realice SNAT para su subred. Una vez configurado pruebe a realizar el `ping` a telem2-9 ¿funciona ahora? Utilice las herramientas de análisis de tráfico que conoce para ver que está sucediendo en la red.

Ejercicio 2

En Figura 5.3 se muestra el típico esquema de *firewall* con doble bastión (bastion externo y bastión interno), zona desmilitarizada o DMZ (*DeMilitarized Zone*) para los servidores con acceso externo, y red interna. En este esquema las máquinas de la red interna pueden establecer conexiones a los servidores de la DMZ y a servidores externos (Internet). Desde el exterior se puede acceder a los servidores de la DMZ pero no a los *hosts* de la red interna.
Proponga e implemente, con sus compañeros, una configuración que cumpla los requisitos anteriores.

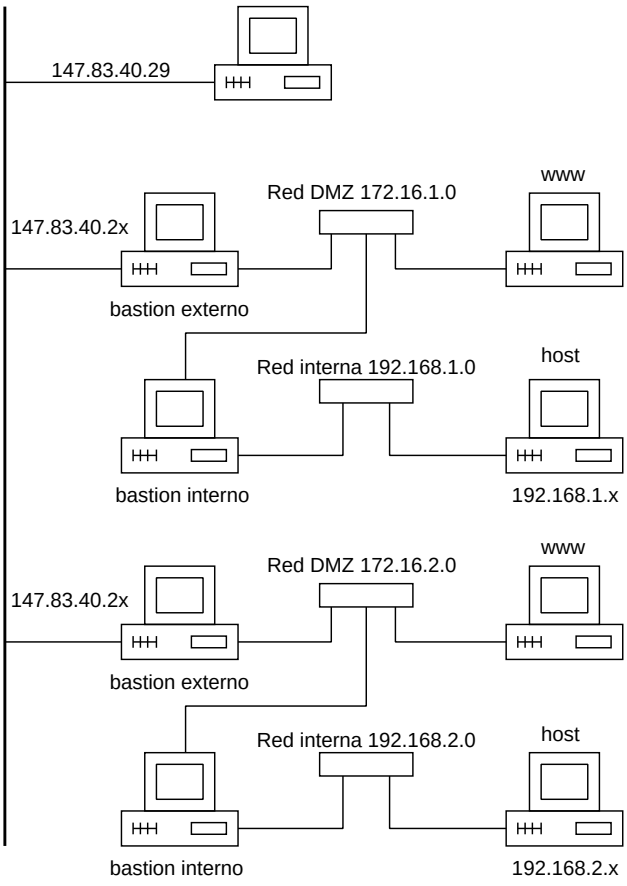

Figura 5.3: Estructura para el ejercicio 2.