Práctica 1.3. Configuración de un proxy directo: TinyProxy

Redes y Seguridad II

Práctica 1.3 – Configuración de un proxy directo: TinyProxy

Grado en Ingeniería Informática

Objetivos

Esta práctica está pensada para que el estudiante aprenda a desplegar un proxy directo.

Índice

Calificación

Esta práctica se califica con 3 puntos máximo y hace media con el resto de prácticas de la asignatura

Equipo

Nombre y Apellidos	Rol
	Programador/a
	Analista

La misión principal del *Programador* es escribir todos los comandos indicados en la práctica en el orden correcto para conseguir un buen aprovechamiento de los objetivos buscados en la misma.

La misión principal del *Analista* es tomar nota del trabajo que está haciendo su pareja de prácticas e intentar buscar la respuesta a todas las preguntas que se plantean en la práctica. Además, es el encargado de contestar el test asociado a la práctica.

Este cuadernillo de prácticas podrá ser requerido por los profesores y profesoras de la asignatura en cualquier momento a lo largo del curso para comprobar y/o recalificar el aprovechamiento del laboratorio.

Configuración de la red

En la práctica 1.3 vamos a usar la estructura de red de la Figura 1. Para ello, vamos a realizar **3 clonaciones** de la máquina virtual rys24.ova (usuario:usuario/contraseña:usuariorys): Router 1, Host 2 y Servidor HTTP (recuerda **generar nuevas direcciones MAC** y realizar **clonación enlazada**).

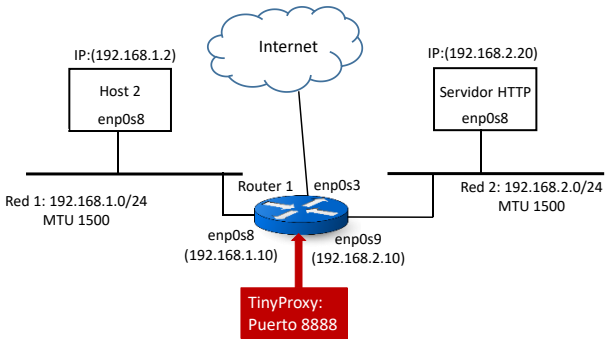


Figura 1

Si en el entorno donde estás realizando la práctica **ya existen las máquinas** Host 2, Router 1 y Servidor HTTP no necesitarás hacer la clonación, sino que **puedes reutilizarlas**. En este caso, **comprueba** que las distintas **interfaces** de las máquinas están **conectadas** a las **redes internas** que se indican en la Figura 1.

A continuación, vamos a describir los **pasos a seguir para configurar nuestro entorno de trabajo**:

1. Comprueba que las máquinas tienen los adaptadores de red que se necesitan habilitados y conectados a la red correspondiente (por ejemplo, Host 2 debe tener el adaptador 1 conectado a NAT y el adaptador 2 conectado a la red interna Red 1).
2. Arranca las máquinas.
3. Configura los interfaces de red de las máquinas que no hayas reutilizado y comprueba que todo está correcto en las reutilizadas.
4. Define en cada máquina los encaminadores correspondientes para que estas máquinas puedan alcanzar la red a la que no pertenecen. **En Host 2 vamos a definir el encaminador como un** encaminador por defecto (para poder salir al exterior a través de Router 1):

   ```
   [Host 2] $ sudo ip route add default via 192.168.1.10
   ```

   No te olvides de activar el forwarding en Router 1.
5. Comprueba que todas las máquinas están correctamente conectadas, haciendo, por ejemplo, un ping de Host 2 a Servidor HTTP.
6. Comprueba que el servidor apache2 está arrancado en la máquina Servidor HTTP:

   ```
   $ sudo systemctl status apache2
   ```

   Si no estuviese activo, arrancarlo con:

   ```
   $ sudo systemctl start apache2
   ```

7. Desconecta las máquinas Host 2 y Servidor HTTP del exterior (en la ventana de VirtualBox, sobre cada máquina, selecciona Configuración ->Red y desmarca la opción Cable conectado del adaptador de red 1).

Documentación y configuración

En la máquina Router 1 vamos a usar un proxy que trabaja en la capa de aplicación y, por tanto, será específico de un determinado protocolo. En nuestro caso, necesitamos un proxy HTTP y hemos elegido Tinyproxy, que ha sido diseñado para ser rápido y ligero. Proporciona filtrado de URLs, monitorización de conexiones, control de acceso, inserción y borrado de cabeceras HTTP…

Ejercicio 1: Configuración de Tinyproxy.

1. Consulta la página del manual (haciendo `man tinyproxy` en el Terminal y en https://linux.die.net/man/5/tinyproxy.conf, **se recomienda especialmente esta última para entender el fichero de configuración**) y revisa su configuración en /etc/tinyproxy/tinyproxy.conf.

   Contesta las siguientes preguntas:

Sobre qué puerto trabaja tinyproxy:
¿Qué función tiene la línea Listen?
¿Qué función tiene la línea Bind?
¿Qué ficheros están activos por defecto? ¿Cuál es su función?
¿Qué función tiene la línea Allow?
¿Qué función tiene “Anonymus Cookie”?

Si has comprendido el significado de la línea Allow en el archivo de configuración del Tinyproxy entenderás que tienes que modificar la configuración por defecto para que se ajuste a la IP de tu red, es decir, tienes que indicar que la red que tiene permiso para acceder a Tinyproxy es Red 1.

2. Edita el archivo de configuración y añade la Red 1 a la lista de redes permitidas (es necesario hacerlo con sudo). Apunta cómo ha quedado tu configuración (se pueden obviar las líneas comentadas):

```
# Allow: Customization of authorization controls. If there are any
# access control keywords then the default action is to DENY. Otherwise,
# the default action is ALLOW.
#
# The order of the controls are important. All incoming connections are
# tested against the controls based on order.
#
Allow
```

3. Ahora asegúrate de que Tinyproxy lee correctamente el fichero de configuración que has modificado:

```
$ sudo /etc/init.d/tinyproxy reload
```

4. Arranca Tinyproxy con la nueva configuración:

```
$ sudo /etc/init.d/tinyproxy restart
```

5. Asegúrate de que el proxy está funcionando y listo para aceptar conexiones, consultando el archivo de log de Tinyproxy:

```
$ sudo tail –n 25 /var/log/tinyproxy/tinyproxy.log
```

Apunta aquí lo que ves en el archivo de log:

¿En qué parte del fichero de configuración se indica cuál es el número máximo de clientes que se pueden conectar a la vez? ¿Qué valor tiene asignado? ¿Se podría modificar este valor?

Configuración del Firewall en Router 1

En la práctica 1.1 se explicó que la configuración más habitual de un firewall es una política de descarte por defecto junto a las reglas que definen el tráfico que está permitido. En esta práctica necesitamos configurar el firewall de Router 1 para que **todo el tráfico HTTP proveniente de los hosts de Red**

1 (Host 2) pase por el proxy, que estará instalado en Router 1. Además, **solo se permitirá pasar por el router el tráfico que haya sido iniciado en la Red 1**, es decir, no se aceptará ningún paquete que venga de Red 2 ni de Internet, salvo que pertenezca a una conexión ya establecida.

Ejercicio 2: Reglas Iptables para la configuración de un firewall de inspección de estado y para un proxy.

1. Comprobamos que la política de iptables que tenemos por defecto es aceptar todo:

   ```
   $ sudo iptables –L -v
   ```

2. Cambiamos la política por defecto de iptables en todas las cadenas para que descarte cualquier paquete:

   ```
   $ sudo iptables -P INPUT DROP
   $ sudo iptables -P OUTPUT DROP
   $ sudo iptables -P FORWARD DROP
   ```

Con esta configuración, ningún paquete puede llegar ni salir de Router 1 ni atravesarlo.

3. Comprueba ejecutando `$ ping 192.168.2.20` y `$ wget 192.168.2.20`, que Host 2 no puede realizar ninguna conexión con el servidor.

4. Ahora, añadiremos una serie de reglas en el firewall de Router 1 para que las conexiones HTTP y HTTPS iniciadas desde Red 1 puedan llegar al proxy por su puerto 8888 (cadena INPUT) y salir de Router 1 hacia el Servidor HTTP (o hacia Internet) con la IP de Router 1 como IP origen (cadena OUTPUT). Trabajamos con las cadenas INPUT y OUTPUT porque el proxy actúa como un intermediario. Será necesario permitir también las consultas DNS para poder conectarnos con un servidor web mediante una url.

Cadena INPUT:

Introduce las siguientes reglas:

- Aceptar paquetes entrantes recibidos por la interfaz interna (`enp0s8`) con IP origen Red 1 con puerto destino 8888 (donde escucha el proxy).

Escribe la regla ¿Por qué especificas que la IP origen sea de Red 1? ¿Qué quieres evitar?

- Aceptar paquetes entrantes de conexiones establecidas.

Cadena OUTPUT:

Introduce las siguientes reglas:

- Aceptar paquetes salientes con puerto destino HTTP.

- Aceptar paquetes salientes con puerto destino HTTPS (443).

- Aceptar paquetes salientes con protocolo udp y puerto destino 53 (consultas DNS).

- Aceptar paquetes salientes de conexiones establecidas.

Para especificar conexiones establecidas en una regla de iptables utiliza la opción:
-m state - - state ESTABLISHED

Cadena FORWARD:

Seguimos manteniendo la política de descartar por defecto de todos paquetes. Si desde Red 1 solo se permitiesen conexiones HTTP/HTTPS con el exterior a través del proxy, ¿sería necesario añadir reglas a esta cadena para aceptar paquetes de este tipo?

Con esta configuración solamente estamos permitiendo a las máquinas de Red 1 que se conecten a servicios HTTP/HTTPS a través del proxy.

5. Comprueba que desde Host 2 no puedes conectarte al Servidor HTTP (con el navegador o ejecutando el comando `$ wget 192.168.2.20`) ni realizar un ping. ¿Por qué no puede conectarse el Host 2 al Servidor HTTP, aunque Tinyproxy esté instalado y el tráfico dirigido al puerto 8888 de Router 1 permitido?

BONUS TRACK

Lo normal es que una organización configure sus equipos para que todas las conexiones HTTP/HTTPS salgan al exterior a través de un proxy web, pero que desde estos equipos se permitan también otro tipo de conexiones, como conexiones ssh. Para que estas conexiones funcionen si el cortafuegos se configura en el router, será necesario introducir reglas de iptables en la cadena FORWARD.

Este apartado es optativo. Vamos a introducir una serie de reglas en la cadena FORWARD de Router 1 que permitan establecer cualquier tipo de conexión desde cualquier máquina de Red 1 con el exterior, salvo conexiones HTTP/HTTPS que deberán ir a través del proxy. Para ello, necesitarás **introducir las 4 reglas** siguientes:

- Aceptar paquetes reenviados recibidos por la interfaz interna (enp0s8) con IP origen Red 1.

_____

- Descartar paquetes reenviados recibidos por la interfaz interna con puerto destino HTTP. Esta regla es necesaria para que todo el tráfico HTTP vaya por el proxy.

  ¿Importa la posición en la que se introduzca esta regla? Razona la respuesta.

  _____

  ¿Por qué se tiene que introducir esta regla?

  _____

  Escribe la regla

  _____

- Descartar paquetes reenviados recibidos por la interfaz interna con puerto destino HTTPS. Esta regla es necesaria para que todo el tráfico HTTPS vaya por el proxy.

  _____

- Aceptar paquetes reenviados pertenecientes a conexiones establecidas.

  _____

Proxy configurado en cliente

Para que todas las conexiones web de Red 1 con el exterior pasen a través del proxy es necesario redirigir el tráfico HTTP de las máquinas de Red 1 al proxy, y para ello una opción es configurar cada uno de los clientes web de las máquinas de Red 1 para que redirijan este tráfico al proxy.

Ejercicio 3: Acceso al servidor web configurando el cliente web de Host 2.

Para conectarnos al Servidor HTTP y entender cómo funciona el proxy, vamos a seguir los siguientes pasos:

1. Abre el navegador en Host 2 y **configúralo para que todos los mensajes HTTP/HTTPS se envíen al Router 1 a través del puerto 8888, donde escucha el proxy**. Para ello en el

navegador hay que pulsar en *Ajustes* (ver Figura 2) y después en la pestaña *Configuración* (en Configuración de Red al final de General). Se desplegará la ventana de *Configuración de conexión*, en ella elegir *Configuración manual de proxy* y en *HTTP proxy* escribir 192.168.1.10 y en *Port* 8888.

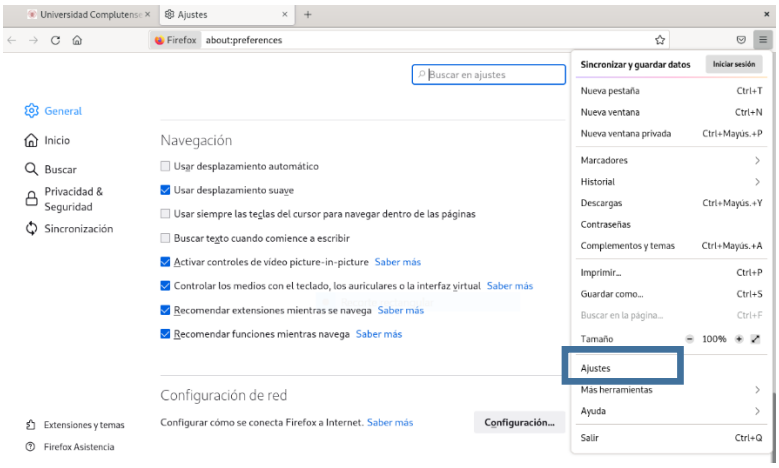


Figura 2

2. Comprueba que ahora sí puedes acceder al servidor desde el navegador de Host 2 escribiendo en la barra de navegación http://192.168.2.20.

7. Si no quieres usar el navegador la conexión web también se puede hacer con wget desde el Terminal (en ese caso no es necesario cambiar la configuración del navegador). Si lo prefieres hacer así, deberás ejecutar (desde el directorio /home/usuario):

```
$ env http_proxy="http://192.168.1.10:8888" wget 192.168.2.20
```

3. Comprueba el fichero de registro de Tinyproxy (en Router 1):

```
$ sudo tail -f /var/log/tinyproxy/tinyproxy.log
```

4. Cierra el navegador en Host 2.

Ahora vamos a **analizar los paquetes TCP que se intercambian a través del Router 1 para entender cómo se realiza la comunicación utilizando un proxy**.

5. Arranca Wireshark en Router 1. Configúralo para que escuche en la interfaz `enp0s8` y filtra los paquetes para ver únicamente los de tipo TCP (Escribe tcp y pulsa Enter, tal y como se muestra en la Figura 3).

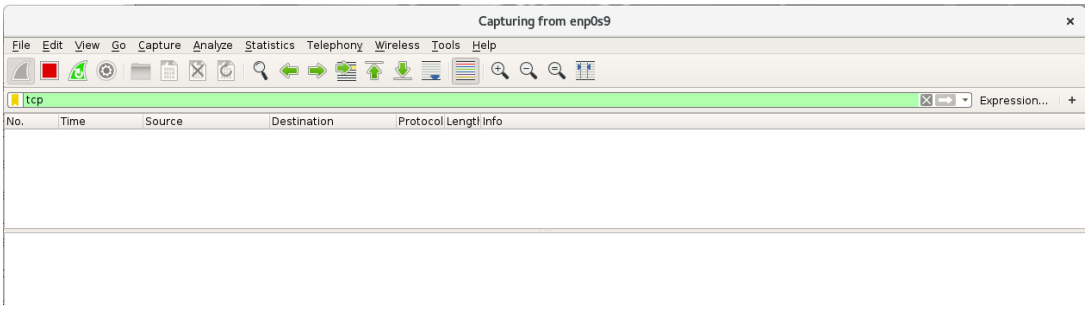


Figura 3

6. Abre de nuevo el navegador en Host 2 y **configúralo para que no use proxy** (sigue las instrucciones indicadas en el paso 2 pero ahora elige la opción *No proxy*). Intenta conectarte con el servidor desde Host 2 y observa que al Router 1 solo llegan los mensajes SYN de Host 2. Captura ahora por la interfaz `enp0s9`, vuelve a intentar conectarte al servidor desde Host 2 y comprueba que no se ve ningún mensaje, ¿por qué?

7. Para la captura de Wireshark (Captura → Detener) y vuelve a capturar el tráfico por la interfaz `enp0s8`.

8. Configura de nuevo en Host 2 el navegador para que las conexiones web vayan a través del proxy. Intenta conectarte con el servidor y observa la captura de Wireshark. **Identifica los 3 primeros mensajes de establecimiento de conexión TCP**. ¿Qué IP origen aparece? ¿Qué IP destino? ¿Qué puerto destino? ¿Por qué no aparece la IP 192.168.2.20 y sin embargo sí que estás conectado con el servidor?

   Busca el primer mensaje en el que aparece la IP del Servidor HTTP (no tiene que estar en las cabeceras IP). ¿Qué tipo de mensaje es? ¿Qué IP destino aparece en la cabecera IP? ¿Y el puerto?

9. Vuelve a cerrar el navegador en Host 2 y reinicia la captura de Wireshark para que escuche en la interfaz `enp0s9`.
10. Arranca de nuevo el navegador en Host 2, conéctate con el servidor y observa el tráfico que sale de Router 1 hacia el Servidor HTTP.

    ¿Qué IP origen y destino aparecen en los mensajes? _____

    ¿Qué puertos? _____

    ¿A qué máquina contesta el servidor ?_____

    Ahora en el mensaje que contiene el comando GET ya no aparece la dirección del Servidor HTTP en el campo datos, ya que este mensaje va dirigido al servidor y la IP de esta máquina es la que aparece en la cabecera IP como dirección IP destino.

Conexiones HTTP/HTTPS fuera del entorno virtual

Los dispositivos que funcionan como proxy web directo, además de proteger a las máquinas de la red interna evitando que las direcciones IP de las máquinas salgan al exterior, se usan para restringir las páginas web a las que pueden acceder los empleados de una organización o los menores de edad si hablamos de un entorno familiar.

Ejercicio 4: Configuración de los filtros de tinyproxy para prohibir el acceso a determinadas páginas web.

Vamos a acceder desde Host 2 a un servidor web de Internet y comprobamos que se puede acceder correctamente. En el navegador de Host 2 accede a https://www.ucm.es. La conexión funcionará, aunque puede ir un poco lenta.

A continuación, vamos a configurar el tinyproxy para no permitir el acceso a determinadas páginas web.

Accede al fichero de configuración de tinyproxy (`/etc/tinyproxy/tinyproxy.conf`). Buscar en este fichero cuál es la política de filtrado por defecto (*filter*).

¿Cuál es? _____

Si mantenemos la política de filtrado por defecto el fichero `/etc/tinyproxy/filter` contiene una lista negra con las direcciones web a las que los usuarios de la organización no podrán acceder. Si por el contrario, cambiamos la política de filtrado del tinyproxy, este fichero se convierte en una lista blanca y contendrá las url que podrán ser accesibles desde la red interna.

En esta práctica vamos a mantener la política de filtrado por defecto del proxy y vamos a suponer que desde los ordenadores de la UCM no se permite el acceso a la página web de la UAM, por lo que habrá que introducir su url en la lista negra. Para ello, habrá que seguir los siguientes pasos:

1. Introduce en el fichero `/etc/tinyproxy/filter` la línea uam.es
2. Descomenta en el fichero `/etc/tinyproxy/tinyproxy.conf` la línea `/etc/tinyproxy/filter`.
3. Reinicia el tinyproxy:

```
$ sudo /etc/init.d/tinyproxy reload
$ sudo /etc/init.d/tinyproxy restart
```

Abre el navegador en Host 2 (comprueba que aún te sigues conectando a través del proxy) y trata de acceder a la página https://www.uam.es/.

¿Qué te devuelve el navegador?

Proxy transparente

A continuación, vamos a introducir una nueva regla de Iptables para configurar el proxy de forma que sea transparente al cliente. El navegador de Host 2 creerá que se está comunicando directamente con el servidor, pero todos los mensajes con destino esta máquina se redireccionarán al puerto 8888 de Router 1, por lo que pasarán a través del proxy.

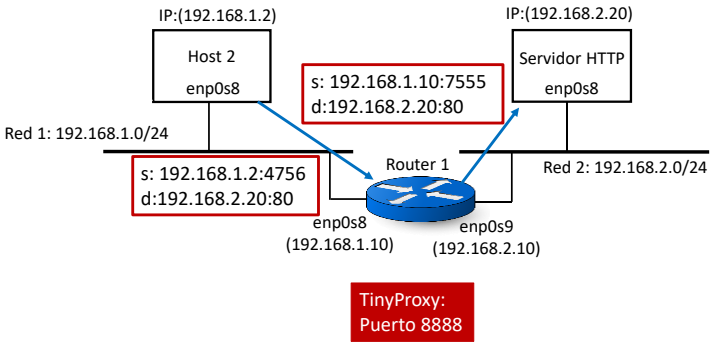


Figura 4

Ejercicio 5: Acceso al servidor web a través de un proxy transparente.

Para no tener que configurar los navegadores de las máquinas de la Red 1, se puede usar un proxy transparente. Esto se haría añadiendo una regla en la tabla NAT de Router 1 que redirija las peticiones web al proxy. La configuración del proxy como transparente no funciona correctamente para tinyproxy cuando las conexiones son con Internet, por lo que solo la vamos a probar con conexiones con el Servidor HTTP.

Vamos a seguir los siguientes pasos:

1. Añade en Router 1 esta regla que redirige las peticiones web al proxy:

```
$ sudo iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 80 -j REDIRECT
  --to-port 8888
```

2. Apaga el proxy para comprobar que la conexión se está realizando a través del proxy:

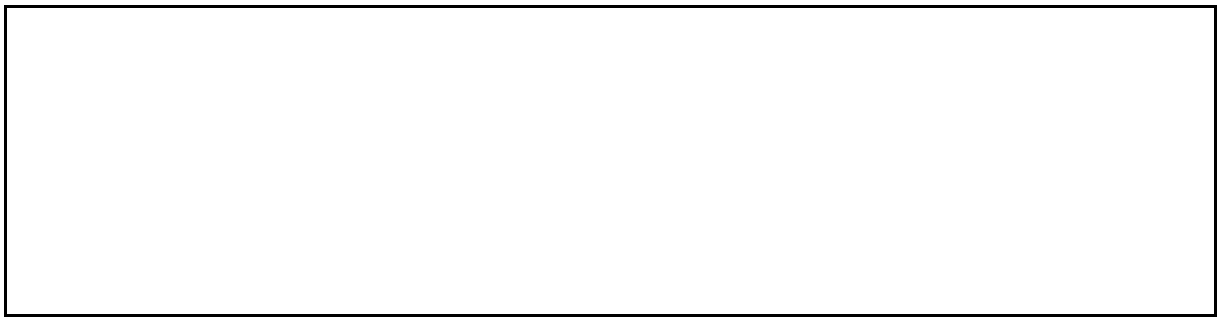
```
$ sudo /etc/init.d/tinyproxy stop
```

3. Vuelve a **configurar el navegador en Host 2 para que no use proxy**. A continuación, intenta volver a conectarte al Servidor HTTP. ¿Puedes conectarte ahora? ¿Por qué?

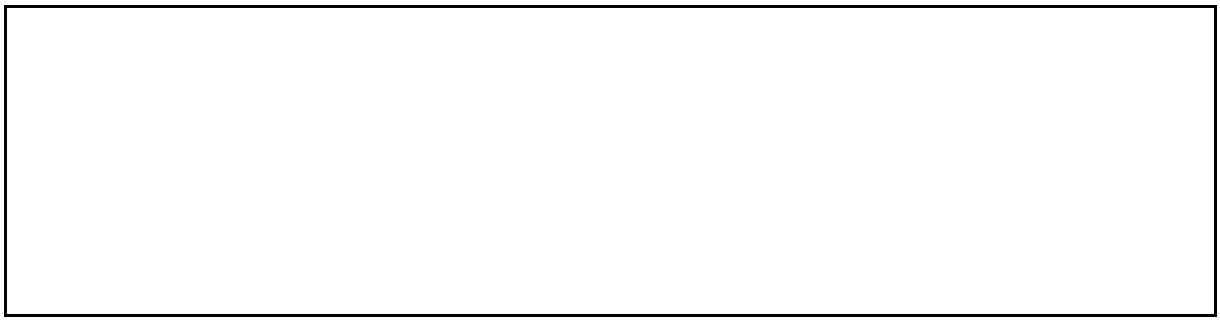
3. Configura de nuevo el Wireshark en Router 1 para capturar tráfico por la interfaz `enp0s8.`
4. Arranca de nuevo el proxy:

   ```
   $ sudo /etc/init.d/tinyproxy start
   ```

5. Desde el navegador de Host 2 vuelve a intentar conectarte al Servidor HTTP. En el Wireshark de Router 1, observa el tráfico en `enp0s8` correspondiente al proxy transparente. ¿Qué diferencias observas con respecto a las IPs y puertos que aparecían cuando el proxy estaba configurado en el cliente? ¿Por qué esta diferencia?



6. Configura el Wireshark para que capture el tráfico por la interfaz `enp0s9`. Desde Host 2 vuelve a conectarte al servidor a través del navegador y observa el tráfico capturado correspondiente al proxy transparente. ¿Hay diferencias con respecto a la configuración del proxy en el cliente? ¿Por qué?



Aquí termina la parte de la práctica dedicada a la configuración del del proxy web.

Conexiones al exterior desde una red interna a través de un router NAT

En esta parte de la práctica vamos a tratar de entender lo que ocurre en nuestras casas (o en cualquier organización) cuando los distintos dispositivos que tenemos conectados a la red se conectan al exterior. Lo normal es que estos dispositivos salgan al exterior a través del router, que estará configurado en modo NAT. Modo NAT significa que el router recibe las peticiones de la red interna y las envía a su destino (en Internet) usando como IP origen su propia dirección IP. Cuando llega la respuesta, el router es capaz de identificar a qué máquina interna va dirigida la respuesta y se la envía.

Para poder reproducir esta configuración, la Red 1 va a representar la red de nuestra casa y Router 1 será nuestro router por el que los paquetes saldrán al exterior. Para ello, todas las máquinas de la red tendrán configurado el Router 1 como su router por defecto, para que cualquier paquete dirigido al exterior salga por él. Los paquetes saldrán del Router 1 a Internet por la interfaz enp0s3 en modo NAT. El problema es que si usamos únicamente los comandos de configuración vistos hasta ahora, cuando el router recibe el paquete de la red interna con destino una máquina de Internet lo enviará con IP origen la que figure como IP de su interfaz enp0s3 y ese paquete llegará a destino. Sin embargo, cuando llega la respuesta, el router entenderá que el destinatario de ese paquete es él y la máquina que ha enviado el paquete no recibirá la respuesta buscada. Así que será necesario ejecutar un comando en Router 1 que le permita guardar información sobre las solicitudes que le llagan desde la Red 1 y cuando reciba mensajes de exterior en respuesta a estas solicitudes sepa a qué máquina de Red 1 debe enviarlos.

Ejercicio 6: Conexión desde Host 2 al exterior a través de un router en modo NAT.

Primero, vamos a comprobar que efectivamente con la configuración usada hasta ahora Host 2 no se podrá comunicar con el exterior.

1. Cambia la política por defecto de la cadena FORWARD de Router 1 para que puedan pasar los mensajes (que no pertenecen a conexiones web) hacia el exterior.

```
$ sudo iptables -A FORWARD -p icmp -j ACCEPT
$ sudo iptables -A OUTPUT -p icmp -j ACCEPT
```

2. Ejecuta `$ ping 8.8.8.8` desde Host 2. ¿Funciona?_____
4. En Router 1 vamos a reconstruir los paquetes que nos llegan de la Red 1 para que puedan salir al exterior y sepan volver, para ello tenemos que introducir una regla en la tabla NAT de iptables con la acción `MASQUERADE`.

```
[Router 1] $ sudo iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o enp0s3 -j
MASQUERADE
```

¿Qué significan cada una de las opciones de esta regla?

5. Ejecuta `$ ping 8.8.8.8` desde Host 2. Ya debería funcionar correctamente, porque los paquetes saben salir y volver.

ANOTACIONES