

Configuración de un servidor Proxy



Orlando Alemán Ortiz
Samuel Díaz Cabrera

4º Ing. Informática
Curso 2005/06

Licencia



Esta obra ha sido publicada bajo licencia "Reconocimiento-NoComercial-CompartirIgual 2.5 Spain" de Creative Commons, la cual implica que:

Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciadador.



No comercial. No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Y además:

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.

Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/es/> o envíe una carta a Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Índice de contenidos

1. Previos.....	2
Introducción.....	2
Equipamiento empleado.....	3
2. Desarrollo.....	4
Instalación del software.....	4
Instalación binaria.....	4
Instalación desde código fuente.....	5
Desinstalación.....	6
Configuración de Squid.....	6
Tamaño de disco de 50 MB.....	6
Realizar el logging de las solicitudes de los clientes.....	6
1 padre, 1 hermano solo para .com y 1 hermano para .es.....	7
Permitir solicitudes de clientes solo de la interfaz de la red interna.....	7
Permitir solicitar conexiones con servidores y caches vecinos por la interfaz externa.....	7
Prohibir como mínimo las siguientes direcciones: Playboy.com y Playgirl.com.....	8
Postconfiguración.....	8
Realización de pruebas.....	9
3. Referencias.....	13

1. Previos

Introducción

En esta última práctica de la asignatura montaremos un servidor proxy, que no es más que un computador que intercepta las conexiones de red que un cliente hace hacia un servidor destino. Nos centraremos en el más común, el proxy web, cuya finalidad específica es la de proporcionar cache de páginas web y contenidos asociados, y filtrar ciertas páginas o contenidos en base a criterios de restricción establecidos por el administrador.

Ventajas:

- Las peticiones de páginas Web se hacen al servidor Proxy y no a Internet directamente.
- El servidor Proxy crea un caché que evita transferencias idénticas de información entre servidores.
- Filtrado de contenido en base a criterios de restricción establecidos por el administrador.
- Modificación de contenidos (Privoxy): Puede bloquear direcciones y Cookies por expresiones regulares y modificar el contenido de una petición.

(Fuente: [**WIK1**])

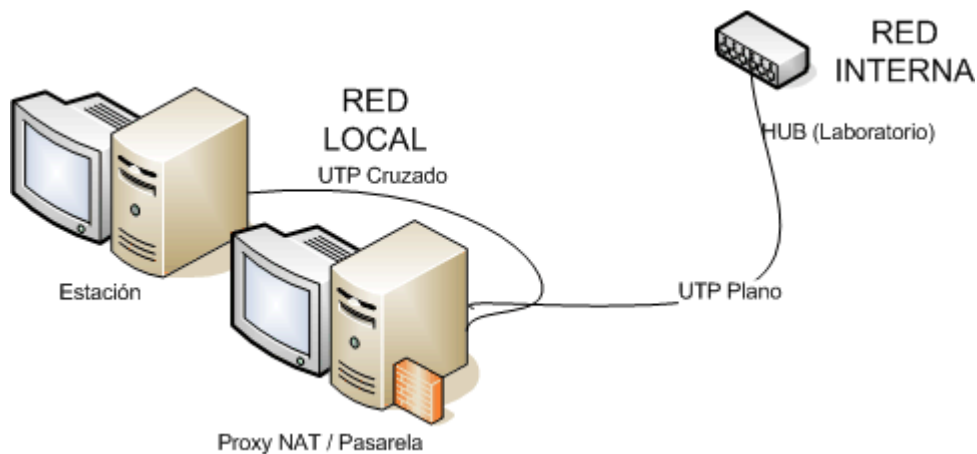
Desventajas:

- Las páginas almacenadas pueden no estar actualizadas.
- El hecho de realizar conexiones indirectas a Internet puede impedir el uso de ciertos puertos o protocolos
- Violación de la intimidad para algunas personas.

(Fuente: [**WIK1**])

Para realizar esta tarea trabajaremos, como siempre, sobre nuestra subred, compuesta de dos ordenadores, uno de ellos con acceso directo a Internet y el otro accediendo a través del primero mediante “routing”. El esquema de conectividad de esta red puede verse a continuación:

Práctica 6: Puesta en marcha de un servidor Proxy



Todo el proceso se llevará a cabo, como siempre, utilizando *Fedora Linux* como entorno sistema operativo residente en ambas máquinas. El software escogido para la implementación del proxy es Squid, gratuito y de código abierto.

Squid es un programa que sirve de proxy-cache de Internet. Es decir, funciona como una cache de Internet. Una vez es accedida una página web, esta se almacena en el disco duro del servidor, de forma que en un futuro acceso a la misma página, se devuelva al usuario la página almacenada en lugar de obtenerla de Internet. Este intermediario da sensación de mayor fluidez en la navegación por Internet, mejorando la experiencia de usuario.

Equipamiento empleado

Hardware:

- 1 x PC con 2 tarjetas de red, que actuará como pasarela.
- 1 x PC con 1 tarjeta de red, que actuará de estación.
- 1 x cable de red UTP Cat.5 plano con conectores RJ45
- 1 x cable de red UTP Cat.5 cruzado y con conectores RJ45

Software:

- Por cada PC, una instalación de *Fedora Linux*.
- En las instalaciones de *Fedora*, el paquete *squid*.

2. Desarrollo

En este apartado explicaremos cómo poner en funcionamiento un servidor de proxy-cache mediante Squid. Lo haremos, como siempre, cumpliendo todos los puntos marcados como obligatorios para esta práctica, y que no son más que los siguientes:

- Tamaño de cache de disco 50 MB
- Realizar el logging de las solicitudes de los clientes
- 1 padre
- 1 hermano sólo para .com
- 1 hermano sólo para .es
- Permitir solicitudes de clientes solo de la interfaz de la red interna
- Permitir solicitar conexiones con servidores y caches vecinos por la interfaz externa
- Prohibir como mínimo las siguientes direcciones:

Playboy.com

Playgirl.com

Instalación del software

Antes de ejecutar cualquier otro paso, comprobamos si “squid” está instalado ya en el sistema con:

```
$ rpm -q squid
Package squid is not installed
```

Como la respuesta ha sido negativa, pasamos a instalarlo. Existen dos posibilidades. Instalar un binario precompilado, con las opciones generales que Fedora estimó oportunas u obtener el código fuente de la última versión estable de la aplicación y compilarla nosotros mismos.

En caso de que la respuesta haya sido afirmativa, podríamos usar esa misma versión ya instalada, o desinstalarla y proceder como en una nueva instalación. Para desinstalar un paquete RPM se suele utilizar:

```
$ rpm -e squid
```

Instalación binaria

La que realizaremos en esta práctica y la preferible desde el punto de vista de la coherencia del sistema, ya que permite tener un mejor control de las aplicaciones instaladas en el sistema (a través de la base datos RPM).

Práctica 6: Puesta en marcha de un servidor Proxy

Como se ha visto, el nombre del paquete binario que nos interesa es “squid”. Puede ser obtenido directamente desde los CDROM o haciendo uso de la herramienta “yum”. La ventaja de la segunda estrategia frente a la primera, es que en caso de existir dependencias con otros paquetes, éstas serían resueltas por la propia aplicación en lugar de tener que resolverlas nosotros mismos. Por contra, deberemos disponer de acceso a Internet desde la máquina implicada.

Se trata de una compilación “made in Fedora”, por lo que seguramente haya sido generado con opciones más generales posibles. Si no fuera así, también podría compilarse la fuente del RPM, pero no es nuestro objetivo llegar ese extremo.

La instalación mediante YUM la haríamos con el siguiente comando:

```
$ yum install <paquete>
```

En cambio, si deseásemos realizar la instalación nosotros mismos, deberíamos teclear:

```
$ rpm -ihv [paquete(s)]
```

Instalación desde código fuente

Descomprimos el paquete descargado utilizando el comando:

```
$ tar -xvzf squid-2.5.STABLE14.tar.gz
```

Se creará un directorio que contendrá los archivos con los que realizaremos la instalación. Situándonos dentro ejecutamos los siguientes comandos para llevar a cabo la instalación:

```
$ ./configure  
$ make  
$ make install
```

Este último comando instalará “Squid” con las opciones por defecto. Para alterar estas opciones podemos añadir al comando “./configure” los parámetros con la nueva configuración mediante el uso del prefijo –prefix.

Por ejemplo, para cambiar el directorio de instalación:

```
$ ./configure –prefix=/some/other/directory/squid
```

Una vez finalizado el proceso de instalación podemos lanzar el demonio, mediante el script que se encuentra en el directorio “/etc/init.d/squid” o mediante el ejecutable “/usr/local/squid/sbin/squid”, pero antes debemos crear los directorios de swap, para ello ejecutamos squid con el parámetro -z:

```
$ /usr/local/squid/sbin/squid -z
```

Una vez completa la instalación, podemos iniciar Squid y probarlo. Una manera de saber si funciona correctamente, es mirando sus mensajes de error. Una manera de obtenerlos

Práctica 6: Puesta en marcha de un servidor Proxy

es mediante:

```
$ /usr/local/squid/sbin/squid -NCd1
```

Si todo esta correcto se mostrará “Ready to serve requests”.

Desinstalación

La aplicación puede ser desinstalada de diversos modos, dependiendo de la instalación usada. En el caso de haber realizado una instalación binaria bastará con:

```
$ yum remove squid
```

Si hemos compilado los fuentes, debemos situarnos en el directorio donde se encuentran los ficheros de instalación y realizar:

```
$ make uninstall
```

Configuración de Squid

La configuración de Squid se lleva a cabo mediante el fichero “squid.conf”. En el caso de haber realizado una instalación binaria (como puede ser el nuestro) se encontrará bajo “/etc/squid”. En el caso de haber instalado desde las fuentes, posiblemente se encuentre en “<INSTALL_DIR>/etc/squid.conf”.

Gracias a haber instalado un paquete oficial, la aplicación se encontrará lo suficientemente probada como para fiarnos de ella y estará totalmente integrada en el sistema. Por ello disfrutaremos de la posibilidad de ejecutar el demonio como un servicio más bajo “/etc/init.d”.

Tamaño de disco de 50 MB

Se especifica en la sentencia `cache_dir` en la que se delimitan los valores relacionados con el directorio de la cache, tales como el directorio donde se encuentra, el número de directorios que pueden existir en el primer y segundo nivel y por último la capacidad del directorio de la cache.

```
cache_dir ufs /var/spool/squid 50 16 256
```

Se especifica que tenga una capacidad límite de 50 MB en el disco duro, 16 directorios de primer nivel y 256 en los directorios secundarios.

Realizar el logging de las solicitudes de los clientes

Para poder almacenar un login de las solicitudes de los clientes es necesario que hallamos realizado previamente una compilación con la opción “`--enable-useragent-log`”.

Práctica 6: Puesta en marcha de un servidor Proxy

```
$ .configure --enable-useragent-log  
$ make  
$ make install
```

E incorporar en squid.conf “*useragent_log*” seguido del nombre del fichero asociado al log.

```
useragent_log /var/log/squid/agentes
```

1 padre, 1 hermano solo para .com y 1 hermano para .es

La definición de una jerarquía de proxys se realiza con el parámetro “*cache_peer*”, con el que nos permite definir tanto relaciones padre-hijo, como hermano-hermano.

Con la inclusión de nuevos proxys podemos delegar el trabajo requerido por ciertas peticiones estos. Para ello declaramos con “*cache_peer_domain*” el dominio que le corresponde a los diversos proxys.

```
cache_peer proxy.rcanaria.es parent 3128 3130 proxy-only  
cache_peer 172.16.1.11 sibling 3128 3130 proxy-only  
cache_peer 172.16.1.8 sibling 3180 3130 proxy-only  
  
cache_peer_domain 172.16.1.11 .com !.es  
cache_peer_domain 172.16.1.8 .es !.com
```

Como se ve, hemos delegado las zonas .com y .es a nuestros hermanos. Para más señas estos son equipos de la red del laboratorio de redes.

Permitir solicitudes de clientes solo de la interfaz de la red interna.

Podemos especificar que sólo se atiendan peticiones desde una interfaz concreta, para ello ponemos la siguiente declaración de parámetro.

```
http_port 172.16.14.1:3128
```

Permitir solicitar conexiones con servidores y caches vecinos por la interfaz externa

Con las clausulas *udp_incoming_address* y *udp_outgoing_address* especificamos las interfaces por donde se recibirán paquetes de otras caches y por donde se enviarán las respuestas a otros proxys.

```
udp_incoming_address 172.16.1.14  
udp_outgoing_address 172.16.1.14
```

Prohibir como mínimo las siguientes direcciones: Playboy.com y Playgirl.com

Para permitir restricciones de acceso necesitamos establecer primero listas de posibles entradas al proxy. Con el tag: acl podemos definirlas. Estas tienen la siguiente estructura:

```
acl aclname acltype string1 ó
```

```
acl aclname acltype "file", teniendo en el fichero un elemento por línea.
```

Los tipos de acl que utilizaremos son los siguientes:

- `dstdomain` .- Indica una lista de nombres de dominio destino.
- `src` .- Indica direcciones ip fuentes.

```
Acl adultos dstdomain playboy.com playgirl.com
acl redlocal src 172.16.14.0/255.255.255.0
acl redexterna src 172.16.1.0/ 255.255.255.0
```

Así declaramos una lista llamada adultos, con las páginas web solicitadas. Una lista con las direcciones ip de la red interna y otra para la externa. Así podremos establecer restricciones de acceso las ips que se encuentran en las listas definidas.

Trás la creación de estas listas, se definen, para unas listas de protocolos y puertos ya establecidos, si las listas previamente creadas tienen o no acceso. Así para la lista de acceso a http establecemos sus restricciones y libertades con "http_access". Con él podemos permitir a los clientes http, es decir, los navegadores el acceso al puerto HTTP.

```
http_access deny adultos
http_access allow redlocal
http_access allow redexterna
```

Con estas opciones estamos bloqueando la lista de paginas adultas, y permitiendo el acceso de los equipos de la red local y de la red externa.

Postconfiguración

Una vez configurado el proxy, podemos arrancarlo utilizando

```
$ service squid start
```

o su equivalente

```
$ /etc/init.d/squid start
```

Si queremos que el servicio squid quede añadido entre los servicios en el arranque del sistema, deberemos ejecutar lo siguiente:

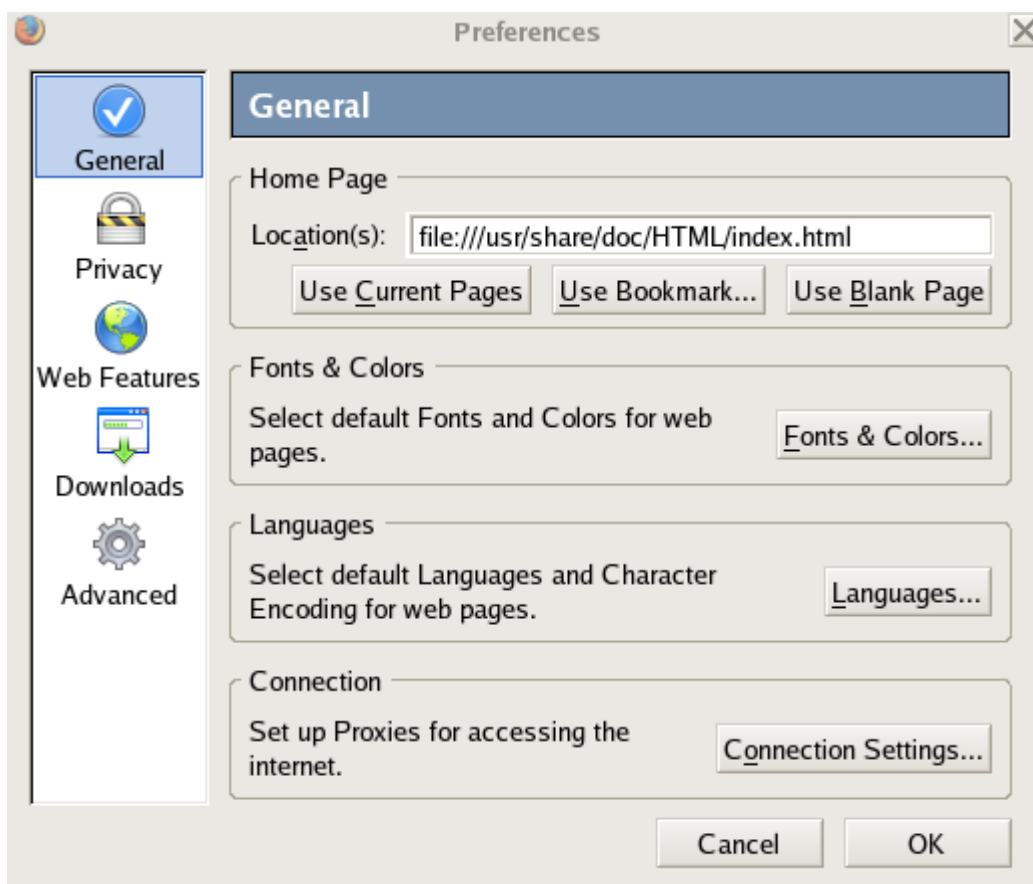
```
$ chkconfig squid on
```

Realización de pruebas

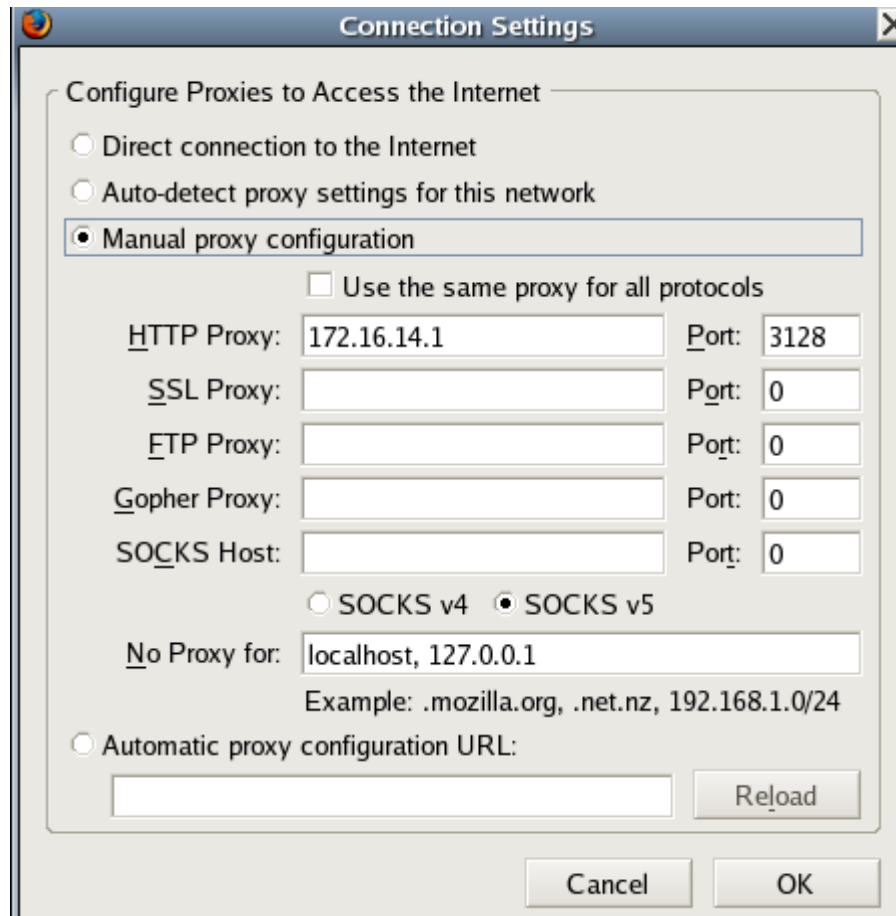
Realizaremos un conjunto de pruebas con el que poder comprobar el correcto funcionamiento del servidor proxy.

Configuraremos un navegador para que utilice nuestro proxy recién creado. De entre todos los posibles navegadores hemos escogido el Firefox, por ser un explorador muy conocido y de fácil acceso.

Para configurarlo, accedemos a Herramientas, Opciones si estamos en una versión windows, o a Editar, Preferencias si estamos trabajando en Linux. En todo caso la ventana resultante es similar en ambos casos:



Práctica 6: Puesta en marcha de un servidor Proxy



En la primera captura seleccionamos “Conection settings” para acceder a la segunda imagen. En donde podremos introducir la dirección ip de nuestro proxy (172.16.14.1) y el puerto 3128.

Una vez configurado el navegador comprobamos que se puede realizar una navegación fluida y sin complicación a excepción de las páginas prohibidas. Con la que obtendremos un bonito mensaje como el siguiente, en donde se nos deniega el acceso a dichas páginas.

ERROR

The requested URL could not be retrieved

While trying to retrieve the URL: <http://playboy.com/>

The following error was encountered:

- **Access Denied.**

Access control configuration prevents your request from being allowed at this time.
Please contact your service provider if you feel this is incorrect.

Your cache administrator is [root](#).

Generated Tue, 23 May 2006 18:30:09 GMT by localhost.localdomain (squid/2.5.STABLE11)

Para finalizar comprobamos la utilización de los proxys hermanos, ayudándonos de la herramienta ethereal en la cual podemos ver los paquetes que circulan por nuestra red.

Para ello navegamos por la página www.match.com página que será solicitada a 172.16.1.1

No.	Time	Source	Destination	Protocol	Info
557	12.25729	172.16.1.11	172.16.1.14	ICMP	Destination unreachable
560	12.25917	172.16.1.11	172.16.1.14	ICMP	Destination unreachable
564	12.26387	172.16.1.11	172.16.1.14	ICMP	Destination unreachable
597	12.63546	172.16.1.11	172.16.1.14	ICMP	Destination unreachable
326	6.964310	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 12
329	6.964914	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 12
331	6.980352	193.146.95.50	172.16.1.14	ICP	Opcode: ICP_MISS (3), Req Nr: 12
339	7.592406	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 13
340	7.592423	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 13
342	7.599011	193.146.95.50	172.16.1.14	ICP	Opcode: ICP_MISS (3), Req Nr: 13
379	9.698503	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 14
380	9.698521	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 14
382	9.704949	193.146.95.50	172.16.1.14	ICP	Opcode: ICP_MISS (3), Req Nr: 14
498	11.39337	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 15
499	11.39339	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 15
501	11.39997	193.146.95.50	172.16.1.14	ICP	Opcode: ICP_MISS (3), Req Nr: 15
532	11.90712	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 16
533	11.90714	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 16
535	11.91327	193.146.95.50	172.16.1.14	ICP	Opcode: ICP_MISS (3), Req Nr: 16
552	12.25561	172.16.1.14	172.16.1.11	ICP	Opcode: ICP_QUERY (1), Req Nr: 17
553	12.25563	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 17
555	12.25713	172.16.1.14	193.146.95.50	ICP	Opcode: ICP_QUERY (1), Req Nr: 18

Request Number: 12
Sender Host IP address 172.16.1.14
Payload
Requester Host Address 0.0.0.0
URL: http://match.com/

Práctica 6: Puesta en marcha de un servidor Proxy

Podemos ver como mandamos y recibimos paquetes al terminal 172.16.1.11. con la información de la página web visitada.

3. Referencias

[**WIKI**] Proyecto Wikipedia. Artículo Proxy

URL: <http://es.wikipedia.org/wiki/Proxy>

[**SQUID**] Oskar Pearson, “*Squid Documentation Project*”.

URL: <http://squid-docs.sourceforge.net>

[**BULMA**] “*Como configurar SQUID, el Proxy-Cache de Internet*”

URL: <http://bulma.net/body.phtml?nIdNoticia=441>

[**TLDP**] Kurt Seifried, José Antonio Revilla. “*Guía de Seguridad del Administrador de Linux*”. Proyecto Lucas, 1999.

URL: <http://es.tldp.org/Manuales-LuCAS/GSAL/gsal-19991128-htm/squid.htm>