Elvira Mifsud-k idatzia
Osteguna, 2008(e)ko maiatza(r)en 22-(e)an 09:51etan

There are no translations available.
Aprende en este artículo todo sobre este servidor web proxy-caché con licencia GPL...

Squid: servidor proxy-caché

1 Introducción

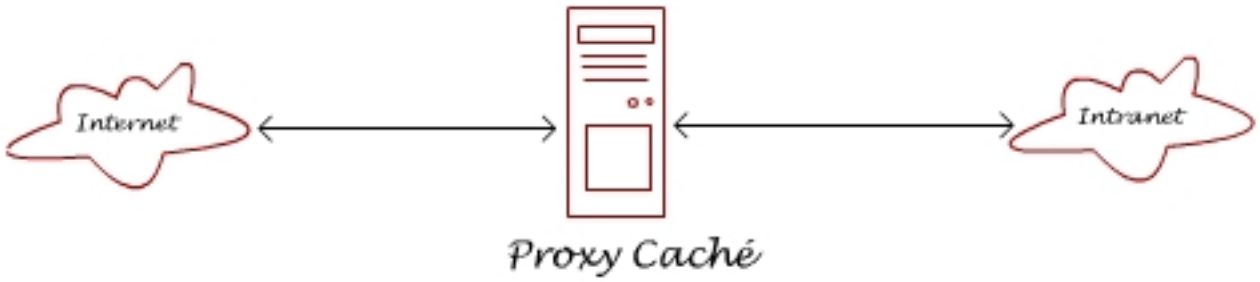
Squid es un servidor web proxy-caché con licencia GPL cuyo objetivo es funcionar como proxy de la red y también como zona caché para almacenar páginas web, entre otros.

La página oficial de SQUID es http://www.squid-cache.org/ .

En el artículo presentamos las principales características y funcionalidades de este potente servicio de amplia difusión en entornos GNU/Linux. Nosotros lo veremos en entorno Ubuntu.

2 Squid: servidor proxy-caché

¿Qué es un servidor proxy-caché?



Es un servidor situado entre la máquina del usuario y otra red (a menudo Internet) que actúa como protección separarando las dos redes y como zona caché para acelerar el acceso a páginas web o poder restringir el acceso a contenidos.

Es decir, la función de un servidor proxy es centralizar el tráfico de una red local hacia el exterior (Internet). Sólo el equipo que incorpora el servicio proxy debe disponer de conexión a Internet y el resto de equipos salen a través de él.

Como las peticiones hacía Internet de los equipos de la red local son interceptadas por el servidor proxy, éste puede realizar una tarea de filtrado de accesos, impidiendo aquellos destinos que estén expresamente prohibidos en los archivos de configuración del servicio. Squid no es un filtro de contenidos pero puede actuar como tal.

En el aula se suele utilizar este servicio ya que permite llevar un control sobre la actividad de la red hacia el exterior del aula. En este caso lo usual es que el equipo que hace la función de servidor proxy disponga de dos interfaces de red. Una de ellas es utilizada para atender a la red local y la otra proporciona la conexión con Internet. Las peticiones de páginas web que se realizan desde el aula son interceptadas por la interfaz interna y reenviadas a la interfaz externa si cumplen los requisitos establecidos desde el servicio proxy.

Hay que tener en cuenta que la mayoría de los servidores web permiten la configuración como proxy-caché (Apache, IIS,...), pero Squid sólo es un proxy y no puede servir páginas por sí mismo.

Cuando decimos que Squid también funciona como caché significa que está guardando copia de los datos obtenidos de otras peticiones y de esa forma acelera el acceso a estos datos si se producen peticiones similares. Sólo se accederá de nuevo a las páginas originales cuando se detecte que se han producido modificaciones, es decir los datos almacenados difieren de los datos en el servidor web de origen.

Normalmente no existe una sola caché, sino que se tienen varios servidores (en máquinas diferentes) relacionados entre sí mediante una estructura en árbol.

3 Funciones

Como resumen, las principales funciones de Squid son las siguientes:

-

Permite el acceso web a máquinas privadas (IP privada) que no están conectadas directamente a Internet.

-

Controla el acceso web aplicando reglas.

-

Registra el tráfico web desde la red local hacia el exterior.

-

Controla el contenido web visitado y descargado.

-

Controla la seguridad de la red local ante posibles ataques, intrusiones en el sistema, etc.

-

Funciona como una caché de páginas web. Es decir, almacena las páginas web visitadas por los usuarios y de esta manera las puede enviar a otros usuarios sin tener que acceder a Internet de nuevo.

-

Guarda en caché las peticiones DNS e implementa una caché para las conexiones fallidas.

-

Registra logs de todas las peticiones cursadas.

-

Soporta el protocolo ICP que permite integrar cachés que colaboran y permite crear jerarquías de cachés y el intercambio de datos.

Como consecuencia de estas funciones, la implantación de un servidor proxy-caché en una red proporciona las siguientes ventajas:

-

Reduce los tiempos de respuesta.

Si la página web que se solicita está en la caché del servidor, ésta se sirve sin necesidad de acceder de nuevo al servidor original, con lo cual se ahorra tiempo.

-

Disminuye el tráfico en la red y el consumo de ancho de banda.

Si la página web está almacenada en la caché del servidor, la petición no sale de la red local y no será necesario hacer uso de la línea exterior consiguiendo así un ahorro en la utilización del ancho de banda.

-

Cortafuegos.

Cuando se utiliza un servidor proxy-caché, éste comunica con el exterior, y puede funcionar como cortafuegos, lo cual aumentará la seguridad del usuario respecto a la información a la que se acceda.

-

Filtrado de servicios.

Es posible configurar el servidor proxy-caché dejando sólo disponibles aquellos servicios (HTTP, FTP,...) que se consideren necesarios, impidiendo la utilización del resto.

4 Instalación

El paquete incluido en Edubuntu Feisty Fawn es squid-2.6.5 Estable. Para instalar el servicio utilizar la herramienta Synaptic (**Sistema -> Administración -> Gestor de paquetes Synaptic**
)
:

Elvira Mifsud-k idatzia
Osteguna, 2008(e)ko maiatza(r)en 22-(e)an 09:51etan

Gestor de Paquetes Synaptic

Archivo Editar Paquete Configuración Ayuda

Recargar Marcar todas las actualizaciones Aplicar Propiedades Buscar

Todo
Administración del sistema
Administración del sistema
Administración del sistema
Bibliotecas
Bibliotecas (multiverse)
Bibliotecas (universe)
Bibliotecas - Antiguo
Bibliotecas - Antiguo (unive
Bibliotecas - Desarrollo
Bibliotecas - Desarrollo (m
Bibliotecas - Desarrollo (ur
Ciencia
Ciencia (multiverse)
Ciencia (universe)
Comunicación
Comunicación (multiverse)

E	Paquete	Versión instalada	Última versión	De
	squeak-image		3.8-6665-0ubuntu1	sq
	squeak-sources		3-3ubuntu3	sq
	squeak-vm		3.7.7-5ubuntu6	sq
	squid	2.6.5-4ubuntu2.1	2.6.5-4ubuntu2.1	Int
	squid3		3.0.PRE5-5ubuntu0.	A f
	squid3-cgi		3.0.PRE5-5ubuntu0.	A f
	squid3-client		3.0.PRE5-5ubuntu0.	A f
	squid3-common		3.0.PRE5-5ubuntu0.	A f
	squid-cgi		2.6.5-4ubuntu2.1	Sq
	squidclient		2.6.5-4ubuntu2.1	Co
	squid-common	2.6.5-4ubuntu2.1	2.6.5-4ubuntu2.1	Int
	squidguard		1.2.0-8.1	filt
	squid-prefetch		1.1-2	Sir
	squidtaild		2.1a6-5	Sq

Internet Object Cache (WWW proxy cache)

This is the Squid Internet Object Cache developed by the National Laboratory for Applied Networking Research (NLANR) and Internet volunteers. This software is freely available for anyone to use. The Squid home page is http://www.squid-cache.org/

Secciones
Estado
Origen
Filtros
Resultados de búsqueda

21525 paquetes listados, 1671 instalados, 0 rotos. 0 para instalar/actualizar, 0 para eliminar

Aplicaciones Lugares Sistema 9 °C mié 5

Directorio del ejecutable.

/usr/sbin/

/var/run/

Archivo con el PID del proceso.

/var/log/squid/

Directorio de logs. Relacionado con la directiva access_log.

/var/spool/squid/

Directorio caché. Relacionado con cache_dir.

/etc/squid/

Archivos de configuración.

/usr/lib/squid/

Complementos.

/etc/rc.d/

Scripts de arranque.

/usr/share/doc/squid/

Documentación.

5 Configuración básica

En el servidor habrá que:

-

Determinar el espacio en disco dedicado al servidor proxy-caché.

-

Configurar el propio servidor a nivel de puerto, directorios, usuarios, etc.

-

Arrancar el servicio

En el cliente para que utilice el servidor proxy habrá que realizar una:

-

Configuración manual (servidor, protocolos y puerto)

-

Configuración automática utilizando un archivo proxy.pac

La utilización del proxy-caché requiere configurar el navegador para indicar que la conexión a Internet no es directa, sino a través del proxy. Suele estar en el menú principal *Ver.Preferencias* .

El archivo de configuración de SQUID es /**etc**/**squid**/**squid.conf**. Una configuración básica debe incluir los siguientes parámetros:

-

cache_effective_user / **group**

Por problemas de seguridad es preferible que Squid y sus procesos asociados se ejecuten como usuario y grupo *proxy*. Este usuario deberá ser el propietario del directorio caché y el directorio de logs.

cache_effective_user proxy

cache_effective_group proxy

¡Ojo! no debe haber ningún blanco en la primera columna.

-

http_port

Por defecto Squid atiende peticiones por el puerto 3128 pero también se puede utilizar el 8080. Se puede cambiar el puerto e incluso Squid puede escuchar por varios puertos a la vez.

http_port 3128

Si se quiere aumentar la seguridad, puede vincularse el servicio a una IP que sólo se pueda acceder desde la red local. Considerando que el servidor utilizado posee una IP 10.0.2.254, puede hacerse lo siguiente:

http_port 10.0.2.254:3128

http_port 10.0.2.254:8080

-

dns_nameservers

Indica las direcciones IP de los servidores DNS donde el servidor realizará las consultas de nombres.

dns_nameservers 10.0.2.254

En el ejemplo el propio servidor está resolviendo nombres.

-

cache_peer

Squid permite crear jerarquías de cachés. Puede haber proxys-cachés padres y hermanos. Si establecemos una jerarquía padre-hijo (**parent**), el padre debe proporcionar el objeto pedido tanto si está en la caché como si no lo está.

Para utilizar un proxy-caché padre éste tiene que darnos permiso para utilizar su línea externa y en el archivo de configuración se deberá indicar:

Sintaxis:

cache_peer servidor tipo http_port icp_port [opciones]

Ejemplo:

cache_peer nombre/IP parent 8080 0 no-query no-digest default

Si establecemos una jerarquía entre hermanos (**sibling**), el proxy hermano sólo sirve el objeto si lo tiene en caché, nunca irá a Internet a buscarlo. Esto sólo es útil para redes con proxys en el mismo nivel.

8080 -> puerto HTTP del servidor remoto.

0 -> indica el puerto ICP del servidor remoto. Se utiliza cuando hay varios padres, para averiguar cuál de ellos tiene el objeto pedido. Si hay un solo padre se coloca un 0.

no-query -> desactiva la petición de paquetes ICP al padre.

no-digest -> no es necesario con un solo padre.

default -> Squid utilizará este servidor para todas las peticiones.

-

cache_mem

Establece la cantidad de memoria RAM dedicada para almacenar los bloques más solicitados. Si la cantidad de memoria necesaria para este tipo de objetos es mayor que la especificada en el parámetro cache_mem, Squid tomará la que le haga falta.

Es una buena norma asignar N/3, siendo N la RAM del equipo.

cache_mem 96 MB

-

cache_dir

Especifica el tamaño de la caché en disco duro. Por defecto 100MB. Se pueden especificar el nº de subdirectorios y el nº de niveles posibles dentro de cada subdirectorio.

cache_dir ufs /var/spool/squid 100 16 256

Esta línea indica una caché en disco de 100MB, con 16 subdirectorios de primer nivel que se pueden utilizar y 256 subdirectorios de segundo nivel.

-

acl Lista de control del acceso

Permite:

-

Proteger al proxy de conexiones externas, evitando que se conecten clientes desconocidos que podrían saturar la conexión con el exterior.

-

Proteger a los clientes de accesos a puertos peligrosos actuando como cortafuegos contra posibles ataques desde la web.

-

Establecer una jerarquía de cachés.

-

Establecer una red como conjunto de trabajo o máquinas individuales.

-

A continuación, a cada ACL se le hace corresponder una Regla de Control de Acceso (**http_access**).

La sintaxis es: **acl [nombre_lista] src [componentes_lista]**

src -> hace referencia al origen, es decir, a la IP de un cliente. Existen mas opciones que iremos viendo.

[componentes_lista] -> se pueden indicar valores IP de redes, con la máscara de red, o archivos cuyo contenido sean las IPs.

Ejemplo:

acl aula src 10.0.2.0/255.255.255.0

acl aula /etc/squid/mi_red

Donde el archivo /**etc**/**squid**/**mi_red** tendría las IPs de las máquinas del aula (una por línea).

Se pueden crear ACLs para impedir el acceso, desde el proxy, a ciertas páginas web:

acl web_denegadas dstdomain .chicas.com .sex.com

El parámetro acl también se utiliza para establecer las conexiones permitidas a través del proxy. Podemos limitar los puertos a los que puede conectarse para atender las peticiones mediante la lista SSL_ports . Para ello se utiliza el método CONNECT, que es una puerta para la conexión a otros servidores desde el proxy.

acl SSL_ports port 443 563

acl CONNECT method CONNECT

http_access deny !SSL_ports

http_access deny CONNECT !SSL_ports

En este caso se está permitiendo la conexión, mediante CONNECT, a los puertos SSL 443 y 563.

Mas abajo se explica el parámetro **http_access**.

Se puede restringir la salida a Internet durante un período de tiempo. Por ejemplo, a las máquinas especificadas se les deja conectar desde las 9h de la mañana hasta las 17h de la tarde:

acl IP_permitidas src 10.0.2.5 10.0.2.8 10.0.2.11

acl horario time de lunes a viernes **9:00-17:00** a 5

acl host2 src 10.0.2.2

acl mañana time 9:00-14:00

Las Listas de control de acceso sirven también para especificar URLs que contienen un texto en concreto y a las cuales no se quiere permitir el acceso. Para ello se crea un archivo **/etc/squid/denegar.txt**
con un texto por línea y que deberá estar contenido en el nombre de la URL.

En el archivo de configuración **/etc/squid/squid.conf** se añade:

acl url_denegar url_regex /etc/squid/denegar.txt

http_access deny url_denegar

Si lo que se quiere es hacer el filtrado por algún contenido del path no por el nombre del host, hay que utilizar la entrada **urlpath_regex** y pasarle como argumento un archivo con las palabras a filtrar.

-

http_access Regla de control de acceso

La regla de control de acceso define qué navegadores u otros proxys podrán acceder o no a Squid para hacer peticiones HTTP. Se aplica sobre la Lista de control de acceso ACL.

La sintaxis es: **http_access [deny** / **allow] [lista_control_acceso]**

Ejemplo: para la ACL **acl aula /etc/squid/mi_red** le correspondería

http_access allow aula

en la que permitimos el acceso a Squid a los equipos del aula.

Si dentro del aula hay equipos a los que no se quiere dar acceso se puede utilizar el carácter ! para excluirlos. Para ello creamos otra ACL con las máquinas no permitidas y escribimos:

http_access allow aula !no_permitidos

Ejemplo completo:

Tenemos una red 10.0.2.0/255.255.255.0 de la que sólo ciertas IPs van a poder acceder a Squid. Creamos con estas IPs un archivo permitidos y su ACL correspondiente:

acl permitidos src /etc/squid/permitidos

Una configuración básica bajo estas condiciones sería:

acl todo src 0.0.0.0/0.0.0.0

acl permitidos src /etc/squid/permitidos

acl web_denegadas dstdomain .chicas.com .sex.com

acl horario time MTWHF 9:00-17:00

acl mañana time 9:00-14:00

acl url_denegar url_regex /etc/squid/denegar.txt

http_access allow permitidos horario

http_access deny permitidos

http_access deny web_denegadas

http_access deny url_denegar

http_access deny todo

En este ejemplo:

Se crea una ACL 'todo' para cualquier IP.

Se crea una ACL para las IPs permitidas.

Se crea una ACL para seleccionar ciertas webs destino.

Se establecen horarios de conexión.

Se seleccionan URLs que cumplen ciertos patrones.

Las líneas http_access permiten o deniegan la conexión en función de la ACL sobre la que

actuan.

La lectura se hace de arriba hacia abajo y se detiene en la primera coincidencia para permitir o denegar la conexión.

Si una línea http_access tiene más de un argumento se evalúan con un AND y con la siguiente línea http_access con un OR.

-

cache_mgr

Este parámetro especifica la dirección de correo del administrador a la que se enviará un mensaje en el caso de que le ocurra algo a la caché.

cache_mgr **webmaster**

-

httpd_accel

Las peticiones de Internet de los usuarios se almacenan en la caché de Squid. Si otros usuarios solicita la misma petición y el elemento en caché no ha sufrido ninguna modificación, Squid muestra el de la caché y no vuelve a descargarlo de Internet con lo cual se aumenta la rapidez en la navegación.

Por ejemplo, para un proxy convencional[1], las opciones para proxy acelerado son:

httpd_accel_host virtual

httpd_accel_port 0

httpd_accel_with_proxy o□n

Donde:

httpd_accel_host indica el nombre del servidor web que se quiere acelerar. Si se escribe 'virtual' se está indicando que se quiere acelerar mas de un servidor.

httpd_accel_port indica el puerto donde escucha el servidor que se quiere acelerar.

httpd_accel_with_proxy permite al proxy trabajar como proxy y como acelerador al mismo tiempo ya que con la primera opción (**httpd_accel_host**) Squid deja de actuar como proxy.

6 Ubicación del proxy

La ubicación del cortafuegos respecto al proxy-caché tiene mucha importancia de cara a su configuración.

¿Qué posibilidades tenemos?

-

Proxy-caché dentro de la zona protegida

-

Proxy-caché fuera de la zona protegida

-

Proxy-caché en la DMZ[2]

Veamos cada una de las situaciones:

1.

Proxy-caché dentro de la zona protegida (proxy interno).

Se asume que el proxy es un host en el que se confía y además queda protegido por el propio cortafuegos.

Si el proxy comparte la caché con otros servidores habrá que configurarla como 'parent' ya que los cortafuegos suelen impedir el tráfico ICP por ser de tipo UDP.

2.

Proxy-caché fuera de la zona protegida (proxy externo).

Se asume que el proxy es un host en el que no se confía, no queda protegido por el propio cortafuegos y está expuesto a posibles ataques.

Un motivo para ubicar al servidor proxy de esta forma es para que pueda comunicarse por ICP

con otros proxys.

En este caso los navegadores deben configurarse para no utilizar el proxy para acceder a los servidores internos.

En este caso también la caché no está protegida por el cortafuegos, por lo que su configuración es mas delicada:

1.
1.

Sólo debe aceptar peticiones HTTP desde el cortafuegos para que usuarios no autorizados no puedan utilizar el proxy.

2.

Al no confiar en dicha máquina debe colocarse en un puerto de un switch separado de otras máquinas.

3. Proxy-caché en la DMZ

Si la instalación dispone de red perimetral el proxy debe ser ubicado en la DMZ junto con los otros servidores de la organización.

Los navegadores de los clientes sí que podrán utilizar siempre el proxy.

El servidor caché acepta las peticiones desde el cortafuegos.

Deben permitirse conexiones entre la caché y los servidores web (80,443) y puede habilitarse

el tráfico UDP para permitir comunicaciones por ICP.

Pensando en un aula de centro educativo lo usual es utilizar el propio servidor de aula como servidor proxy-caché. No es lo mas seguro pero si lo mas habitual.

7 Archivos de logs

Squid genera los siguientes archivos de log:

/var/log/squid/access.log almacena las peticiones que se le hacen al proxy. De esa forma se puede conocer cuántos usuarios utilizan el proxy, cuáles son las páginas más visitadas,... Mantiene una entrada por cada consulta HTTP con la IP del cliente, la URL pedida, etc. Existen aplicaciones que obtienen informes de este archivo de logs con el análisis de los datos almacenados. Por ejemplo Squid-Log_Analyzer.

Archivos/Directorios	Descripción
/var/log/squid/access.log	Archivo registro de peticiones al proxy.
/var/log/squid/cache.log	

Archivo de accesos a la caché, errores, mensajes de inicio,... Relacionado con cache_log.

/var/log/squid/store.log

Histórico de la caché propiamente. Estado de los objetos almacenados en la caché. Páginas que se añ

Su información no es muy importante y se puede desactivar añadiendo la línea:

cache_store_log none

/var/spool/squid/

Directorio caché. Relacionado con cache_dir.

Se puede indicar en **/etc/squid/squid.conf** los paths en los que serán creados estos archivos:

cache_dir ufs /var/spool/squid 100 16 256

cache_access_log /var/log/squid/access.log

cache_log /var/log/squid/cache.log

cache_store_log /var/log/squid/store.log

De ellas, la primera indica donde quedará almacenada la caché y, el resto, los paths de los archivos de log.

Las páginas de error pueden mostrarse en diferentes idiomas. Para obtenerlas en español hay que establecer un enlace de la forma:

 # ln -s /usr/share/squid/errors/Spanish /etc/squid/errors

En función del número de peticiones que reciba el proxy este archivo puede llegar a crecer a 1MB por minuto. Puede, por tanto, colapsar la partición.

Como este archivo aumenta de tamaño muy rápido, conviene hacer que se reinicie cada día. Si queremos guardar los registros de los 7 últimos días en el archivo **/etc/squid/squid.conf** establecemos una rotación de 7 con el parámetro
logfile_rotate
:

logfile_rotate 7

 # squid -k rotate

8 Arranque del servicio SQUID

Antes de arrancar Squid, y sólo la primera vez, habrá que ejecutar la orden siguiente para crear los directorios de la caché donde se guardarán las páginas.

```
# squid -z
```

Para activar o desactivar el servicio:

```
# /etc/rc.d/init.d/squid {start|stop|reload|force-reload
```

Si no se dispone de DNS hay que iniciar el servicio con la opción -D.

Una buena práctica consiste en incluir todas las modificaciones al archivo **/etc/squid/squid.conf** en el archivo **/etc/squid/local-squid.conf**.
Este archivo local irá con un **Include** en el archivo general de configuración.

Después de hacer modificaciones en el archivo de configuración hay que relanzar el servicio. También se pueden activar las modificaciones, sin necesidad de parar el servicio, mediante la orden:

```
# squid -k reconfigure
```

Si queremos automatizar el arranque de Squid utilizamos la orden **rcconf**:

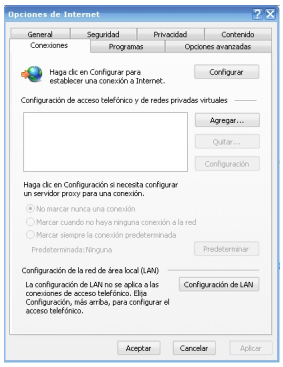
 # rcconf squid

9 Configuración del navegador web

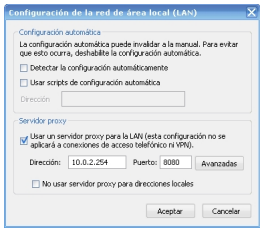
La mayoría de los navegadores permiten su configuración para trabajar a través de un proxy y dicha configuración es muy similar en todos ellos.

Estos navegadores deben ser configurados con el puerto e IP del servidor proxy.

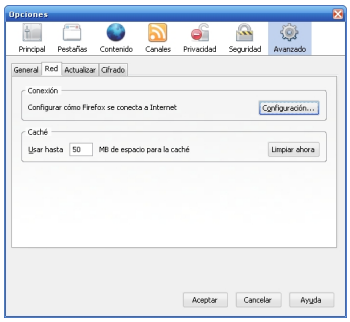
Internet Explorer: Herramientas -> Opciones de Internet



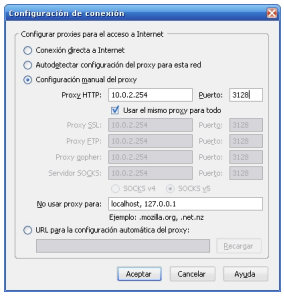
Conexiones -> Configuración de LAN (en servidor proxy introducir la IP del servidor Squid y el puerto).



Firefox: Herramientas -> Opciones



Configuración -> Configuración manual del proxy



10 Archivos de autoconfiguración

Es posible también realizar la configuración de los clientes utilizando un archivo de configuración automática y de esa forma centralizar los cambios realizados. Este archivo se llama **proxy.pac** y es un script creado por el administrador del sistema.

Este script establece el modo con el que los navegadores web acceden a Internet y se guarda en una dirección a la que tienen acceso, en modo lectura, todos los clientes web. Normalmente se sitúa en un servidor web, como por ejemplo, Apache. Dicha dirección se tendrá que indicar al configurar el navegador.

La utilización de **proxy.pac** tiene la ventaja de que el administrador del sistema puede realizar cambios de forma transparente al usuario y sin necesidad de introducir modificaciones en los

navegadores web. Por ejemplo, un cambio en la IP del proxy.

Incluimos el contenido de un archivo **proxy.pac** genérico para que sobre él se comprenda su estructura y funcionamiento.

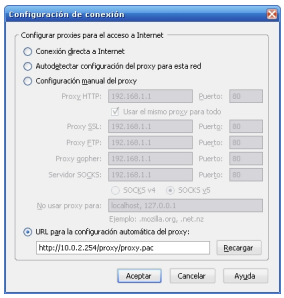
```
function FindProxyForURL(url,host)

{ if (dnsDomainIs(host, "aula"))

return "DIRECT";

else if (isInNet(host, "10.0.2.0", "255.255.255.0"))

return "DIRECT";

else if (isInNet(host, "127.0.0.1", "255.255.255.255"))

return "DIRECT";

else            return "PROXY proxy:3128"; }
```

El archivo contiene una función **FindProxyURL** que lleva como parámetros la URL a la que quiere acceder el navegador y la máquina que contiene el recurso.

Los bloques if-else indican que hay contenidos que están en el dominio **'aula'** o en máquinas de la propia red local (
10.0.2.0
). El acceso, por tanto, es directo (DIRECT), sin intermediarios.

El resto de contenidos que no entran dentro de la estructura -if- se direccionan a través del proxy SQUID indicando PROXY máquina:puerto.

Este contenido se guarda en el archivo /var/www/proxy/proxy.pac y se le dan permisos 644 (rw - r - - r - -) y se deja disponible mediante un host virtual
http://proxy
o en la dirección
http://10.0.2.254/proxy/proxy.pac
.



11 Configuración de SQUID para el acceso a Internet por autenticación

Por defecto Squid se configura de forma que el usuario tiene acceso sin ningún tipo de autenticación.

Una forma de ampliar la seguridad con Squid es utilizar los usuarios Ubuntu validados para controlar la conexión a Internet. Para ello vamos a suponer que existe un usuario '*profesor*' con su correspondeinte contraseña, que va a disponer de acceso total a Internet.

La lista ACL correspondiente a incluir será:

acl profesor_autorizado ident profesor

que indica que el usuario de identidad '*profesor*' dispone de una regla de control de acceso específica.

La regla de control de acceso correspondiente será:

http_access allow profesor_autorizado

Para controlar el acceso a Internet desde Squid se puede también utilizar el método de las autorizaciones (**proxy_auth**) que requiere de un procedimiento de validación de los usuarios para la utilización del Squid.

Un sistema de autenticación permite controlar quien va a poder conectarse a Internet independientemente de la máquina de la red local desde la que se haga la conexión. Como mecanismo de autenticación se utilizará **ncsa_auth** que viene incluido en Squid. Hay otros mecanismos de autenticación válidos como LDAP, SMB, PAM.

Para ello:

-

Creamos el archivo **squid_passwd** en el que se almacenarán los logins de los usuarios junto con las contraseñas cifradas:

 # touch /etc/squid/squid_passwd

y almacena en el archivo **squid_passwd** parejas de valores **nombre_usuario:contraseña**.

-

Cedemos la propiedad del archivo al usuario '*proxy*':

 # chown proxy:proxy /etc/squid/squid_passwd

- Sólo el usuario '*proxy*' podrá leer y escribir en este archivo:

 # chmod 600 /etc/squid/squid_passwd

- Damos de alta a los usuarios:

 # htpasswd /etc/squid/squid_passwd usuario1

Introducimos la contraseña y confirmamos.

Estas cuentas son independientes de las del sistema o de cualquier otro servicio y sólo sirven para las conexiones a Internet a través del proxy Squid.

A nivel de archivo de configuración /**etc**/**squid**/**squid.conf** hay que incorporar una serie de parámetros necesarios:

-

Indicar el programa de autenticación que va a ser utilizado (en nuestro caso **ncsa_auth**). Añadir la línea:

autenticate_program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd

-

Se establece como lista de control de acceso la obligación de autenticarse ante Squid:

acl control proxy_auth REQUIRED

Ahora, siguiendo con el ejemplo de la red local con sólo ciertas IPs permitidas, la configuración de Squid queda de la siguiente forma:

acl todo src 0.0.0.0/0.0.0.0

acl localhost src 127.0.0.1/255.255.255.255

acl aula src /etc/squid/permitidos

acl control proxy_auth REQUIRED

http_access allow localhost

http_access allow aula control

http_access deny todo

Por último, hay que relanzar el servicio:

 # /etc/rc.d/init.d/squid restart

12 Configuración de un proxy transparente

Hasta aquí hemos explicado el objetivo, configuración y funcionamiento de un proxy normal. Hay ocasiones en las que no interesa que los usuarios sepan que están saliendo a Internet a través de un proxy, o se quiere forzar la utilización del proxy sin tener que estar configurando todos los navegadores disponibles en la red.

Bien, pues esta es la misión del proxy transparente: interceptar todas las peticiones web de los clientes de forma transparente, de forma que los clientes creen estar saliendo directamente a Internet.

Sin embargo tiene el inconveniente de que no se puede hacer una autenticación del usuario por contraseña.

El proxy transparente utiliza el puerto 80 y el redireccionamiento de peticiones, por lo que no hay necesidad de modificar la configuración de los navegadores web para utilizar el servidor proxy, será suficiente utilizar como puerta de enlace la IP del servidor. Como el servidor Apache utiliza el 80 será necesario configurar el servidor web para que utilice otro puerto de los

disponibles.

Hay que tener el sistema de reenvío (forwarding) activado, que es el que permite a la máquina actuar como router. Ejecutar[3]:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Las líneas siguientes de **/etc/squid/squid.conf** hay que dejarlas de la forma soguiente para que Squid reconozca el tráfico:

```
http_port 3128
```

```
httpd_accel_host virtual
```

```
httpd_accel_port 80
```

```
httpd_accel_with_proxy on
```

Para **proxy transparente** conviene dejar Squid en el puerto por defecto 3128.

Para el reenvío de las peticiones hay que incorporar una regla IPTABLES mediante la línea:

```
/sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Cualquier petición que vaya al puerto 80 es redirigida a Squid, y no hay que modificar cada uno de los navegadores de los clientes. Aunque sólo sirve para peticiones HTTP.

13 Conclusión

A lo largo del artículo se ha explicado la configuración básica del servicio Squid, así como los parámetros mas importantes de su configuración, valores posibles y funcionalidades. Como se habrá podido observar las posibilidades de Squid son muy grandes y no se ha tocado nada del tema de monitorización del servicio e incluso su utilización como filtro de contenidos. Existen potentes herramientas para llevar a cabo estás funciones, como son Sarg o Squidguard.

Hemos utilizado la edición directa de los archivos de configuración y la ejecución de las órdenes relacionadas con su funcionamiento. Es una forma de conocer mejor los detalles de la configuración del servicio. Eso no quita la posibilidad de utilizar algún tipo de herramienta gráfica para su configuración, como puede ser Webmin y que puede ser tema de un artículo específico para esta herramienta.

Comentarios

1) Para un proxy transparente escuchando en el puerto 80 habría que modificar: httpd_accel_port 80 y añadir la línea: httpd_accel_uses_host_header o-n.

2)DMZ: En seguridad informática, una zona desmilitarizada (DMZ) o red perimetral es una red local (una subred) que se ubica entre la red interna de una organización y una red externa, generalmente Internet. (www.ibercom.com)

3)Solo para esta sesión. Si se quiere de forma permanente hay que incluirlo en algún script rc de arranque.