Proxy-caché squid

1. Introducción

Un proxy de conexión a Internet es un servidor que hace de intermediario entre los PCs de la red y el router de conexión a Internet, de forma que cuando un usuario quiere acceder a Internet, su PC realiza la petición al servidor Proxy y es el Proxy quien realmente accede a Internet.

Posteriormente, el Proxy enviará los datos al PC del usuario para que los muestre en su pantalla. El PC del usuario no tendrá conexión directa con el router, sino que accederá a Internet por medio del proxy.

2. Ventajas de disponer de un proxy

- Los PCs de los usuarios **no tienen acceso al router**, todas las comunicaciones exteriores pasarán por el Proxy, lo que nos permitirá tener las comunicaciones bajo control. Podemos permitir o denegar el acceso web, ftp, email, messenger, p2p, etc...
- Las páginas se **cachean** en la memoria temporal del proxy lo cual acelera la descarga cuando varios usuarios acceden a las mismas páginas a la vez. Esta circunstancia se da mucho en los centros educativos cuando el profesor está explicando un tema y todos los alumnos acceden a la vez a la misma página.
- Es fácil crear una lista de **urls prohibidas** a las que el proxy denegará el acceso.
- Permite crear una lista de **palabras prohibidas en url**. El proxy denegará el acceso cuando se introduzcan en formularios de búsqueda o en la barra de direcciones.
- Se puede permitir o denegar el acceso a **subredes **o a PCs concretos. Si diseñamos la red de forma que cada aula del centro tenga un rango determinado, por ejemplo 10.0.X.Y donde X es el número de aula e Y el número de PC, sería posible permitir o denegar la conexión a Internet aula por aula.
- El proxy guarda informes de todas las conexiones que hacen los usuarios. Al principio puede ser interesante ver a qué páginas de contenido inadecuado acceden nuestros alumnos, para agregarlas a la lista de urls prohibidas.
- Los PCs de nuestra red están más seguros de ataques externos ya que el proxy hace de barrera cortafuegos.

3. Inconvenientes de la utilización de un Proxy

No todo son ventajas, también hay algún inconveniente en la utilización de un Proxy:

- Para que las aplicaciones accedan a Internet a través del proxy, es **necesario configurar** cada aplicación: navegador web, cliente ftp, cliente de correo, etc...
- Todas las comunicaciones con el exterior pasarán por el servidor. Si el proxy **falla**, la red se quedará sin conexión a Internet. Para subsanar lo más rápidamente posible el problema ante un fallo del Proxy, será conveniente disponer de un proxy de repuesto.
- El proxy requiere **mantenimiento**. Para que todo funcione, es necesario que exista un administrador de la red que se encargue de actualizar, revisar, mantener y reparar el proxy cuando deje de funcionar.

4. introducción a los servicios

La mayoría de servidores funcionan como servicios. Un **servicio** es un proceso (programa en ejecución) que generalmente:

- Se ejecuta en segundo plano
- No tiene normalmente ninguna interfaz gráfica
- No tiene interacción con el usuario
- Suele configurarse para arrancar al iniciarse el equipo
- Suelen registrar su actividad en ficheros de **log**

Linux ofrece multitud de servicios o servidores, estos pueden iniciar o arrancar junto con la carga del sistema o pueden después ser puestos a funcionar cuando se requieran (es lo mejor). Parte esencial de la administración de sistemas Linux es continuamente trabajar con los servicios que este proporciona, cosa que es bastante sencilla.

En windows se llaman servicios, pero en Linux se les conoce como **daemons**, y su nombre suele terminar en d, como por ejemplo el daemon http**d**.

5. Scripts de servicios

Dentro de esta carpeta ubicada en /etc o en /etc/rc.d dependiendo de la distribución, se encuentran una serie de scripts que permiten iniciar/detener la gran mayoría de los servicios/servidores que estén instalados en el equipo. Estos scripts están programados de tal manera que la mayoría reconoce los siguientes argumentos:

- start
- stop
- restart
- status

Los argumentos son autodescriptivos, y tienen permisos de ejecucción, siendo root.

6. Instalación del Proxy squid

Linux dispone del proxy **squid**. Se trata de una aplicación de gran éxito que se lleva utilizando muchos años y dispone de cientos de posibilidades para personalizar su funcionamiento a nuestras necesidades.

Squid es un **proxy** de almacenamiento en caché para la Web que admite **HTTP, HTTPS, FTP** y más. Entre otras, tiene las siguientes funciones:

- Reduce el ancho de banda y mejora los tiempos de respuesta al almacenar en caché
- Permite reutilizar las páginas web solicitadas con frecuencia.
- Permite crear controles de acceso

Para instalar la última versión de squid, podemos hacerlo con **apt** desde una consola de root:

Instalación del servidor Proxy squid:

```
sudo apt install squid
```

De esta forma instalaríamos los programas necesarios para disponer de un completo servidor Proxy en nuestra red. Tan solo será necesario configurarlo y ponerlo en marcha.

7. Arranque y parada del proxy squid

El servicio squid, al igual que todos los servicios, dispone de **scripts** de arranque y parada en la carpeta /etc/init.d. Debemos ejecutarlos desde una consola de root.

Arrancar o reiniciar el servidor squid:

```
sudo /etc/init.d/squid restart \
sudo systemctl start squid
```

// Parar el servidor squid \

```
sudo /etc/init.d/squid stop \
sudo systemctl stop squid
```

// Recargar configuración del servidor squid
sudo /etc/init.d/squid reload

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Trucos > Arranque automático de servicios al iniciar el sistema.

8. Comprobar el estado del servicio

Se utiliza el comando **systemctl status squid. **Como se puede comprobar, el servicio squid está activo. Podemos visualizar también las últimas entradas del log.

Podemos comprobar como el servicio está activo, y está utilizando:

- El archivo de configuración /etc/squid/squid.conf
- El archivo de log /var/log/squid/access.log

 alt_text

Podemos también visualizar si el puerto está activo o no, mediante el siguiente comando. Tener en cuenta que solo veremos los procesos arrancados por el usuario, por lo que para verlos todos tendremos que ejecutarlo como root.

sudo netstat -tnap | grep squid

 alt_text

Podemos comprobar que está escuchando en el puerto 3128, en este caso. El puerto se puede modificar.

9. Configuración básica del proxy squid

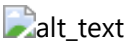
El archivo de configuración del proxy es el archivo **/etc/squid/squid.conf**. Si observamos dicho archivo, veremos que es un archivo muy extenso en el que hay cientos de parámetros que podemos establecer, pero para una utilización básica, son unos pocos los parámetros que debemos configurar.

Antes de editar el archivo de configuración, es recomendable hacer una copia del archivo original y protegerlo contra escritura para tener el archivo original como referencia, y reutilizarlo si fuese necesario. Se puede hacer con los siguientes comandos:

```
sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.original
sudo chmod a-w /etc/squid/squid.conf.original
```

Deberíamos modificar el fichero para que escuche en IPv4 y en IPv6, cambiando la variable http_port a 0.0.0.0:3128.

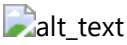
Reiniciamos: systemctl restart squid. Comprobamos que el cambio ha surtido efecto:

 alt_text

10. Comprobar que el puerto esté abierto

Tenemos que comprobar si el firewall está habilitado. Si no lo está, no tendremos problema. Si lo tenemos habilitado, tendremos que añadir las reglas necesarias, de dos formas:

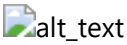
1. Agregamos los puertos manualmente
2. Permitimos la aplicación Squid, que habrá creado un perfil para ufw al instalarse

 alt_text

11. filtrar comentarios del archivo de configuración

El archivo de configuración de squid es muy largo y contiene muchos comentarios. Para ver solo las líneas de configuración sin los comandos:

```
grep -E -v '^(#|;|$|[ ]*#)' /etc/squid/squid.conf
```

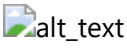
 alt_text

12. Comprobación del archivo de configuración

Para comprobar que el archivo de configuración está bien (a ser posible como root)

```
squid -k parse
```

Obtendremos una salida como la siguiente:

 alt_text

13. Crear listas de control y accesos

En esta sección estableceremos los permisos de acceso, es decir, quien puede navegar y quién no. Lo primero que tendremos que hacer es crear listas de control de acceso (Access Control List - ACL) y luego dar permisos a dichas listas.

Listas de control ACL

Una** lista de control de acceso (acl) **se crea utilizando la palabra acl seguido del nombre que queramos dar a la lista y seguido de una condición que cumplirán los miembros de la lista.

Entre las condiciones más utilizadas destacamos: src (IPs o URLs origen), dst (IPs o URLs destino), port (puertos) y proto (protocolos). Ejemplos:

Definir una ACL para toda la red

Si en mi red local utilizo el direccionamiento 10.0.0.0/8, puedo crear una lista para definir a toda mi red:

```
//acl para definir toda mi red \
acl todos src 10.0.0.0/8
```

Definir ACLs para cada aula

Si en mi red local utilizo el direccionamiento 10.0.X.0/24, para el aula X, puedo crear una lista para cada aula:

```
//Una acl para cada aula \
acl aula1 src 10.0.1.0/24 \
acl aula2 src 10.0.2.0/24 \
acl aula3 src 10.0.3.0/24 \
acl aula4 src 10.0.4.0/24 \
acl aula5 src 10.0.5.0/24
```

Accesos HTTP_ACCESS

Luego tendría que dar permiso a las listas. Para ello se utiliza la palabra clave **http_access** seguido del permiso **allow** (permitir) o **deny** (denegar) y seguido del nombre de la lista.

Si quiero dar permiso a toda mi red para que navegue por Internet:

//Permiso para que navegue toda mi red
http_access allow todos

Si quiero dar permiso a las aulas 1, 2 y 3 para que navegue por Internet, pero no quiero que naveguen las aulas 4 y 5:

```
//Permiso para que naveguen las aulas 1, 2 y 3 y no naveguen las aulas 4 y 5 \
http_access allow aula1 \
```

```
http_access allow aula2 \
http_access allow aula3 \
http_access deny aula4 \
http_access deny aula5
```

Por defecto, squid viene configurado para actuar como **caché **de acceso a Internet, pero no tiene creadas listas de control de acceso. Si configuramos el navegador de Internet de los PCs cliente para que utilicen el Proxy, veremos que tenemos denegado el acceso al Proxy.

Para empezar a disfrutar del Proxy, tendremos que crear una lista de control de acceso con el rango de nuestra red y darle permiso. Si en nuestra red utilizamos el rango 10.0.0.0/8, deberíamos añadir en /etc/squid/squid.conf:

```
//Permiso para que navegue toda mi red. \
acl todos src 10.0.0.0/8 \
http_access allow todos
```

Cuando creamos ACLs, podemos sustituir el rango de IPs por el nombre de un** archivo externo**, y de esa manera podemos indicar el en archivo externo el rango o los rangos de IPs a los que queremos referirnos, sin necesidad de estar continuamente modificando el archivo squid.conf.

Más adelante veremos un ejemplo cómo tener un archivo externo con las URLs prohibidas a las que no podrán navegar nuestros alumnos.

Limitaciones remporales

Se pueden crear ACLs para especificar franjas horarias, como por ejemplo:

```
acl HorarioTrabajo time 08:00-17:00
```

Podríamos negar la conexión fuera del horario de trabajo de la siguiente manera:

```
http_access deny !HorarioTrabajo
```

Diferencias entre las sentencias acl y http_access

Todas las conjunciones de la misma regla se tienen que cumplir (AND) juntas. Es decir, se tienen que dar todas las condiciones para que se apliquen.

- acl = acl1 Y acl2 Y acl3 Y ...

A continuación, todas las reglas de http_access se aplican de tal manera que por lo menos se cumpla una de ellas.

- http_access = regla1 O regla2 O ...

Ejemplo:

- acl A src 1.2.3.4
- acl B src 5.6.7.8
- http_access allow A B (permitir equipos que cumplan A y B)

En este caso, no se pueden cumplir las condiciones A y B al mismo tiempo, por lo que no funciona.

Si quisiéramos hacerlo, lo deberíamos configurar así.

- acl A src 1.2.3.4 5.6.7.8 (Equipos con alguna de las IP anteriores)
- http_access allow A (permitir equipos que cumplan las reglas de A)

14. opciones de red

Configuración normal

En esta sección estableceremos con el parámetro http_port, el puerto en el que escucha el Proxy. Lo mejor es dejar el valor por defecto que es el puerto 3128:

```
//Configurar squid en el puerto 3128 \
http_proxy 3128
```

Configuración como proxy transparente (no lo haremos)

Squid puede trabajar en modo **transparente**. La ventaja de configurar squid en dicho modo de trabajo, es que no sería necesario configurar el navegador de los PCs clientes para trabajar con el proxy, sino que simplemente configuramos la puerta de enlace del PC cliente con la IP del servidor proxy.

Posteriormente tendremos que configurar el cortafuegos del servidor para que redirija las peticiones al puerto 80 hacia el puerto 3128 y así las reciba squid. Si deseamos poner el Proxy en modo transparente, deberemos indicarlo después del puerto. En tal caso, el parámetro http_port quedaría así:

//Configurar squid en el puerto 3128, en modo transparente \

```
http_proxy 3128 transparent
```

//Redirigir las peticiones al puerto 80 hacia el puerto 3128. Ejecutar como root: \

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

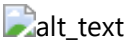
El inconveniente de trabajar en modo transparente es que no sirve para el protocolo HTTPS. \

15. Configuración del navegador de los PCs clientes, para que utilicen el Proxy

Supongamos que nuestro servidor Proxy tiene la IP 192.168.1.239 y el servidor **squid** está escuchando en el puerto 3128 que es el puerto que utiliza por defecto. Con estos dos datos, la IP y el puerto, ya podemos configurar el navegador de Internet de los PC clientes.

Mozilla Firefox

Para que Firefox utilice nuestro Proxy en sus conexiones, debemos ir a Herramientas > Opciones > Avanzado > Red y en el apartado Conexión, hacer clic en el botón Configuración. En la ventana que aparece, debemos configurar la IP y el puerto de nuestro servidor Proxy:



A partir de este momento, Firefox enviará a nuestro Proxy cualquier consulta web que realice, y será nuestro Proxy quien realizará la conexión en caso necesario.

Internet Explorer

Para indicar a Internet Explorer que debe utilizar un Proxy para realizar conexiones, debemos ir a Herramientas > Opciones de Internet > Conexiones > Configuración de LAN y activar la casilla 'Usar un servidor proxy para la LAN'. En la casilla 'Dirección' pondremos la IP de nuestro Proxy y el 'Puerto' el puerto, tal y como se muestra en la siguiente ventana:

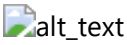
16. Configuración del Proxy a través de un archivo de configuración (no lo haremos)

Para no tener que recordar la dirección del proxy y facilitar la tarea a la hora de configurar el proxy en los PCs clientes, existe la posibilidad de crear un archivo de configuración automática del proxy. Dicho archivo indicará al navegador, en función de la url a la que quiera conectarse, si debe hacerlo directamente o debe hacerlo a través del proxy.

En un direccionamiento como el que tenemos en nuestro centro, cuando accedemos a nuestra red 10.0.0.0/8 o a la dirección de localhost 127.0.0.1, la conexión debe ser directa, en cambio, cuando accedemos a cualquier otra dirección, deberá ser a través de proxy.

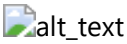
Archivo de configuración automática del proxy

```
//Archivo de configuración automática del proxy \
//Archivo /var/www/proxy.pac \
function FindProxyForURL(url,host){ \
if (isInNet(host, "10.0.0.0", "255.0.0.0")) \
return "DIRECT"; \
else if (isInNet(host, "127.0.0.1", "255.255.255.255")) \
return "DIRECT"; \
else return "PROXY 192.168.1.239:3128"; \
}
```

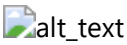


17. Primera conexión al proxy

Lo más probable es que a la primera no nos deje:



Al menos esto significa que estamos pasando a través del firewall sin problemas. Miramos el archivo de configuración:



Cambiamos el http_access y le metemos allow all

18. Permitir o denegar el acceso desde ciertos rangos de IPs

Tal y como se ha comentado anteriormente, con squid es sencillo permitir o denegar el acceso a Internet por rangos de IPs. Si tenemos nuestra red diseñada de forma que cada aula utiliza un rango concreto, podremos permitir o denegar el acceso a un aula de forma sencilla.

Para no tener que tocar el archivo squid.conf, lo mejor es crear una ACL que cargue las aulas desde un archivo externo. Podemos crear con un editor de texto el archivo /etc/squid/aulas-prohibidas.txt en el que indicaremos los rangos de IPs que no queremos que naveguen. Por ejemplo, si no queremos que naveguen las aulas 2 y 3, el contenido del archivo /etc/squid/aulas-prohibidas.txt deberá ser:

```
//Archivo /etc/squid/aulas-prohibidas.txt \
10.0.2.0/24 \
10.0.3.0/24
```

Después tendremos que editar squid.conf para crear una acl que cargue los rangos desde el archivo /etc/squid/daulas-prohibidas.txt y deniegue el acceso a dichos rangos.

Archivo externo para indicar las aulas a las que no las permitimos navegar \

Editar squid.conf e introducir estas dos líneas: \

```
acl aulas-prohibidas src "/etc/squid/aulas-prohibidas.txt" \
http_access deny aulas-prohibidas
```

Por último, tan solo tenemos que recargar la configuración de squid para que entre en funcionamiento la nueva configuración:

//Recargar la configuración de squid \

```
sudo /etc/init.d/squid reload
```

19. Indicar URL prohibidas

Podemos crear ACL para listas de direcciones.

```
acl urls-prohibidas dst .facebook.com .twitter.com \
http_access deny urls-prohibidas
```

Las direcciones que comienzan por un punto (.) hacen referencia a cualquier URL que acabe en esta cadena.

Ejemplo: `.facebook.com` afecta a `static.facebook.com`

Igualmente podemos crear una ACL para indicar las URL prohibidas desde un archivo externo:

```
//Archivo externo para indicar las urls prohibidas \
//Editar squid.conf e introducir estas dos líneas: \
acl urls-prohibidas dst "/etc/squid/urls-prohibidas.txt" \
http_access deny urls-prohibidas
```

Si no queremos que nuestros alumnos accedan a www.sex.com ni a www.misvecinitas.com, el contenido del archivo /etc/squid/urls-prohibidas.txt debería ser:

```
//Archivo /etc/squid/urls-prohibidas.txt \
www.twitter.com \
www.facebook.com
```

La filosofía sería:

1. Denegar las aulas prohibidas
2. Denegar las urls prohibidas
3. Luego permitir todo lo demás

Resumiendo, nuestro archivo squid.conf será como el original con las siguientes modificaciones, justo después de la línea # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS que podríamos traducir como: Inserte sus propias reglas para permitir acceso a sus clientes:

//Resumen de modificaciones en squid.conf \

```
acl aulas-prohibidas src "/etc/squid/aulas-prohibidas.txt" \
http_access deny aulas-prohibidas \
acl urls-prohibidas dst "/etc/squid/urls-prohibidas.txt" \
http_access deny urls-prohibidas \
http_access allow all
```

Así, editando los archivos /etc/squid/aulas-prohibidas.txt y /etc/squid/urls-prohibidas.txt y recargando la configuración de squid ejecutando /etc/init.d/squid reload, podemos reconfigurar squid sin necesidad de tocar el archivo de configuración squid.conf.

20. SCRIPTS

El inconveniente es que cada vez que queremos permitir o denegar el acceso a Internet a un aula, tenemos que andar editando el archivo aulas-prohibidas.txt lo que puede resultar un poco engorroso.

Podemos crear dos scripts de unix que hagan el trabajo por nosotros y solamente tengamos que ejecutar los scripts indicando el número de aula que queremos prohibir o permitir:

Script prohibir aula

Nombre del script: **prohibir-aula.sh** \

```
#/bin/bash \
#  \
# Script para prohibir la navegación de un aula \
# Se creará el rango del aula en /etc/squid/aulas-prohibidas.txt \
# Indicar el número de aula al ejecutar el script \
 \
if [ $# -ne 1 ]; then \
echo "Es necesario introducir el número de aula a prohibir" \
exit -1 \
fi \
echo Prohibir navegar aula $1, subred 10.0.$1.0/24 \
echo 10.0.$1.0/24 >> /etc/squid/aulas-prohibidas.txt \
/etc/init.d/squid reload \
echo subredes denegadas: \
cat /etc/squid/aulas-prohibidas.txt
```

Script permitir aula

```
//Nombre del script: **permitir-aula.sh** \
#/bin/bash \
#  \
# Script para permitir la navegación de un aula \
# Se eliminará el rango del aula de /etc/squid/aulas-prohibidas.txt \
# Indicar el número de aula al ejecutar el script \
 \
if [ $# -ne 1 ]; then \
echo "Es necesario introducir el numero de aula" \
exit -1 \
fi \
subred=10.0.$1.0/24 \
echo Permitir navegar aula $1, subred $subred \
patron=`echo /10.0.$1.0/d` \
cat /etc/squid/aulas-prohibidas.txt | sed -e $patron > /tmp/temp.txt  \
```

```
cat /tmp/temp.txt > /etc/squid/aulas-prohibidas.txt \
/etc/init.d/squid reload \
echo Subredes denegadas: \
cat /etc/squid/aulas-prohibidas.txt
```

Si deseamos que el aula 1 no navegue, deberíamos ejecutar: prohibir-aula 1. Si luego deseamos permitir que el aula 1 navegue, tendríamos que ejecutar:

```
permitir-aula 1
```

Aún con los scripts prohibir-aula.sh y permitir-aula.sh, sigue siendo engorroso realizar cambios ya que el profesor tendría que iniciar sesión en el servidor por ssh y lanzar el script. Lo mejor será crear una página en PHP con botones de comando, donde con un simple clic podamos ejecutar los scripts cómodamente desde el navegador.

21. Análisis de conexiones

Una de las funcionalidades principales que nos ofrece squid es que registra todos los accesos a Internet. Cada vez que un PCs accede a Internet, squid registrará en el archivo /var/log/squid/access.log la fecha y hora, el PC y la URL a la que ha accedido.

//Archivo de registro de squid
/var/log/squid/access.log

Para mostrar el contenido del log completo:

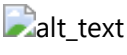
```
cat /var/log/squid/access.log
```

Para mostrar el contenido del log en tiempo real:

```
tail -f /var/log/squid/access.log
```

Para mostrar las últimas 10 líneas

```
tail -n 10 /var/log/squid/access.log
```



Ejemplo de clase

Listas:

```
acl smx2a src 192.168.1.0/24
acl smx2b src 192.168.2.0/24
acl sociales dst .facebook.com .twitter.com .instagram.com
acl periodicos dst .elmundo.es .elpais.com
acl horarioclase time 08:00-15:00
```

Reglas

```
http_access **allow smx2b sociales // se permite a las IP 192.168.1.0/24 acceder a
redes sociales
http_access **allow periodicos horarioclase // se permite a todo el mundo acceder
a los periódicos en horario de clase
http_access **deny sociales // se prohibe a todos los que no cumplan las reglas
anteriores acceso a redes sociales
```