



Linux: info y utilidades del squid web proxy

Indice

- [1. Software de Proxy](#)
- [2. Cómo configurar Squid: Servidor Proxy.](#)
- [3. Cómo configurar Squid: Restricción de acceso a sitios Web.](#)
- [4. Cómo configurar Squid: Acceso por Autenticación.](#)
- [5. Bibliografía](#)

1. Software de Proxy

Existe una variedad de paquetes de software de proxy para Linux. Algunos son a nivel de aplicación (como SQUID) y otros son a nivel de sesión (como SOCKS). Squid es un proxy a nivel de aplicación para HTTP, HTTPS y FTP. También puede ejecutar peticiones DNS bastante más rápido de lo que puede hacerlo la mayoría del software cliente. SQUID es ideal para acelerar el acceso a www, y para controlar el acceso a sitios web (utilizando paquetes como squidGuard).

Squid es el servidor Proxy más popular y extendido entre los sistemas operativos basados sobre UNIX®. Es muy confiable, robusto y versátil. Al ser *software libre*, además de estar disponible el código fuente, está libre del pago de costosas licencias por uso o con restricción a un uso con determinado número de usuarios.

Entre otras cosas, Squid puede hacer Proxy y cache con los protocolos HTTP, FTP, GOPHER y WAIS, Proxy de SSL, cache transparente, WWCP, aceleración HTTP, cache de consultas DNS y más.

2. Cómo configurar Squid: Servidor Proxy.

Software requerido.

Para poder llevar la cabo los procedimientos descritos en este manual y documentos relacionados, usted necesitará tener instalado al menos lo siguiente:

- squid-2.4.STABLE1
- iptables-1.2.4
- kernel-2.4.9

Tómese en consideración que, de ser posible, se debe utilizar la versión estable más reciente de todo el software que vaya a instalar al realizar los procedimientos descritos en este manual, a fin de contar con los parches de seguridad necesarios. Ninguna versión de Squid anterior a la 2.4.STABLE1 se considera como apropiada debido a fallas de seguridad de gran importancia, y ningún administrador *competente* utilizaría una versión inferior a la 2.4.STABLE1. Por favor visite el sitio Web de su distribución predilecta para estar al tanto de cualquier aviso de actualizaciones de seguridad. Ejemplo: para Red Hat Linux 7.1 y 7.2 hay paquetería de actualización en los siguientes enlaces:

- <ftp://updates.redhat.com/7.1/en/os/i386/>, si posee alguna distribución basada sobre Red Hat(TM) Linux 7.1
- <ftp://updates.redhat.com/7.2/en/os/i386/>, si posee alguna distribución basada sobre Red Hat(TM) Linux 7.2

Instalación del software necesario.

Regularmente Squid no se instala de manera predeterminada a menos que especifique o contrario durante la instalación del sistema operativo, sin embargo viene incluido en casi todas las distribuciones actuales. El procedimiento de instalación es exactamente el mismo que con cualquier otro software:

```
mount /mnt/cdrom/  
  
rpm -Uvh /mnt/cdrom/*/RPMS/squid-*.i386.rpm  
  
eject
```

Iptables se utilizará para un guión de Enmascaramiento de IP. Se instala por defecto en todas las distribuciones actuales que utilicen kernel-2.4.

Es importante tener actualizado el kernel por diversas cuestiones de seguridad. No es recomendable utilizar versiones del kernel anteriores a la 2.4.9. En el manual "Cómo actualizar el Kernel a partir de paquetes RPM®" se describe a detalle lo necesario.

Antes de continuar

Tenga en cuenta que este manual ha sido comprobado varias veces y ha funcionado en todos los casos y si algo no funciona solo significa que usted no lo leyó a detalle y no siguió correctamente las indicaciones.

Evite dejar espacios vacíos en lugares indebidos. El siguiente es un ejemplo de como no debe descomentarse un parámetro.

```
Mal  
  
# Opción incorrectamente descomentada  
  
http_access 3128
```

El siguiente es un ejemplo de como si debe descomentarse un parámetro.

```
Bien
# Opción correctamente descomentada
http_access 3128
```

Configuración básica.

Squid utiliza el fichero de configuración localizado en `/etc/squid/squid.conf`, y podrá trabajar sobre este utilizando su editor de texto preferido. Existen un gran número de parámetros, de los cuales recomendamos configurar los siguientes:

- `http_port`
- `cache_mem`
- `ftp_user`
- `ftp_passive`
- `cache_dir`
- Al menos una *Lista de Control de Acceso*
- Al menos una *Regla de Control de Acceso*
- `cache_mgr`
- `httpd_accel_host`
- `httpd_accel_port`
- `httpd_accel_with_proxy`

Parámetro `http_port`: ¿Que puerto utilizar para Squid?

Squid por defecto utilizará el puerto 3128 para atender peticiones, sin embargo se puede especificar que lo haga en cualquier otro puerto o bien que lo haga en varios puertos a la vez.

En el caso de un *Proxy Transparente*, regularmente se utilizará el puerto 80 y se valdrá del re-direccionamiento de peticiones de modo tal que no habrá necesidad alguna de modificar la configuración de los navegadores Web para utilizar el servidor Proxy. bastará con utilizar como puerta de enlace al servidor. Es importante recordar que los servidores Web, como Apache, también utilizan dicho puerto, por lo que será necesario reconfigurar el servidor Web para utiliza otro puerto disponible, o bien desinstalar o deshabilitar el servidor Web.

Hoy en día ya no es del todo práctico el utilizar un *Proxy Transparente*, a menos que se trate de un servicio de *Café Internet* u oficina pequeña, siendo que uno de los principales problemas con los que lidian los administradores es el mal uso y/o abuso del acceso a Internet por parte del personal. Es por esto que puede resultar más conveniente configurar un servidor Proxy con restricciones por contraseña, lo cual no puede hacerse con un *Proxy Transparente*, debido a que se requiere un diálogo de nombre de usuario y contraseña.

Regularmente algunos programas utilizados comúnmente por los usuarios suelen traer por defecto el puerto 8080 -*servicio de cacheo WWW*- para utilizarse al configurar que servidor proxy utilizar. Si queremos aprovechar esto en nuestro favor y ahorrarnos el tener que dar explicaciones innecesarias al usuario, podemos especificar que Squid escuche peticiones en dicho puerto también. Siendo así localice la sección de definición de `http_port`, y especifique:

```
#
# You may specify multiple socket addresses on multiple lines.
#
# Default: http_port 3128
http_port 3128
http_port 8080
```

Parámetro `cache_mem`

El parámetro `cache_mem` establece la cantidad ideal de memoria para lo siguiente:

- Objetos en tránsito.
- Objetos Hot.
- Objetos negativamente almacenados en el caché.

Los datos de estos objetos se almacenan en bloques de 4 Kb. El parámetro `cache_mem` especifica un límite máximo en el tamaño total de bloques acomodados, donde los objetos en tránsito tiene mayor prioridad. Sin embargo los objetos *Hot* y aquellos negativamente almacenados en el caché podrán utilizar la memoria no utilizada hasta que esta sea requerida. De ser necesario, si un objeto en tránsito es mayor a la cantidad de memoria especificada, Squid excederá lo que sea necesario para satisfacer la petición.

Por defecto se establecen 8 MB. Puede especificarse una cantidad mayor si así se considera necesario, dependiendo esto de los hábitos de los usuarios o necesidades establecidas por el administrador.

Si se posee un servidor con al menos 128 MB de RAM, establezca 16 MB como valor para este parámetro:

```
cache_mem 16 MB
```

Parámetro `cache_dir`: ¿Cuanto desea almacenar de Internet en el disco duro?

Este parámetro se utiliza para establecer que tamaño se desea que tenga el cache en el disco duro para Squid. Para entender esto un poco mejor, responda a esta pregunta: ¿Cuanto desea almacenar de Internet en el disco duro? Por defecto Squid utilizará un cache de 100 MB, de modo tal que encontrará la siguiente línea:

```
cache_dir ufs /var/spool/squid 100 16 256
```

Se puede incrementar el tamaño del cache hasta donde lo desee el administrador. Mientras más grande el cache, más objetos de almacenarán en éste y por lo tanto se utilizará menos el ancho de banda. La siguiente línea establece un cache de 700 MB:

```
cache_dir ufs /var/spool/squid 700 16 256
```

Los números *16* y *256* significan que el directorio del cache contendrá 16 subdirectorios con 256 niveles cada uno. No modifique esto números, no hay necesidad de hacerlo.

Es muy importante considerar que si se especifica un determinado tamaño de cache y este excede al espacio real disponible en el disco duro, Squid se bloqueará inevitablemente. Sea cauteloso con el tamaño de cache especificado.

Parámetro `ftp_user`

Al acceder a un servidor FTP de manera anónima, por defecto Squid enviará como contraseña Squid @. Si se desea que el acceso anónimo a los servidores FTP sea más informativo, o bien si se desea acceder a servidores FTP que validan la autenticidad de la dirección de correo especificada como contraseña, puede especificarse la dirección de correo electrónico que uno considere pertinente.

```
ftp_user proxy@su-dominio.net
```

Parámetro `ftp_passive`

Si se tiene un muro contrafuegos que no permite acceder a servidores FTP más que de modo pasivo, debe habilitarse `ftp_passive` con el valor *on*.

```
ftp_passive on
```

Controles de acceso.

Es necesario establecer *Listas de Control de Acceso* que definan una red o bien ciertas maquinas en particular. A cada lista se le asignará una *Regla de Control de Acceso* que permitirá o denegará el acceso a Squid. Procedamos a entender como definir unas y otras.

Listas de control de acceso.

Regularmente una lista de control de acceso se establece siguiendo la siguiente sintaxis:

```
acl [nombre de la lista] src [lo que compone a la lista]
```

Si uno desea establecer una lista de control de acceso que defina sin mayor trabajo adicional a toda la red local definiendo la IP que corresponde a la red y la máscara de la sub-red. Por ejemplo, si se tienen una red donde las máquinas tienen direcciones IP 192.168.1.n con máscara de sub-red 255.255.255.0, podemos utilizar lo siguiente:

```
acl miredlocal src 192.168.1.0/255.255.255.0
```

También puede definirse una *Lista de Control de Acceso* invocando un fichero localizado en cualquier parte del disco duro, y en el cual se en cuenta una lista de direcciones IP. Ejemplo:

```
acl permitidos "/etc/squid/permitidos"
```

El fichero `/etc/squid/permitidos` contendría algo como siguiente:

```
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.15
192.168.1.16
192.168.1.20
192.168.1.40
```

Lo anterior estaría definiendo que la *Lista de Control de Acceso* denominada *permitidos* estaría compuesta por las direcciones IP incluidas en el fichero `/etc/squid/permitidos`.

Reglas de Control de Acceso

Estas definen si se permite o no el acceso a Squid. Se aplican a las *Listas de Control de Acceso*. Deben colocarse en la sección de reglas de control de acceso definidas por el administrador, es decir, a partir de donde se localiza la siguiente leyenda:

```
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
```

La sintaxis básica es la siguiente:

```
http_access [deny o allow] [lista de control de acceso]
```

En el siguiente ejemplo consideramos una regla que establece acceso permitido a Squid a la *Lista de Control de Acceso* denominada *permitidos*:

```
http_access allow permitidos
```

También pueden definirse reglas valiéndose de la expresión `!`, la cual significa *excepción*. Pueden definirse, por ejemplo, dos listas de control de acceso, una denominada *lista1* y otra denominada *lista2*, en la misma regla de control de acceso, en donde se asigna una expresión a una de estas. La siguiente establece que se permite el acceso a Squid a lo que comprenda *lista1* excepto aquello que comprenda *lista2*:

```
http_access allow lista1 !lista2
```

Este tipo de reglas son útiles cuando se tiene un gran grupo de IP dentro de un rango de red al que se debe permitir acceso, y otro grupo dentro de la misma red al que se debe denegar el acceso.

Aplicando Listas y Reglas de control de acceso.

Una vez comprendido el funcionamiento de la Listas y las Regla de Control de Acceso, procederemos a determinar cuales utilizar para nuestra configuración.

Caso 1

Considerando como ejemplo que se dispone de una red `192.168.1.0/255.255.255.0`, si se desea definir toda la red local, utilizaremos la siguiente línea en la sección de *Listas de Control de Acceso*:

```
acl totalared src 192.168.1.0/255.255.255.0
```

Habiendo hecho lo anterior, la sección de listas de control de acceso debe quedar más o menos del siguiente modo:

```
Listas de Control de Acceso: definición de una red local completa
```

```
#  
  
# Recommended minimum configuration:  
acl all src 0.0.0.0/0.0.0.0  
acl manager proto cache_object  
acl localhost src 127.0.0.1/255.255.255.255  
acl todalared src 192.168.1.0/255.255.255.0
```

A continuación procedemos a aplicar la regla de control de acceso:

```
http_access allow todalared
```

Habiendo hecho lo anterior, la zona de reglas de control de acceso debería quedar más o menos de este modo:

```
Reglas de control de acceso: Acceso a una Lista de Control de Acceso.
```

```
#  
  
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS  
  
#  
http_access allow localhost  
http_access allow todalared  
  
http_access deny all
```

La regla `http_access allow todalared` permite el acceso a Squid a la *Lista de Control de Acceso* denominada *todalared*, la cual está conformada por 192.168.1.0/255.255.255.0. Esto significa que cualquier máquina desde 192.168.1.1 hasta 192.168.1.254 podrá acceder a Squid.

Caso 2

Si solo se desea permitir el acceso a Squid a ciertas direcciones IP de la red local, deberemos crear un fichero que contenga dicha lista. Genere el fichero `/etc/squid/lista`, dentro del cual se incluirán solo aquellas direcciones IP que desea confirmen la Lista de Control de acceso. Ejemplo:

```
192.168.1.1  
192.168.1.2  
192.168.1.3  
192.168.1.15  
192.168.1.16  
192.168.1.20  
192.168.1.40
```

Denominaremos a esta lista de control de acceso como *redlocal*:

```
acl redlocal src "/etc/squid/lista"
```

Habiendo hecho lo anterior, la sección de listas de control de acceso debe quedar más o menos del siguiente modo:

```
Listas de Control de Acceso: definición de una red local completa
```

```
#  
# Recommended minimum configuration:  
acl all src 0.0.0.0/0.0.0.0  
acl manager proto cache_object  
acl localhost src 127.0.0.1/255.255.255.255  
acl redlocal src "/etc/squid/lista"
```

A continuación procedemos a aplicar la regla de control de acceso:

```
http_access allow redlocal
```

Habiendo hecho lo anterior, la zona de reglas de control de acceso debería quedar más o menos de este modo:

```
Reglas de control de acceso: Acceso a una Lista de Control de Acceso.
```

```
#  
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS  
#  
http_access allow localhost  
http_access allow redlocal  
http_access deny all
```

La regla `http_access allow redlocal` permite el acceso a Squid a la *Lista de Control de Acceso* denominada *redlocal*, la cual está conformada por las direcciones IP especificadas en el fichero `/etc/squid/lista`. esto significa que cualquier máquina no incluida en `/etc/squid/lista` no tendrá acceso a Squid.

Parámetro `cache_mgr`.

Por defecto, si algo ocurre con el Cache, como por ejemplo que muera el proceso, se enviará un mensaje de aviso a la cuenta *webmaster* del servidor. Puede especificarse una distinta si acaso se considera conveniente.

```
cache_mgr joseperez@midominio.net
```

Cache con aceleración.

Cuando un usuario hace petición hacia un objeto en Internet, este es almacenado en el cache de Squid. Si otro usuario hace petición hacia el mismo objeto, y este no ha sufrido modificación alguna desde que lo accedió el usuario anterior, Squid mostrará el que ya se encuentra en el cache en lugar de volver a descargarlo desde Internet.

Esta función permite navegar rápidamente cuando los objetos ya están en el cache de Squid y además optimiza enormemente la utilización del ancho de banda.

En la sección *HTTPD-ACCELERATOR OPTIONS* deben habilitarse los siguientes parámetros:

```
Proxy Acelerado: Opciones para Proxy Convencional.
```

```
httpd_accel_host virtual
httpd_accel_port 0
httpd_accel_with_proxy on
```

Si se trata de un Proxy transparente -Squid *escuchando peticiones en el puerto 80*-, debe hacerse con las siguientes opciones:

Proxy Acelerado: Opciones para Proxy Transparente.

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Por defecto el parámetro `httpd_accel_with_proxy` viene con el valor *off*, es importante no olvidar cambiar este valor por *on*.

Estableciendo el idioma por defecto.

Squid incluye traducción a distintos idiomas de las distintas páginas de error e informativas que son desplegadas en un momento dado. Dichas traducciones se pueden encontrar en `/usr/lib/squid/errors/`. Para poder hacer uso de las páginas de error traducidas al español, es necesario cambiar un enlace simbólico localizado en `/etc/squid/errors` para que apunte hacia `/usr/lib/squid/errors/Spanish` en lugar de hacerlo hacia `/usr/lib/squid/errors/English`. Elimine primero el enlace simbólico actual:

```
rm -f /etc/squid/errors
```

Coloque un nuevo enlace simbólico apuntando hacia `/usr/lib/squid/errors/Spanish`.

```
ln -s /usr/lib/squid/errors/Spanish /etc/squid/errors
```

Iniciando, reiniciando y añadiendo el servicio al arranque del sistema.

Una vez terminada la configuración, ejecute el siguiente comando para iniciar por primera vez Squid:

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

Si necesita reiniciar para probar cambios hechos en la configuración, ejecute lo siguiente:

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

Si desea que Squid inicie de manera automática la próxima vez que inicie el sistema, ejecute lo siguiente:

```
/sbin/chkconfig --level 345 squid on
```

Lo anterior habilitará a Squid en los niveles de corrida 3, 4 y 5.

Nota para los novatos: Usted NO tiene que editar nada en `/etc/rc.d/rc.local` o `/etc/inittab` para que Squid -*así como cualquier otro servicio*- inicie en el arranque del sistema. Mientras usted sea novato, por favor, olvide que existen esos ficheros y exclame una fuerte amenaza y alejese de quien le indique que desde ahí debe arrancar servicios.

Ajustes para el muro contrafireos o guión de Enmascaramiento de IP.

A continuación comentaremos algunos ajustes que pueden añadirse o editarse en el guión de el muro contrafuegos, como el generado por herramientas como Firestarer, o bien un simple guión de Enmascaramiento de IP.

Sugerimos utilizar Firestarer debido a que permite configurar tanto el enmascaramiento de IP como el muro contrafuegos y la importancia que tiene la presencia de éste último en un servidor que sirve como puerta de enlace para la red local.

Iptables en lugar de ipchains.

Desde el kernel 2.4, GNU/Linux utiliza Netfilter, el cual se configura a través de *iptables*. La sintaxis cambia con respecto a *ipchains*, y a fin de permitir a los administradores darse tiempo de adaptarse, distribuciones como Red Hat (TM) incluyeron soporte para *ipchains* a manera de *aplicación de legado*. Pudiendo utilizarse *iptables* no tiene sentido mantener instalado *ipchains*, que aún es utilizado por defecto en Red Hat Linux (TM) 7.1 y 7.2. Se recomienda desinstalar *ipchains* y los paquetes que dependan de este.

Es importante utilizar la más reciente versión de *iptables* para la distribución utilizada. Ninguna versión de *iptables* anterior a la 1.2.4 se considera como apropiada debido a fallas de seguridad de gran importancia, y ningún administrador *competente* utilizaría una versión inferior a la 1.2.4. Por favor visite el sitio Web de su distribución predilecta para estar al tanto de cualquier aviso de actualizaciones de seguridad. Ejemplo: para Red Hat Linux 7.1 y 7.2 hay paquetería de actualización en los siguientes enlaces:

- <ftp://updates.redhat.com/7.1/en/os/i386/>, si posee alguna distribución basada sobre Red Hat(TM) Linux 7.1
- <ftp://updates.redhat.com/7.2/en/os/i386/>, si posee alguna distribución basada sobre Red Hat(TM) Linux 7.2

Antes de desinstalar *ipchains*, primero debe eliminarse cualquier regla que pudiese existir.

```
/sbin/ipchains -X  
  
/sbin/ipchains -F  
  
/sbin/ipchains -Z
```

A continuación debe removerse el módulo de *ipchains* para permitir la carga del módulo *ip_tables*.

```
/sbin/rmmod ipchains  
  
/sbin/modprobe ip_tables
```

Para terminar, se desinstala *ipchains* y toda la paquetería que dependa de éste.

```
rpm -e ipchains lokkit gnome-lokkit firewall-config
```

Esto ajustes deben poder permitir utilizar *iptables* en lugar de *ipchains* sin mayor problema.

Re-direccionamiento de peticiones.

En un momento dado se requerirá tener salida transparente hacia Internet para ciertos servicios, pero al mismo tiempo se necesitará re-direccionar peticiones hacia servicio Web, Web SSL, ftp, gopher o WAIS hacia el el puerto donde escucha peticiones Squid (3128), de modo que no haya salida alguna hacia alguno de estos protocolos sin que ésta pase antes por Squid.

El re-direccionamiento lo hacemos a través de *iptables*. Considerando para este ejemplo que la red local se accede a través de una interfaz eht0, el siguiente esquema ejemplifica un re-direccionamiento:

```
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port  
3128
```

Lo anterior hace que cualquier petición hacia el puerto 80 (servicio HTTP) hecha desde la red local hacia el exterior, se re-direccionará hacia el puerto 3128 del servidor.

Considerando lo anterior, con el fin de re-direccionar peticiones hacia los puertos 20 (FTP-data), 21 (FTP), 70 (GOPHER), 80 (HTTP), 210 (WAIS) y 443 (HTTPS), podemos añadir al guión del muro contrafuegos lo siguiente:

Re-direccionamiento de servicios ordinarios con iptables.

```
# FTP-data
```

```
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 20 -j REDIRECT --to-port  
3128
```

```

# FTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j REDIRECT --to-port
3128

# GOPHER

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 70 -j REDIRECT --to-port
3128

# HTTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
3128

# WAIS

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 210 -j REDIRECT --to-port
3128

# HTTPS

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-port
3128

```

Puede añadirse re-direccionamiento hacia otros puertos menos usuales pero que llegan ser utilizados para acceder hacia algún servicio, com por ejemplo el 81, utilizado en ocasiones como alternativo para HTTP, y el 563, utilizado para NNTP sobre SSL.

También se pueden re-direccionar los puertos utilizados por los clientes de mensajería instantánea, siempre que estos permitan hacer uso de un servidor proxy, ya que de lo contrario quedarían bloqueados, como sería el caso del protocolo ICQ, mismo que no tiene soporte para servidor proxy.

- AIM: puertos 9898, 5190 al 5193.
- Yahoo! Messenger: puertos 5050 u 80 para mensajes, 5000 al 5010 para conversaciones por voz y 5100 para vídeo.
- MSN Messenger: puerto 1863, y 80 si no puede usarse el primero.

Re-direccionamiento de otros servicios con iptables.

```

# A veces utilizado para HTTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 81 -j REDIRECT --to-port
3128

# NNTP sobre SSL

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 563 -j REDIRECT --to-port
3128

# AIM

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 9898 -j REDIRECT --to-port
3128

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5190:5193 -j REDIRECT
--to-port 3128

# Yahoo! Messenger

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5000:5010 -j REDIRECT
--to-port 3128

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5050 -j REDIRECT --to-port
3128

# MSN Messenger

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 1863 -j REDIRECT --to-port
3128

```

Guión ejemplo de Enmascaramiento de IP con iptables.

El guión que mostramos en la tabla a continuación considera que se dispone de dos interfaces: eth0 y eth1. Para nuestro ejemplo la red local se accede por la interfaz eth0 y la salida hacia Internet se hace por la interfaz eth1. Utilice Firestarter para configurar el enmascaramiento y muro contrafuegos siempre que sea posible. Este guión NO es sustituto para un guión de muro contrafuegos.

Guión básico de Enmascaramiento de IP.

```
#!/bin/sh

# cargamos los módulos del kernel necesarios:

/sbin/modprobe ip_conntrack

/sbin/modprobe ip_conntrack_ftp

/sbin/modprobe ip_conntrack_irc

/sbin/modprobe ipt_REJECT

/sbin/modprobe ipt_REDIRECT

/sbin/modprobe ipt_TOS

/sbin/modprobe ipt_MASQUERADE

/sbin/modprobe ipt_LOG

/sbin/modprobe iptable_mangle

/sbin/modprobe iptable_nat

/sbin/modprobe ip_nat_ftp

/sbin/modprobe ip_nat_irc

# Habilitamos el reenvío de direcciones IP

if [ -e /proc/sys/net/ipv4/ip_forward ]; then

echo 0 > /proc/sys/net/ipv4/ip_forward

fi

# Estableciendo política de reenvío del enmascaramiento

/sbin/iptables -t filter -P FORWARD DROP

# Reenvío de tráfico intento-externo y externo-interno

/sbin/iptables -t filter -A FORWARD -d 0/0 -s 192.168.1.0/255.255.255.0 -o eth0 -j ACCEPT

/sbin/iptables -t filter -A FORWARD -d 192.168.1.0/255.255.255.0 -j ACCEPT

# Enmascaramiento de todo el tráfico saliente

# NOTA: recordemos que la salida hacia Internet es por

# la interfaz eth0

/sbin/iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

# No enmascararemos tráfico externo

/sbin/iptables -t nat -A POSTROUTING -o eth1 -d 0/0 -j ACCEPT

# Permitir al tráfico de la red interna ir a donde sea

/sbin/iptables -t filter -A INPUT -s 192.168.1.0/255.255.255.0 -d 0/0 -j ACCEPT

/sbin/iptables -t filter -A OUTPUT -s 192.168.1.0/255.255.255.0 -d 0/0 -j ACCEPT
```

```
/sbin/iptables -t filter -A OUTPUT -p icmp -s 192.168.1.0/255.255.255.0 -d 0/0 -j ACCEPT

# Re-direccionamiento hacia el puerto 3128 (donde Squid escucha
# peticiones) para cualquier petición originada desde la red
# local hacia servicios que utilicen protocolo http, https y ftp
# Pueden añadirse más re-direccionamientos a discreción del
# administrador.

# NOTA 1: recordemos que la red local se accede con la interfaz eth1
# NOTA 2: Si se utiliza un Proxy transparente, deberá cambiarse
# --to-port 3128 por --to-port 80

# FTP-data

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 20 -j REDIRECT --to-port
3128

# FTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j REDIRECT --to-port
3128

# GOPHER

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 70 -j REDIRECT --to-port
3128

# HTTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
3128

# WAIS

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 210 -j REDIRECT --to-port
3128

# HTTPS

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-port
3128

# A veces utilizado para HTTP

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 81 -j REDIRECT --to-port
3128

# NNTP sobre SSL

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 563 -j REDIRECT --to-port
3128

# AIM

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 9898 -j REDIRECT --to-port
3128

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5190:5193 -j REDIRECT
--to-port 3128

# Yahoo! Messenger

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5000:5010 -j REDIRECT
--to-port 3128

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5050 -j REDIRECT --to-port
3128
```

```
# MSN Messenger
```

```
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 1863 -j REDIRECT --to-port 3128
```

3. Cómo configurar Squid: Restricción de acceso a sitios Web.

Introducción.

Denegar el acceso a ciertos sitios Web permite hacer un uso más racional del ancho de banda con el que se dispone. El funcionamiento es verdaderamente simple, y consiste en denegar el acceso a nombres de dominio o direcciones Web que contengan patrones en común.

Este manual considera que usted ya ha leído previamente, a detalle y en su totalidad el manual "Como configurar Squid: Servidor Proxy" y que ha configurado exitosamente Squid como servidor proxy.

Software requerido.

Para poder llevar la cabo los procedimientos descritos en este manual y documentos relacionados, usted necesitará tener instalado al menos squid-2.4STABLE1.

Definiendo patrones comunes.

Lo primero será generar una lista la cual contendrá direcciones Web y palabras usualmente utilizadas en nombres de ciertos dominios. Ejemplos:

```
www.sitioporno.com
www.otrositioporno.com
sitioundeseable.com
otrositioundeseable.com
napster
sex
porn
mp3
xxx
adult
warez
```

Esta lista, la cual deberá ser completada con todas las palabras (muchas de está son palabras obscenas en distintos idiomas) y direcciones Web que el administrador considere pertinentes, la guardaremos como */etc/squid/sitios-denegados*.

Parámetros en */etc/squid/squid.conf*

Debemos definir una *Lista de Control de Acceso* que as u vez defina al fichero */etc/squid/sitios-denegados*. Esta lista la denominaremos como *denegados*. De modo tal, la línea correspondiente quedaría del siguiente modo:

```
acl negados url_regex "/etc/squid/sitios-denegados"
```

Habiendo hecho lo anterior, deberemos tener en la sección de *Listas de Control de Acceso* algo como lo siguiente:

```
#
# Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl redlocal src 192.168.1.0/255.255.255.0
acl password proxy_auth REQUIRED
acl negados url_regex "/etc/squid/sitios-denegados"
```

A continuación especificaremos una regla de control de acceso para dicha *Lista de Control de Acceso*:

```
http_access deny negados
```

Note que esta debe ir antes de cualquier otra regla que permita el acceso a cualquier otra lista. Ejemplo:

```
Reglas de control de acceso: denegación de sitios.
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
http_access deny negados
http_access allow localhost
http_access allow redlocal password
http_access deny all
```

Si por ejemplo el incluir una palabra en particular afecta el acceso a un sitio Web, puede generarse una lista de dominios o palabras que contengan un patrón pero que consideraremos como apropiados.

Como ejemplo: vamos a suponer que en la lista de sitios denegados está la palabra *sex*. esta denegaría el acceso a cualquier nombre de dominio que incluya dicha cadena de caracteres, como *extremesex.com*. Sin embargo también estaría bloqueando a sitios como *sexualidadjovel.cl*, el cual no tiene que ver en lo absoluto con pornografía, sino orientación sexual para la juventud. Podemos añadir este nombre de dominio en un fichero que denominaremos */etc/squid/sitios-inocentes*.

Este fichero será definido en una Lista de Control de Acceso del mismo modo en que se hizo anteriormente con el fichero que contiene dominios y palabras denegadas.

```
acl inocentes url_regex "/etc/squid/sitios-inocentes"
```

Para hacer uso de el fichero, solo bastará utilizar la expresión **!** en la misma línea utilizada para la *Regla de Control de Acceso* establecida para denegar el mismo.

```
http_access deny negados !inocentes
```

La regla anterior especifica que se denegará el acceso a todo lo que comprenda la *Lista de Control de Acceso* denominada *denegados* excepto lo que comprenda la *Lista de Control de Acceso* denominada *inocentes*. es decir, se podrá acceder sin dificultad a *www.sexualidadjoven.cl* manteniendo la restricción para la cadena de caracteres *sex*.

Finalmente, solo bastará reiniciar Squid para que tomen efecto los cambios y podamos hacer pruebas.

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

4. Cómo configurar Squid: Acceso por Autenticación.

Introducción.

Es muy útil el poder establecer un sistema de autenticación para poder acceder hacia Internet, pues esto permite controlar quienes si y quienes no accederán a Internet sin importar desde que máquina de la red local lo hagan. Sera de modo tal que tendremos un doble control, primero por dirección IP y segundo por nombre de usuario y contraseña.

Para tal fin nos valdremos de un programa externo para autenticar, como es *ncsa_auth*, de la NCSA (National Center for Supercomputing Applications), y que ya viene incluido como parte del paquete principal de Squid en la mayoría de las distribuciones actuales.

Este manual considera que usted ya ha leído previamente, a detalle y en su totalidad el manual "Como configurar Squid: Servidor Proxy" y que ha configurado exitosamente Squid como servidor proxy.

Software requerido.

Para poder llevar la cabo los procedimientos descritos en este manual y documentos relacionados, usted necesitará tener instalado al menos lo siguiente:

- squid-2.4.STABLE1
- apache-1.3.22

Procedimientos

Creación del fichero de contraseñas.

Se requerirá la creación previa de un fichero que contendrá los nombres de usuarios y sus correspondientes contraseñas (cifradas). El fichero puede localizarse en cualquier lugar del sistema, con la única condición que sea asequible para el usuario *squid*.

Debe procederse a crear un fichero */etc/squid/squid-passwd*:

```
touch /etc/squid/squid-passwd
```

Como medida de seguridad, este fichero debe hacerse leíble y escribible solo para el usuario *squid*:

```
chmod 600 /etc/squid/squid-passwd
```

```
chown squid:squid /etc/squid/squid-passwd
```

A continuación deberemos dar de alta las cuentas que sean necesarias, utilizando el comando *htpasswd* -mismo que viene incluido en el paquete *apache-1.3.22*-. Ejemplo:

```
htpasswd /etc/squid/squid-passwd joseperez
```

Lo anterior solicitará teclear una nueva contraseña para el usuario *joseperez* y confirmar tecleando ésta de nuevo. Repita con el resto de las cuentas que requiera dar de alta.

Todas las cuentas que se den de alta de este modo son independientes a las ya existentes en el sistema. Al dar de alta una cuenta o cambiar una contraseña lo estará haciendo exclusivamente para el acceso al servidor Proxy. Las cuentas son independientes a las que se tengan existentes en el sistema como serían *shell*, correo y Samba.

5. Bibliografía

<http://www.linuxparatodos.com>
<http://bulmalug.net>
<http://squid.nlanr.net/>
<http://www.opensourcefirewall.com>
<http://www.phpnuke-espanol.org>
www.squid-cache.org