Proxy Server - Squid

Documentado por Andrés Ruslan Abadías Otal | Nisamov

Introduccion:

Squid es uno de los proxy cachés más utilizados hoy en día y tiene como principal función atender a las peticiones HTTP de una red interna. Es un popular programa de software libre que implementa un servidor proxy y un dominio para caché de páginas web, publicado bajo licencia GPL.

Tiene una amplia variedad de utilidades, desde acelerar un servidor web, guardando en caché peticiones repetidas a DNS y otras búsquedas para un grupo de gente que comparte recursos de la red, hasta caché de web, además de añadir seguridad filtrando el tráfico.

Está especialmente diseñado para ejecutarse bajo entornos tipo Unix. Squid ha sido desarrollado durante muchos años y se le considera muy completo y robusto. Aunque orientado principalmente a HTTP y FTP es compatible con otros protocolos.

Principales características:

Squid posee las siguientes características:

- Proxy y Memoria Caché de HTTP, FTP, y otras direcciones: Squid proporciona un servicio de Proxy que soporta peticiones HTTP, HTTPS y FTP a equipos que necesitan acceder a Internet y a su vez provee la funcionalidad de caché especializado en el que se almacena de forma local las páginas consultadas recientemente por los usuarios. De esta forma, incrementa la rapidez de acceso a los servidores de la Web y FTP que se encuentra fuera de la red interna.
- Proxy por SSL: Squid también es compatible con SSL (Secure Socket Layer) con el que también acelera las transacciones cifradas, y es capaz de ser configurado con amplios controles de acceso sobre las peticiones de usuarios.
- Jerarquías de caché: Squid puede formar parte de una jerarquía de caches. Varios proxys trabajan conjuntamente sirviendo las peticiones de las páginas. Un navegador solicita siempre las páginas a un solo proxy, si éste no tiene la página en la caché hace peticiones a sus hermanos, que si tampoco las tienen las hacen a su/s padre/s. Estas peticiones se pueden realizar mediante dos protocolos: HTTP e ICMP.
- ICP, HTCP, CARP, memoria caché digests: Squid sigue los protocolos ICP, HTCP, CARP y memoria caché digests que tienen como objetivo permitir a un proxy "preguntar" a otros proxys caché si poseen almacenado un recurso determinado.
- Caché transparente: Squid puede ser configurado para ser usado como proxy transparente por lo que las conexiones son enrutadas dentro del proxy sin configuración por parte del cliente, y habitualmente sin que el propio cliente conozca de su existencia. De forma predeterminada,
- Squid utiliza el puerto 3128 para atender peticiones, pero se puede especificar que lo haga en cualquier otro puerto disponible o que lo haga en varios puertos disponibles a la vez.

- WCCP: A partir de la versión 2.3 Squid implementa WCCP (Web Cache Control Protocol).
- Permite interceptar y redirigir el tráfico que recibe un router hacia uno o más proxys caché, haciendo control de la conectividad de éstos. Además permite que uno de los proxys caché designado pueda determinar puede distribuir el tráfico redirigido a lo largo de todo el array de proxys caché.
- Control de acceso: Ofrece la posibilidad de establecer reglas de control de acceso. Esto permite establecer políticas de acceso en forma centralizada, simplificando la administración de una red.
- Aceleración de servidores HTTP: Cuando un usuario hace petición hacia un objeto en Internet, éste es almacenado en la caché, si otro usuario hace petición hacia el mismo objeto, y éste no ha sufrido ninguna modificación desde que lo accedió el usuario anterior, Squid mostrará lo que ya se encuentra en la caché en lugar de volver a descargar desde Internet. Esta función permite navegar rápidamente cuando los objetos ya están en la guarida y además optimiza enormemente la utilización del ancho de banda.
- SNMP: Squid permite activar el protocolo SNMP, éste proporciona un método simple de administración de red, que permite supervisar, analizar y comunicar información de estado entre una gran variedad de máquinas, pudiendo detectar problemas y proporcionar mensajes de estados.
- Caché de resolución DNS: Squid está compuesto también por el programa dnsserver, que se encarga de la búsqueda de nombres de dominio. Cuando Squid se ejecuta, produce un número configurable de procesos dnsserver, y cada uno de ellos realiza su propia búsqueda en DNS. De esta forma, se reduce la cantidad de tiempo que la caché debe esperar a estas búsquedas DNS.

Esquema de simulación:

PC-PT (Ubuntu Desktop) – `172.30.1.10`

Server-PT (Ubuntu Proxy Server) – `172.30.0.1`

`[Ubuntu Desktop]----[Switch]----[Ubuntu Proxy Server]----[Internet]`

Requisitos:

El software del servidor requiere la instalación, entre otros, del paquete squid (servidor).

Para llevar a cabo este proceso completamente, es necesario instalar lo siguiente dentro del equipo servidor (Ubuntu Proxy Server [172.30.0.1]):

- Apache2
- Squid
- Squidguard

```
sudo apt update
sudo apt install apache2
sudo apt install squid
sudo apt install squidguard
```

Revisión de Conexión

Antes de continuar, es necesario revisar si hay conexión entre los equipos, de lo contrario, no funcionará lo que se haga a continuación.

Fichero Configuración Squid:

El archivo de configuración es **squid.conf** situado en el directorio **/etc/squid/** resultante de la instalación. La configuración del servicio será pues modificando, eliminando o añadiendo nuevas opciones a este archivo. Por tanto se recomienda que antes de modificar nada en su interior se realice una copia de seguridad para poder restaurarlo a su estado original en caso de ser necesario. La ruta completa es: **/etc/squid/squid.conf**

Configuración del Proxy en Ubuntu Server:

Descomentar los parámetros **Proxy-cache»**, asegurándose de que tienen el valor indicado. Estos parámetros, han sido explicados previamente, es importante revisar la infromación para llevar a cabo la práctica sin problemas.

Permitir que cualquier ordenador de nuestra red interna tenga acceso a internet. Si existen otras redes preconfiguradas para poder acceder a internet, eliminar su permiso.

El proxy sólo debe permitir acceder al puerto **http (80)** y al **https (443)**, pero **no** al **ftp (21)**. Si existen otros puertos preconfigurados para poder acceder a internet, eliminar su permiso. (Comentar los puertos no indicados), dejando el siguiente resultado:

```
acl localnet src 0.0.0.1-0.255.255.255
acl localnet src 10.0.0.0/8
acl localnet src 100.64.0.0/10
acl localnet src 169.254.0.0/16
acl localnet src 172.16.0.0/16
acl localnet src 172.30.0.0/16           # Red local 172.30.0.0/16
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10

acl SSL_ports port 443
acl Safe_ports port 80
#acl Safe_ports port 21
acl Safe_ports port 443
#acl Safe_ports port 70
#acl Safe_ports port 210
#acl Safe_ports port 1025-65535
#acl Safe_ports port 280
#acl Safe_ports port 488
#acl Safe_ports port 591
#acl Safe_ports port 777
```

Reincia el servicio para aplicar los cambios, depende de la distribución usada, el comando para reiniciar el servicio puede variar: Distribuciones Debian/Ubuntu: `service squid restart`/`service squid3 restart`

En algunos casos, los servicios se reinician mediante el sitema init, es posible que puedas reiniciarlo mediante este servicio: `/etc/init.d/squid restart` Si no es viable ninguna opcion semejante, lo más seguro es apagar y volver a encender el equipo.

Configuración del Proxy en Ubuntu Desktop - Sistema Operativo y Navegador

Configurar manualmente el proxy de sistema en Ubuntu Desktop y modificar la configuración del navegador web de forma que se conecte a internet a través de este proxy de sistema. Conseguir acceder a dos sitios web de internet desde el navegador web de la máquina cliente.

Para llevar esto a cabo hay que realizar algunos ajustes en el proxy del Ubuntu Desktop, estableciéndolo en `configuración manual` y posteriormente, dentro de los parámetros indicados, hay que agregar lo siguiente:

Esta información es presentada más adelante nuevamente.

```
ip: 172.30.0.1 (La dirección Ip del Servidor)
Puerto: 3128
```

Lectura Manual de los Accesos a Internet a trvés del Proxy

Para este proceso, es necesario conocer las rutas de los ficheros de acceso **/var/log/squid/access.log**

Dentro de la ruta: **/var/log/squid/** Para mostrar las ultimas 50 lineas, se ha usado el comando: **sudo tail -n 50 access.log** Para mostrar todo el contenido pudiendo subir o bajar: **sudo less access.log**

Instalación y Configuración de SARG

Para continuar con el proceso, instalaremos SARG, este nos permitirá ver y previsualizar las direcciones a las que accede el cliente, de una manera más comoda.

Instalación de SARG:

```
sudo apt update
sudo apt install sarg
```

Ruta de SARG: **/etc/sarg/sarg.conf**

Cambios en el fichero sarg.conf

IDIOMA: si aparece esta opción, según la versión utilizada, se recomienda establecer el Español **'Spanish'**.

Ubicación de los archivos de SQUID access.log: revisar que la ruta es correcta.

Utilización de gráficas: para utilizar información visual se aconseja aplicar **graphs yes** y **graph_days_bytes_bar_color "ESCRIBIR-UN-COLOR-EN-INGLÉS"** para habilitar la visualización de las gráficas, cambiar el color naranja definido como predeterminado por otro color a su elección.

Directorio de salida de los informes: Se recomienda revisar la ruta donde se almacenarán los informes.

Es muy interesante visitar esta ruta para saber cómo SARG organiza la información.

Para elegir que los informes se guarden sin límites compruebe que la opción lastlog esté a 0.

Formato de fecha: elegir la opción e para establecer el formato europeo **(dd/mm/yy)**.

Información de SARG: habilitar la opción para ver la información que SARG extrae de los logs de SQUID. Para ello habilitar la opción **show_sarg_info** a **yes**.

Ejecución Manual de Informes

Una vez tenemos SARG configurado, sólo queda que genere informes de acceso de los datos guardados en los logs de squid. Para ello ejecutar el comando: **sudo sarg** (o bien) **sudo sarg-reports today**

[Troubleshooting]: Es posible que ejecutando el comando se devuelva un error respecto a la resolución de IPs; si es así, leerlo y arreglar el problema en el archivo de configuración para que ya no salga este aviso cuando ejecutemos el SARG.

Con este comando generará un informe con todos los datos almacenados en los logs de squid. También se puede realizar otros tipos de informes más detallados: por días, por franja de fechas, por una hora específica, por un usuario específico (éste último no es aplicable en este punto de la práctica), por un dominio destino específico, etc...

Primer Acceso a la Información de SARG

Para poder visualizar los informes generados por SARG sólo deberemos acceder desde un cliente y su navegador **http://IP_DEL_SERVIDOR/sarg**, en este caso sería **http://172.30.0.1/sarg**.

[Troubleshooting]: Si no es accesible se deberá generar un enlace simbólico del directorio donde se almacenan las páginas web de nuestro servidor en el directorio de salida de los informes del SARG con el comando: **sudo ln –sv /var/lib/sarg/ /var/www/html**

Una vez que hemos conseguido acceder, ver qué información proporciona esta herramienta una vez instalada y configurada.

Para establecer conexión en el dispositivo servidor, es necesario configurar el proxy en el equipo cliente de la siguiente forma:

- Poxy por configuración Manual
- Proxy por HTTPS: `172.30.0.1` por el puerto `3128` Todos los puertos se reenvían al puerto `3128`

Autenticación de Conexión por Usuario

A continuación configuraremos el servidor SQUID para habilitar el acceso a internet sólo a tres usuarios definidos por usted. En el nombre de estos usuarios deberá constar las iniciales de uno de los miembros del grupo, por ejemplo en este caso serían: abg1, abg2 y abg3.

Crear tres usuarios: Utilizamos el comando **-c** para crear una lista nueva de usuarios Usaremos el nombre del usuario como contraseña **(user:abg1 – passwd: abg1)**

```
sudo htpasswd -c /etc/squid/passwd abg1
sudo htpasswd /etc/squid/passwd abg2
sudo htpasswd /etc/squid/passwd abg3
```

Permitir a squid leer el archivo de usuarios:

```
sudo chmod 400 /etc/squid/passwd
sudo chown proxy /etc/squid/passwd
```

En el archivo de configuración de SQUID modificar los siguientes comandos:

```
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic casesensitive off
```

Finalmente modificar el acceso de los usuarios de nuestro proxy para dar sólo permiso y pedir autenticación a los usuarios creados anteriormente. Se deben añadir las siguientes líneas al archivo de configuración **squid.conf**:

```
acl ncsa_users proxy_auth REQUIRED
http_access allow ncsa_users
```

Posibles Preguntas sobre el Programa

¿Es posible saber si ha habido intentos de acceso no autorizado de algún usuario (credenciales mal escritas o erróneas)?:

Si, mediante los ficheros **cache.log**(Registro general de avisos y errores, aquí es posible ver accesos no autorizados) y **access.log**(Registro de acceso permitido por parte de dispositivos cliente conectados).

Las rutas de estos ficheros son: **/var/log/squid/access.log** y **/var/log/squid/cache.log**

Definición de Listas de Control de Acceso (ACL)

Las ACL son el mecanismo que determinan un elemento o grupos de elementos. Posteriormente estas ACL son utilizadas para determinar el acceso o prohibición a los elementos descritos anteriormente en el listado.

acl dominios_prohibidos dstdomain www.marca.es www.as.com

En esta ACL se describen dos dominios distintos que queremos que estén prohibidos. Para aplicar esta negación de servicio sólo deberemos utilizar esta ACL creada:

http_access allow localnet !dominios_prohibidos

En el comando anterior fíjemonos que lo que se hace es dar permiso a toda la red pero no al listado de dominios presentes en la ACL llamada **dominios_prohibidos**.

Podemos definir entonces la sintaxis general de las ACL según sus partes:

acl name type (string|"filename") [string2] [string3] ["filename2"]

Las ACL también pueden ser archivos que contengan los elementos a tener presentes. Por ejemplo, y siguiendo con el ejemplo anterior, podemos generar un archivo llamado dominios_prohibidos donde cada línea del archivo sea un dominio diferente.

Este archivo se crea normalmente en **/etc/squid/NOMBRE_DE_ACL**. Es decir, debe existir un archivo llamado dominios_prohibidos en **/etc/squid** que contenga en su interior los dominios a denegar.

Para crear esta ACL sólo deberemos hacerlode la siguiente manera:

ACL dominios_prohibidos dstdomain "/etc/squid/dominios_prohibidos"

Debe tenerse en cuenta que las ACL dependiendo del tipo de filtrado que se desea utilizar y su futura utilización. Es importanyr recordar que con escribir la línea de la ACL no es suficiente, hay que aplicarla posteriormente con una línia de acceso http **(http_access)**.