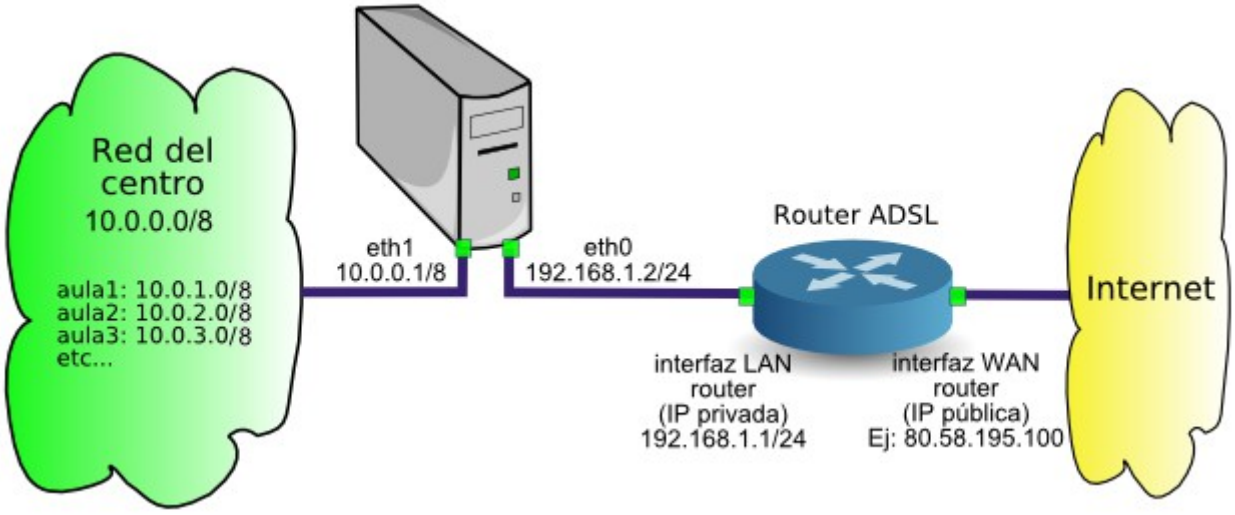
SERVIDOR PROXY

INTRODUCCIÓN

Un servidor 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 las peticiones irán dirigidas al proxy y este se las pasará al router.



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.

-Es fácil 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.

Inconvenientes de 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.

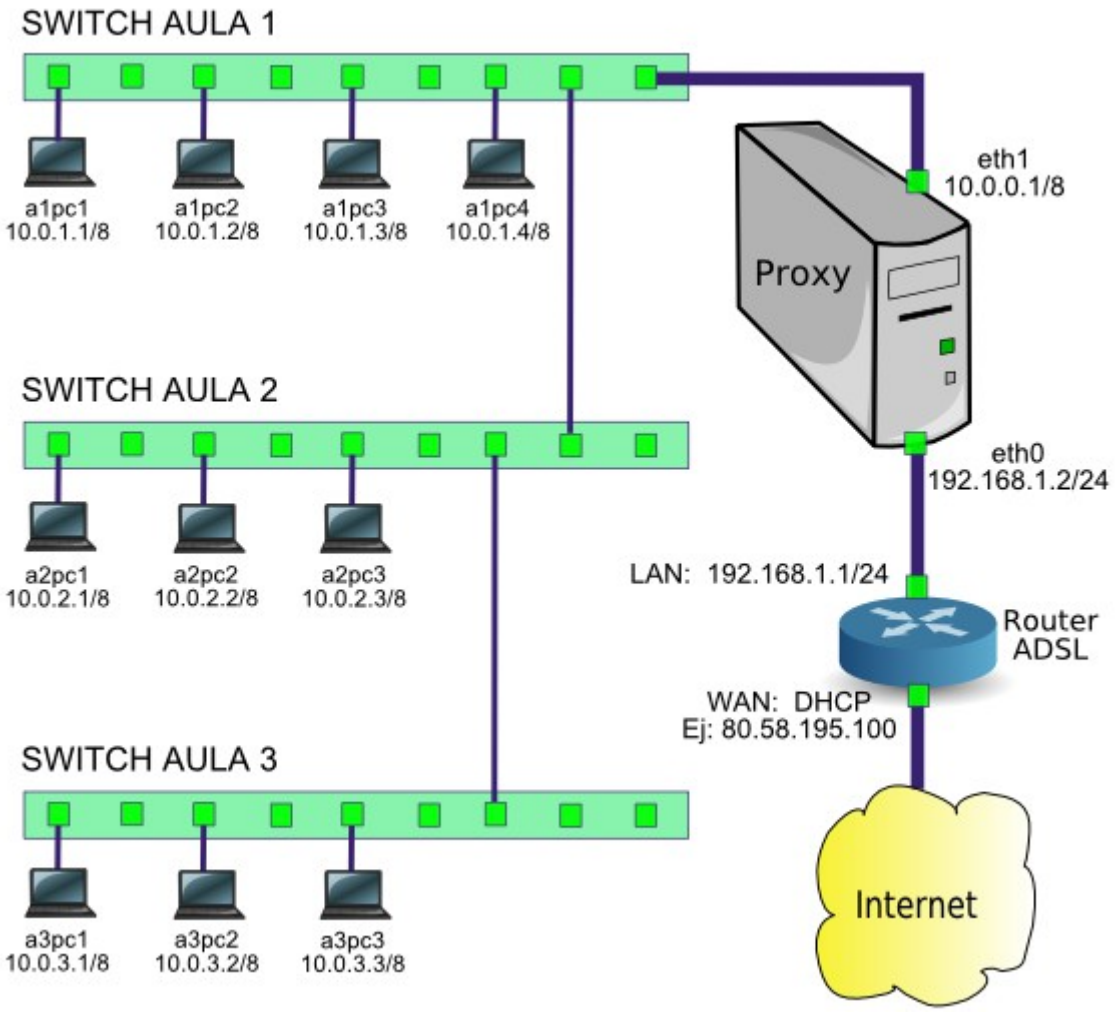
Diseño recomendado de la red del centro

Para facilitar la gestión del acceso a Internet en el centro, es recomendable diseñar la red de forma que cada aula tenga un rango de IPs determinado. Para no quedarnos cortos, lo mejor es utilizar el rango 10.0.0.0/8 siguiendo el esquema 10.W.X.Y donde W sería el número de edificio, X el número de aula e Y el número de PC, que nos permitiría tener un máximo de 254 edificios con 254 aulas cada uno y 254 PCs por aula.

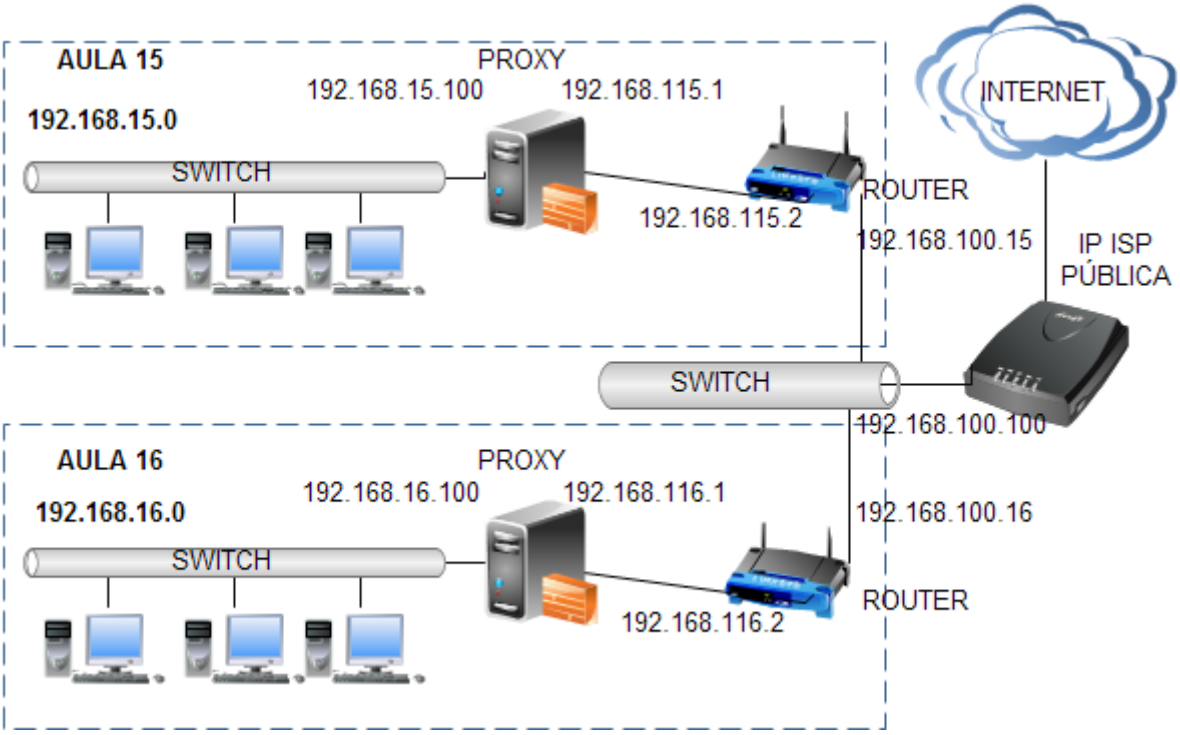
También podemos usar el rango de direcciones de **192.168.X.Y** donde **X representaría al número de aula** y **Y al host dentro de ese aula** incluso usando un servidor proxy y un router por aula para poder controlar aún más el acceso a Internet de cada aula.

Como podemos observar en las imágenes, lo normal es que la máquina que va a hacer de servidor proxy **disponga de dos interfaces de red** para poder separar físicamente las redes del centro, de la red donde estará el router que da salida a Internet. Si usamos un portátil podríamos usar la red cableada para conectar con la red del centro y la red wifi para conectar con la red del proxy/router o viceversa.

Ejemplo de un proxy para todo el centro:



Ejemplo de un proxy y router por aula:



SQUID PROXY

Dentro de los servidores mas importantes que existen en GNU/Linux existe el servidor proxy, puede controlar el acceso a Internet de nuestra red local, también es conocido como servidor intermedio, el servidor proxy más representativo en sus diferentes distribuciones es Squid. Squid es un programa que hace caché de datos obtenidos de Internet para poder optimizar recursos de banda ancha de Internet, entre sus características mas importantes son:

- **Proxy/caché:** Proporciona servicio proxy a peticiones del tipo http, https y ftp a equipos que se encuentran en nuestra red local para que puedan acceder hacia Internet y a su vez provee la funcionalidad de caché en el cual se almacenan localmente las páginas consultadas por los usuarios de forma que incrementa la rapidez de acceso a la información web y ftp.
- **Proxy SSL:** Es un servicio de squid compatible con SSL, con el cual se aceleran las peticiones y las peticiones hacia internet estarían cifradas.
- **Jerarquías de Caché:** Nuestro squid puede pertenecer a una jerarquía de caché que trabaja conjuntamente sirviendo peticiones. En este caso tendremos varios servidores squid resolviendo peticiones de una pagina web, si no la tiene registrada le pregunta a otro hasta que es encontrada la información.
- **ICP, HTCP, CARP, Caché digests:** Squid sigue los protocolos ICP, HTCP, CARP y caché digests que tienen como objetivo permitir a un proxy "preguntarle" a otros proxys caché si poseen almacenado un recurso determinado.
- **Proxy Transparente:** Puede ser configurado para ser usado como proxy transparente de manera que las solicitudes son enrutadas por medio de un reglas de firewall y sean enviadas al squid sin tener que configurar los clientes dentro de una red.
- **WCCP:** Permite interceptar y redirigir el trafico que recibe un router hacia uno o más proxys caché, haciendo control de la conexión de los mismos.
- **Control de Accesos:** En este parte establecemos reglas de control de acceso, esto permite establecer políticas de denegación o aceptación.
- **Aceleración de servidores HTTP:** Cuando hacemos peticiones hacia Internet la información es almacenada en el caché del squid y si hay otra solicitud hacia el mismo recurso el squid le devolverá la información que tiene el squid en caché. Si hay algún cambio entonces la información deberá ser actualizada.
- **SNMP:** Permite activar el protocolo SNMP, esto permite la 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 este modo, se reduce la cantidad de tiempo que la caché debe esperar a estas búsquedas DNS.

Instalación de Squid

Para instalar la última versión de squid, podemos hacerlo con apt-get desde una consola de root o usando sudo con el comando

```
apt-get install squid
```

Archivos de configuración del squid

Ya teniendo instalado nuestro servidor squid, ahora deberemos saber en donde se encuentra toda la configuración del mismo. La configuración está en el directorio

/etc/squid/

Dentro de este directorio se encontrarán varios archivos, pero el mas importante es el **squid.conf** el cual se encarga de la configuración del servicio.

/etc/squid/squid.conf

Como siempre es recomendable antes de editar un archivo de configuración de algún servicio hacer una copia de respaldo del original del mismo

```
sudo cp /etc/squid/squid.conf  /etc/squid/squid.ori
```

Arranque y parada de squid

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

Arrancar o reiniciar el servidor squid con cualquiera de estos dos comandos

```
sudo /etc/init.d/squid restart
sudo service squid restart
```

Parar el servidor squid con cualquiera de estos dos comandos

```
sudo /etc/init.d/squid stop
sudo service squid stop
```

Recargar configuración del servidor squid con cualquiera de estos dos comandos

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

Configuración básica del proxy squid

Como en cualquiera de los servicios que estamos viendo existen infinidad de parámetros configurables, aunque nosotros veremos los más interesantes para configurar nuestro servidor proxy.

OPTIONS FOR AUTHENTICATION (Opciones para autenticación)

Aquí se establecen las opciones de autenticación del Proxy. Aunque no vamos a usarlas en el curso, existe la posibilidad de configurar squid para que solicite usuario y contraseña para poder navegar por Internet. Si se quiere hacer uso de esta funcionalidad, lo normal sería tener almacenados los usuarios y las contraseñas en un servidor LDAP y en función de los grupos a los que pertenezcan los usuarios, podríamos habilitar o deshabilitar el acceso. Esto puede ser interesante en empresas, donde el administrador de red da acceso a Internet sólo a los usuarios que lo necesitan. Aunque en administraciones y empresas se usa mucho estas opciones, en un centro educativo supondría bastante trabajo llevar una administración de este tipo ya que habría que crear y gestionar un usuario para cada alumno y para cada profesor. Es más fácil administrar por redes y por aulas.

ACCESS CONTROL (Control de Acceso)

En esta sección estableceremos los permisos de acceso, es decir, quien puede navegar y quien 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.

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)

proto (protocolos)

Ejemplos:

Si en nuestra red local utilizamos el direccionamiento 192.168.2.0/24 (donde /24 es la máscara de red escrita en slash, es decir, que se usan 24 bits para ella), podemos crear una lista para definir a toda nuestra red con la siguiente lista:

acl todos src 192.168.2.0/24

De esta forma si en nuestra red local utilizamos el direccionamiento 192.168.X.0/24, para el aula X, podemos crear una lista para cada aula. En el ejemplo creamos una para las aulas 15 y 16:

acl aula15 src 192.168.15.0/24

acl aula16 src 192.168.16.0/24

Si dentro de una de las aulas quisiéramos crear una lista para los ordenadores de los profesores sería algo así, suponiendo que las ips abajo usadas son los dos ordenadores que usan los profesores del aula 15.

acl profes15 scr 192.168.15.10 192.168.15.11

Incluso podemos crear una lista usando un archivo donde incluiremos las ips a las que queremos aplicar dicha regla. Por ejemplo supongamos que queremos aplicar una regla al departamento de informática del centro

acl depinf scr /etc/squid/depinf

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.

Por ejemplo si quiero dar permiso a toda mi red para que navegue por Internet:

http_access allow todos

Si quiero dar permiso al aula15 para que navegue por Internet pero no quiero que navegue el aula 16:

http_access allow aula15
http_access deny aula16

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 equipos clientes 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 192.168.2.0/24, deberíamos añadir en /etc/squid/squid.conf:

acl todos src 192.168.2.0/24
http_access allow todos

NETWORK OPTIONS (Opciones de red)

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

http_proxy 3128

Squid puede trabajar en **modo transparente**. La ventaja de configurar squid en dicho modo de trabajo, es que no es necesario configurar el navegador de los equipos clientes para trabajar con el proxy, sino que simplemente configuramos la puerta de enlace del equipo 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:

http_proxy IP_servidor:3128 transparent
http_proxy 192.168.2.100:3128 transparent

Y luego Redirigir las peticiones al puerto 80 hacia el puerto 3128 en el cortafuegos

Reglas del Firewall para la configuración transparente

Para poder configurar este tipo de proxy transparente, tendremos que configurar reglas de firewall, en nuestro caso usaremos reglas de iptables ya que es la herramienta mas utilizada en todas distribuciones GNU/Linux. Pero para que funcione de manera transparente debemos de aplicar la siguiente regla en iptables.

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

Con esto estamos desviando el trafico que venga por la LAN que vaya por web al puerto 3128 donde eth1 es el interfaz de red a la escucha en el servidor. Con esto ya hicimos transparente nuestro proxy pero no se pueden desplegar las paginas seguras, para eso necesitamos aplicar otras reglas en iptables liberando el puerto 443, y lo hacemos de la siguiente manera:

```
sudo iptables -t nat -A PREROUTING -i eth1 -p tcp –dport 443 -j REDIRECT --to-port  3128
```

Habilitamos el reenvío de paquetes dentro de la red

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

Y Guardamos las reglas con el siguiente comando

```
sudo iptables-save > /etc/sysconfig/iptables
```

Reiniciamos el servicio de firewall

```
sudo /etc/init.d/iptables restart
```

Aunque de momento probaremos la configuración normal sin usar este modo hasta ver algo de las reglas de iptables.

MEMORY CACHE OPTIONS

Aquí estableceremos la memoria RAM utilizada para la caché. Una buena opción es utilizar sobre un tercio de la memoria RAM del sistema. Ejemplo, si nuestro sistema tiene 512 MB de memoria RAM, una buena opción sería:

cache_mem 192 MB

DISK CACHE OPTIONS

En esta sección estableceremos el espacio de disco duro utilizado para la caché. En nuestro caso como nuestro disco tiene 8GB usaremos **1GB para caché**, aunque estos valores pueden variar en función del sistema donde estemos implementando squid y de la carga del mismo. Deberemos utilizar la palabra clave **cache_dir** seguida de la palabra **ufs** que es el formato utilizado por squid, de la carpeta donde queremos que se almacene la cache, el tamaño de la caché en **MB**, el **número de subdirectorios de primer nivel** y el **número de subdirectorios de segundo nivel**. Ejemplo, si queremos que la caché se guarde en **/var/spool/squid**, que utilice 1 GB y que cachee hasta 16 subdirectorios de primer nivel y hasta 256 subdirectorios de segundo nivel, escribiremos:

cache_dir ufs /var/spool/squid 1000 16 256

Permitir/Denegar el acceso desde ciertos rangos de Ips o a urls concretas.

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_sin_inet.txt** en el que indicaremos los rangos de IPs que no queremos que naveguen. Por ejemplo, si no queremos que navegue el aula 16, el contenido del archivo /etc/squid/aulas_sin_inet.txt deberá ser:

192.168.16.0/24

Ahora vamos a crear la regla en el archivo **squid.conf** que deniegue el acceso a las redes de las aulas contenidas en este archivo

acl aulas_sin_inet src "/etc/squid/aulas_sin_inet.txt"
http_access deny aulas_sin_inet

Siempre que cambiemos algo en la configuración del archivo squid.conf tendremos que reiniciar el servicio o recargar la configuración de squid para que entre en funcionamiento la nueva configuración:

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

Denegando páginas web

Otra de las opciones que nos serán de gran utilidad usando squid es denegar el acceso a determinadas páginas web frecuentemente visitadas por nuestros alumnos en horario lectivo, con el fin de perder el tiempo. Para esto crearemos una lista dentro de un archivo llamado **/etc/squid/webs_prohibidas.txt** donde introduciremos una lista de estas urls a las que no queremos que tengan acceso desde los ordenadores del aula.

```
acl webs_prohibidas dst "/etc/squid/webs_prohibidas.txt"
http_access deny webs_prohibidas
```

El contenido del archivo /etc/squid/webs_prohibidas.txt podría ser

```
www.tuenti.com
www.msn.com
```

Denegando palabras en urls

Lista tipo url_regex

Permite especificar expresiones regulares para comprobar dicha url, a este tipo de regla se recomienda tener un archivo en cual agregamos todas la palabras que nosotros creamos que importantes. Nosotros crearemos el archivo **/etc/squid/palabrastabu.txt** con el siguiente contenido

```
sex
tuenti
```

Para definir la lista se usa

```
acl [Nombre] url_regex “Path”
acl palabrastabu url_regex “/etc/squid/palabrastabu.txt”
```

Resumiendo

La política a aplicar sería denegar las aulas sin Internet, denegar las webs prohibidas y las palabras prohibidas y 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 (Inserte sus propias reglas para permitir acceso desde sus clientes)

acl webs_prohibidas dst "/etc/squid/webs_prohibidas .txt"

http_access deny webs_prohibidas

acl aulas_sin_inet src "/etc/squid/aulas_sin_inet.txt"

http_access deny aulas_sin_inet

acl palabrastabu url_regex "/etc/squid/palabrastabu.txt"

http_access deny palabrastabu

acl todos src 192.168.2.0/24

http_access allow todos

Así, editando los archivos /etc/squid/aulas_sin_internet.txt , /etc/squid/webs_prohibidas.txt , /etc/squid/palabrastabu.txt y recargando la configuración de squid ejecutando sudo **/etc/init.d/squid reload,** podemos reconfigurar squid sin necesidad de tocar el archivo de configuración squid.conf

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
acl websprohibidas dst "/etc/squid/websprohibidas.txt"
http_access deny websprohibidas

acl aulas_sin_inet src "/etc/squid/aulas_sin_inet.txt"
http_access deny aulas_sin_inet

acl palabrastabu url_regex "/etc/squid/palabrastabu.txt"
http_access deny palabrastabu

acl todos src 192.168.2.0/24
http_access allow todos
```

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.**

/var/log/squid/access.log

Analizando periodicamente este archivo podremos ver a que páginas webs están

accediendo desde nuestra red por si procede ir añadiéndolas a la lista de webs prohibidas dentro del archivo /etc/squid/webs_prohibidas .txt que creamos anteriormente.

```
1334677352.777     83 192.168.2.105 TCP_MISS/200 6659 GET http://safebrowsing-c$
1334677353.066    103 192.168.2.105 TCP_MISS/200 25635 GET http://safebrowsing-$
1334677353.367    262 192.168.2.105 TCP_MISS/200 139727 GET http://safebrowsing$
1334677353.669     57 192.168.2.105 TCP_DENIED/403 1491 GET http://www.tuenti.c$
```

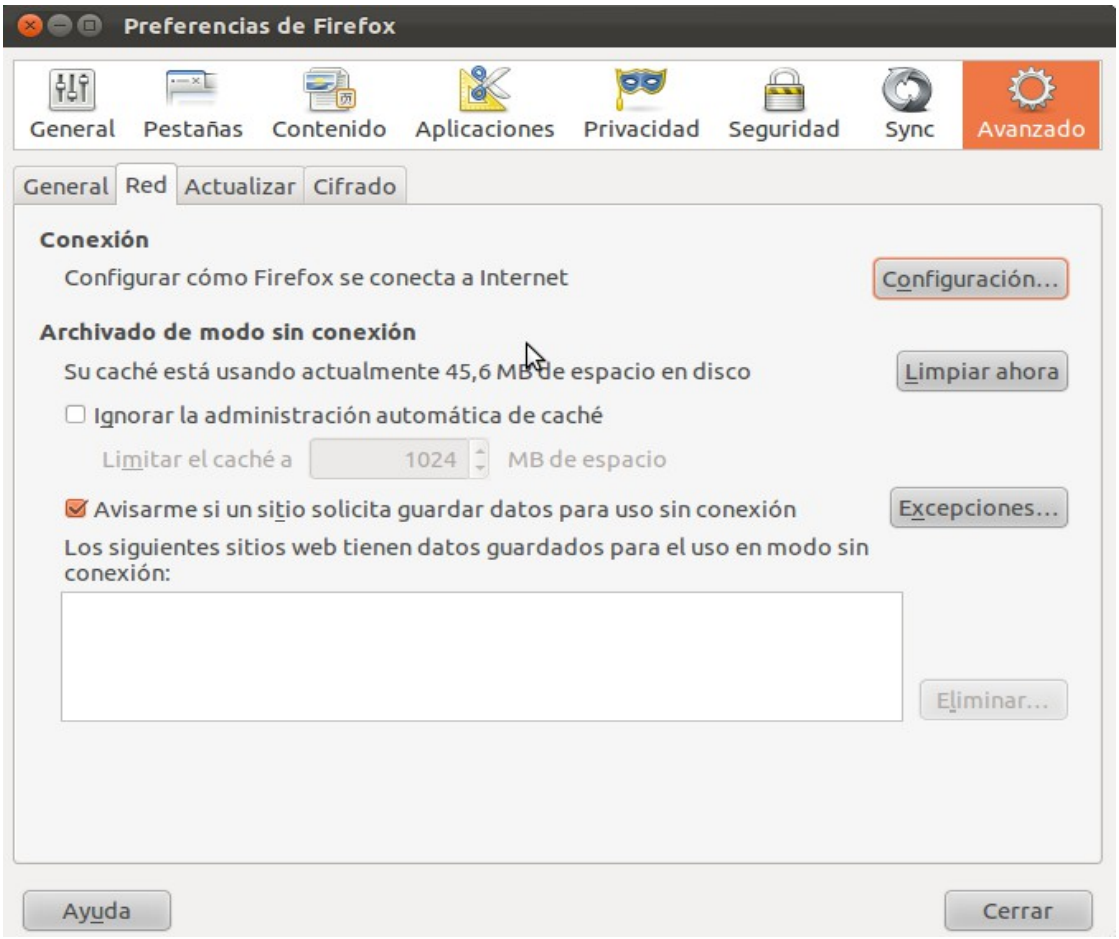
Configuración del navegador de los equipos clientes, para que utilicen el Proxy

Como hemos visto hay varias formas para configurar a los clientes que usen el proxy. La más sencilla es configurando directamente los navegadores de los clientes de manera que lancen sus peticiones al proxy de nuestra red.

Como existen multitud de navegadores nosotros veremos los que a nuestro parecer son los más utilizados actualmente.

Configuración de Mozilla Firefox

Para configurar Mozilla Firefox nos iremos al menú **Editar** → **Preferencias** o al menú **Herramientas** → **Opciones** (dependiendo de la versión y plataforma sobre la que este instalado). Una vez allí seleccionaremos la pestaña **Avanzado** y dentro de ella **Red**



Dentro de esa pestaña seleccionamos “**Configuración...”**

Y pinchamos en “**Configuración manual del proxy”** donde indicamos la IP de nuestro servidor y el puerto del proxy a la escucha que es el 3128 si no lo hemos modificado. Tambíen seleccionamos **“Usar el mismo proxy para todo”**

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.

Si intentamos cargar alguna de las páginas de la lista de webs_prohibidas debería aparecer el siguiente mensaje

ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: http://www.msn.com/

Access Denied.

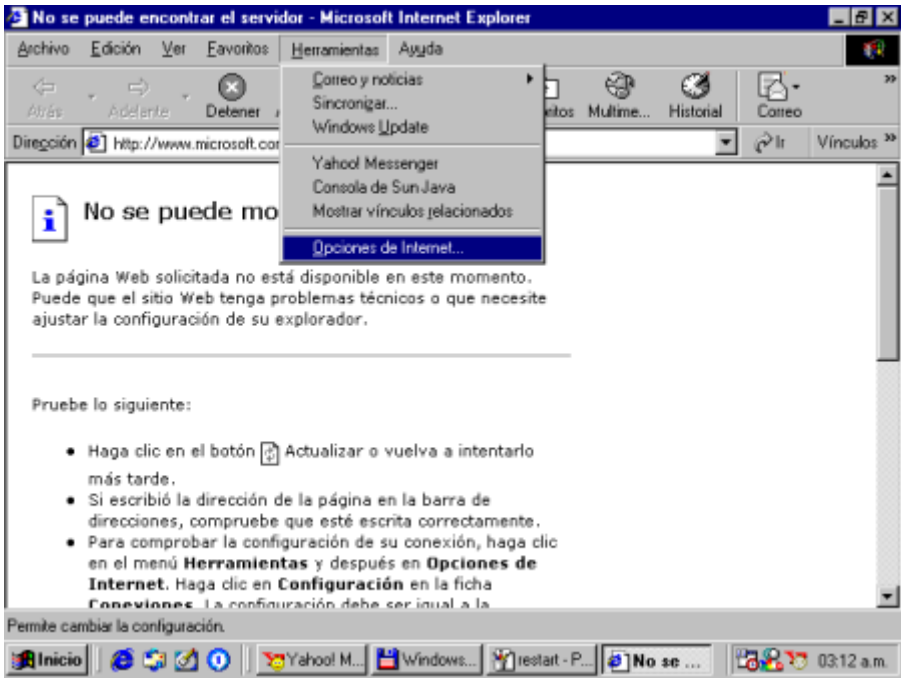
Access control configuration prevents your request from being allowed at this time. Please contact your service provider if you feel this is incorrect.

Your cache administrator is webmaster.

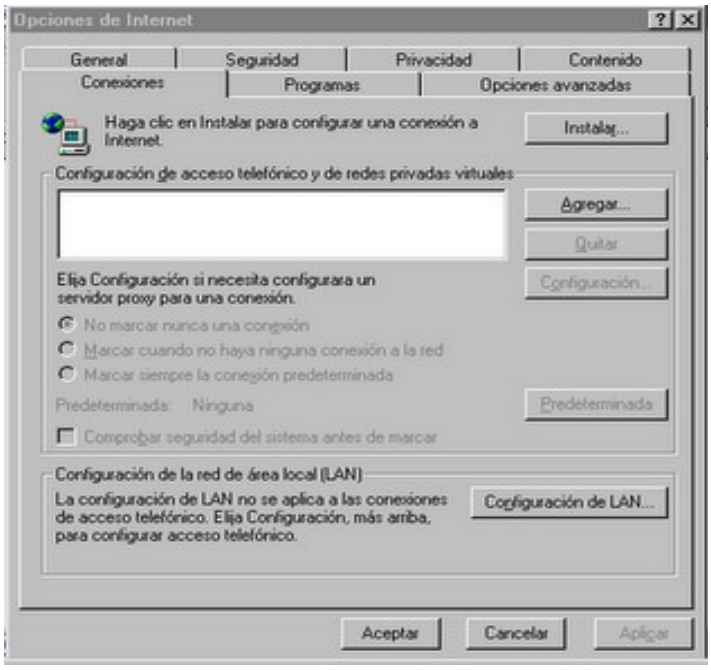
Generated Sun, 15 Apr 2012 20:55:02 GMT by ServidorCep (squid/2.7.STABLE9)

Configuración de Internet Explorer

Al igual que con Firefox para configurar Explorer debemos dirigirnos a **Herramientas → Opciones de Internet**



Seleccionamos la pestaña **Conexiones → Configuración de LAN...**



Pinchamos sobre **"Utilizar servidor proxy para su lan."** Introducimos la dirección IP de nuestro servidor y el puerto usado 3128 si no lo hemos modificado. Además comprobamos que esta seleccionado **"No usar servidor proxy para conexiones locales"**.

Configuración del proxy a través de un archivo de configuración automática.

Aparte de la configuración de proxy transparente que hemos visto, también podemos hacer una configuración de los clientes a través de un archivo de configuración situado en el servidor. En este archivo podemos definir redes a las queremos que se pueda acceder directamente y otros destinos que si tendrán que pasar a través del proxy. Como tenemos instalado nuestro servidor Apache el archivo lo guardaremos en **/var/www/proxy.pac** y podría tener el siguiente contenido donde estamos indicando que para acceder a la red local o a la dirección de loopback lo haga directamente, mientras que si es cualquier otra dirección debe pasar por el proxy.

```
function FindProxyForURL(url,host){
      if (isInNet(host, "192.168.2.0", "255.255.255.0"))
            return "DIRECT";
      else if (isInNet(host, "127.0.0.1", "255.255.255.255"))
            return "DIRECT";
      else return "PROXY 192.168.2.100:3128";
}
```

Usando expresiones regulares para excluir URLs

Si deseamos excluir las peticiones a determinado URL ya sea que contengan nombres de host planos, FQDN o direcciones IP usaremos la función **shExpMatch** para crear una expresión regular, por ejemplo:

```
if (shExpMatch(url, "http://192.168*"))
  return "DIRECT";
```

También puede usar el operador OR **(||)** para crear múltiples condiciones:

```
if (shExpMatch(url, "*vpn.example.com*") || shExpMatch(url,
"*webdelcentro.com/data/*"))
return "DIRECT";
```

Podemos usar esta opción cuando los clientes están habituados a acceder a recursos web internos por ejemplo: http://www.servidorcep.es, también es aconsejable este esquema cuando se conectan a sitios remotos probablemente por VPN y el acceso es por una ruta diferente a la del proxy.

En esta unidad hemos visto como configurar la mayoría de los parámetros básicos de nuestro servidor proxy, aunque existen más opciones de configuración que podemos seguir implementando hasta adaptar el proxy completamente a nuestras necesidades.