

Squid proxy web

By Kessusa

Lo primero que tenemos que hacer es descargarnos el proxy web a la máquina donde queramos configurarlo.

- ***sudo apt-get update***
- ***sudo apt install squid***

```
ambite@ambite-VirtualBox:~$ sudo apt-get install squid
[sudo] password for ambite:
Reading package lists... Done
```

Una vez que está instalado vamos a comprobar que está funcionando y en qué puerto escucha, esto lo podemos mirar en la página de squid que nos dirá el puerto en el que escucha por defecto pero si queremos mirarlo podemos ejecutar el siguiente comando.

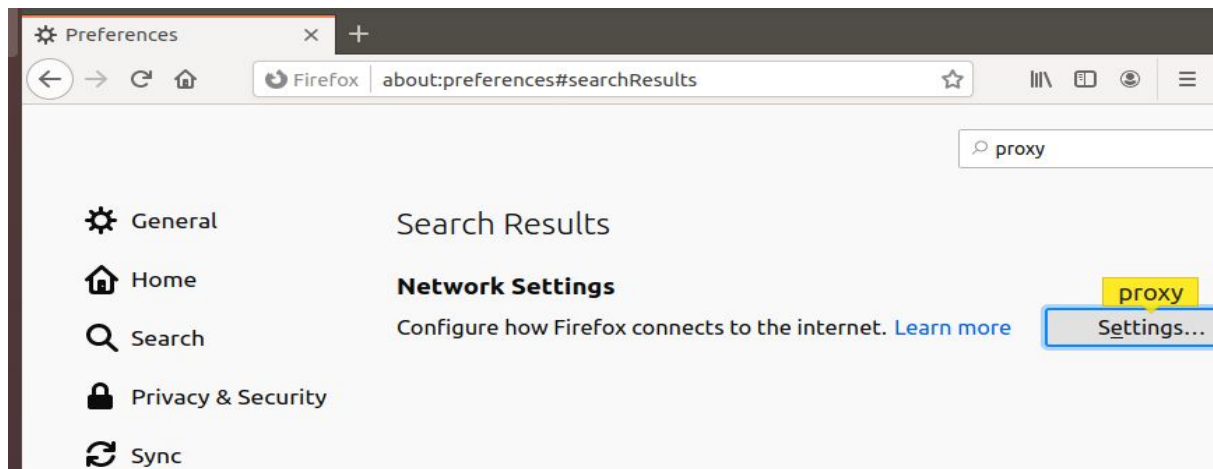
- ***sudo lsof -i -P -n | grep squid***

```
ambite@ambite-VirtualBox:/etc/squid$ sudo lsof -n -i -P | grep squid
[sudo] password for ambite:
squid        6158      proxy    6u     IPv6  229412      0t0  UDP *:46964
squid        6158      proxy    8u     IPv4  229413      0t0  UDP *:59358
squid        6158      proxy   11u     IPv6  229416      0t0  TCP *:3128 (LISTEN)
squid        6158      proxy   13u     IPv6  229418      0t0  UDP [::1]:54785->[::1]:45615
```

Podemos ver en la captura que está escuchando en el puerto TCP 3128 que es el que usa squid por defecto.

Lo siguiente para que probemos que está funcionando es ir a nuestro navegador web por defecto y configurar el proxy.

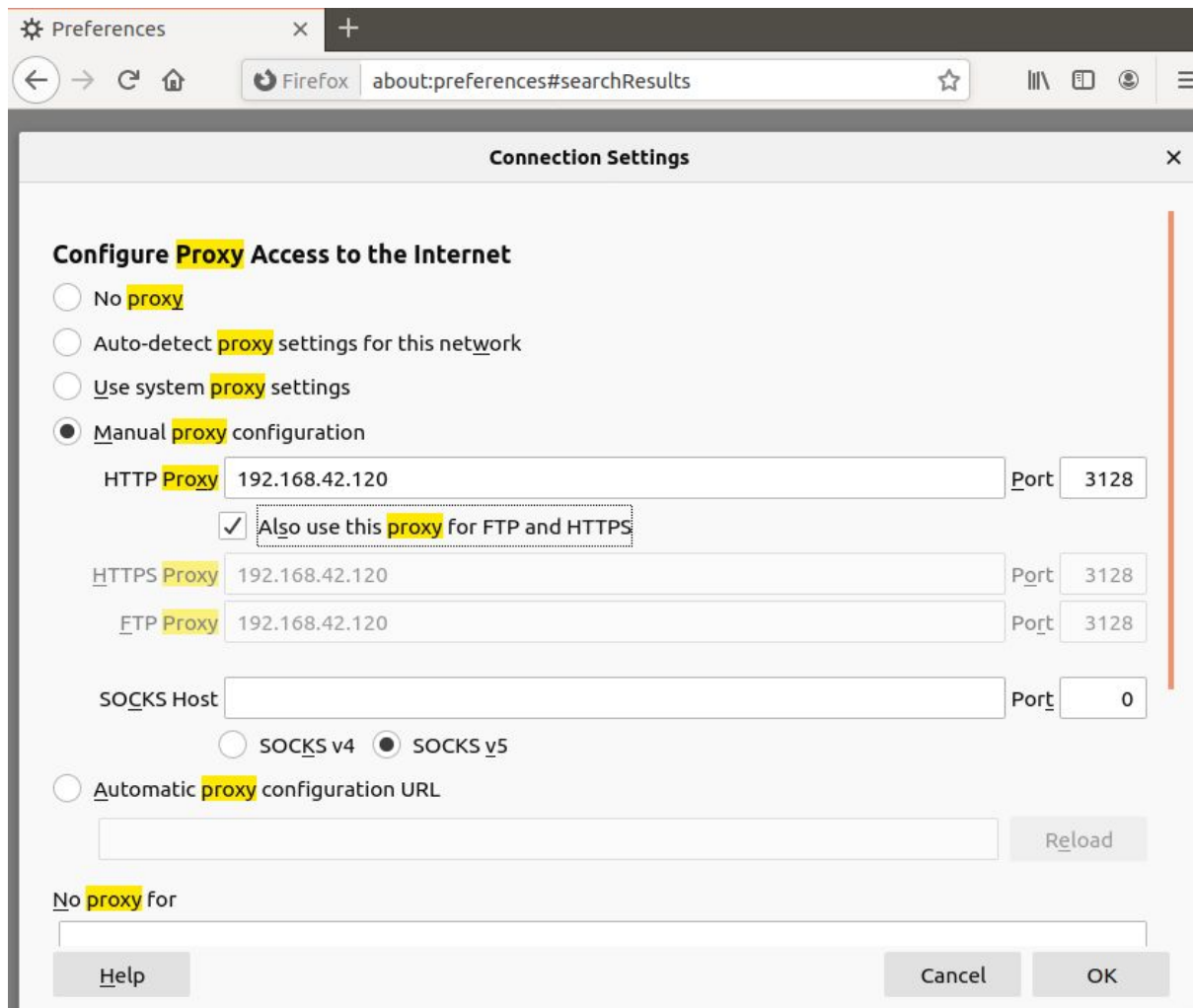
Por ejemplo en firefox tienes que ir a settings y puedes escribir proxy en el buscador y te saldrá lo siguiente.



Si pinchamos en settings podremos entrar a la página de configuración de proxy. Yo voy a configurarlo para la ip de mi ordenador 192.168.42.120

```
ambite@ambite-VirtualBox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:14:c6:24 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.120/24 brd 192.168.42.255 scope global dynamic noprefixroute enp0s3
```

Por tanto pondre la ip y el puerto en el que está escuchando que compramos que era 3128. Rellenamos los campos de la siguiente foto añadiendo que lo use también para las conexiones https y ftp y le doy a Ok.



Una vez que tenemos ya esto configurado podemos intentar ver si esta funcionando fácilmente buscamos algo en mozilla y nos vamos al fichero de logs del proxy a ver si lo está utilizando.

- **`sudo cat /var/log/squid/access.log`**

```
ambite@ambite-VirtualBox: /var/log/squid$ sudo cat access.log
1590397370.109 78 192.168.42.120 TCP_MISS/200 515 GET http://detectportal.firefox.com/success.txt - HIER_DIRECT/212.145.41.137 text/plain
1590397370.137 16 192.168.42.120 TCP_MISS/200 515 GET http://detectportal.firefox.com/success.txt? - HIER_DIRECT/212.145.41.137 text/plain
1590397370.158 37 192.168.42.120 TCP_MISS/200 515 GET http://detectportal.firefox.com/success.txt? - HIER_DIRECT/212.145.41.137 text/plain
1590397372.311 71 192.168.42.120 TCP_MISS/200 954 POST http://ocsp.digicert.com/ - HIER_DIRECT/93.184.220.29 application/ocsp-response
1590397373.163 22 192.168.42.120 TCP_TUNNEL/200 39 CONNECT snippets.cdn.mozilla.net:443 - HIER_DIRECT/13.224.118.18 -
1590397373.164 19 192.168.42.120 TCP_TUNNEL/200 39 CONNECT snippets.cdn.mozilla.net:443 - HIER_DIRECT/13.224.118.18 -
1590397373.289 163 192.168.42.120 TCP_MISS/200 856 POST http://ocsp.pki.goog/gts101 - HIER_DIRECT/216.58.211.227 application/ocsp-response
1590397373.642 41 192.168.42.120 TCP_MISS/200 954 POST http://ocsp.digicert.com/ - HIER_DIRECT/93.184.220.29 application/ocsp-response
1590397374.898 25 192.168.42.120 TCP_TUNNEL/200 39 CONNECT snippets.cdn.mozilla.net:443 - HIER_DIRECT/13.224.118.18 -
1590397375.902 156 192.168.42.120 TCP_MISS/200 856 POST http://ocsp.pki.goog/gts101 - HIER_DIRECT/216.58.211.227 application/ocsp-response
1590397376.450 144 192.168.42.120 TCP_MISS/200 856 POST http://ocsp.pki.goog/gts101 - HIER_DIRECT/216.58.211.227 application/ocsp-response
```

Vemos que si está utilizando el proxy.

Bueno para ver una utilidad real del proxy tenemos que empezar a usar acl y como se configura .

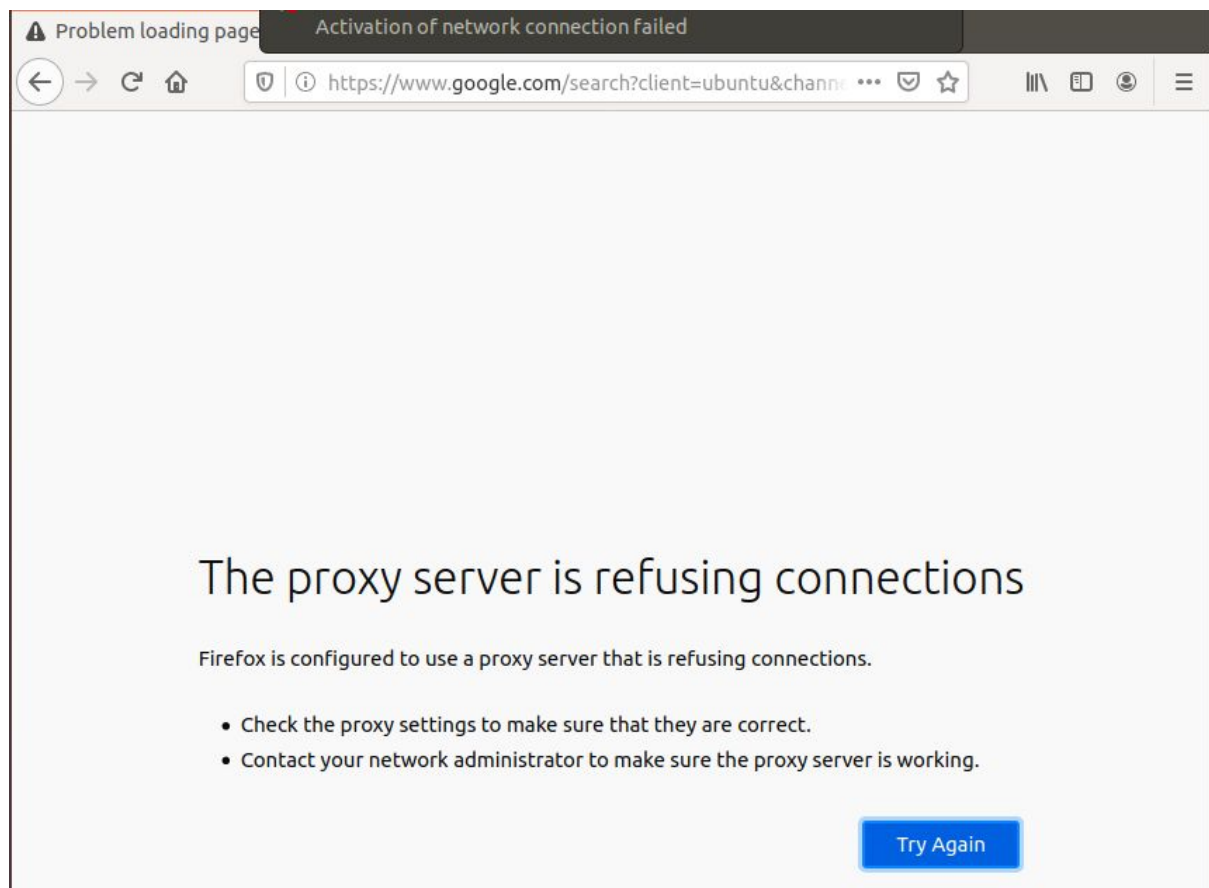
Con las acl podemos limitar la navegación y restringir de muchísimas formas, por url, dominio, ip de origen, ip de destino, horarios, personas autorizadas etc.

Estas acl se configuran en el archivo squid.conf que se encuentra en la ruta /etc/squid

En este archivo viene explicado los diferentes tipos de acl y como configurarlos. Una cosa muy importante que deberemos tener en cuenta a la hora de crear acl es lo primero que hasta que nos hacemos un http_access no se aplica la acl y segundo y muy importante que por temas de rendimiento las acl se ejecutan en orden descendente del fichero de configuración y una vez que encuentra una acl que permite la solicitud o que la de niega ya no sigue buscando por tanto si ponemos una restricción quw3e afecta a la petición antes de una que la permitiría nunca hará esa solicitud y al contrario igual si denegamos por acl algo que se permite en una acl que está antes en el fichero se ejecutará la solicitud web sin problemas.

Si tenemos la configuración como yo os he dicho al intentar navegar por internet os saldrá esto.

Esto es debido a que no hemos incluido aún una acl que permita la navegación desde nuestra ip , y por tanto está cortando todas las solicitudes que se hagan ahora mismo.



Entramos al fichero de configuración, como recomendación antes de modificar un fichero de configuración de cualquier servicio yo hago una copia de seguridad.

- **`sudo cp squid.conf squid.conf.back`**

Y una vez que tengo la copia me meto a configurarlo con el el gestor de ficheros que más te guste.

- ***sudo nano squid.conf***

Para permitir la navegación desde mi ordenador voy a crear una acl que sea de tipo origen para mi ip.

acl "Nombre que le queremos" dar src "mi ip"

ejemplo

acl mi_linux src 192.168.42.120

```
acl CONNECT method CONNECT
acl mi_linux src 192.168.42.120
```

Una vez que tenemos la acl definida nos vamos a ir a la parte del fichero de configuración en la que están los http_access.

Y vamos a colocar el siguiente http_access allow mi_linux.

```
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
http_access allow mi_linux
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
```

Guardamos el fichero y restablecemos el servicio de squid.

- ***sudo systemctl restart squid.service***

Una vez hacemos esto vemos que podemos navegar sin problemas desde nuestro ordenador.

Ahora vamos a probar a quitar un dominio por ejemplo marca.com y marca.es

Se puede hacer de dos formas o se hace referencia a un archivo donde ponemos los dominios que vamos a prohibir o hacemos ponemos el dominio directamente en el fichero de config, yo prefiero hacer un fichero ya que considero que es más limpio si vamos a denegar muchos dominios.

```
ambite@ambite-VirtualBox:/etc/squid$ ls
dominiosprohibidos  errorpage.css  squid.conf  squid.conf.back
ambite@ambite-VirtualBox:/etc/squid$
```

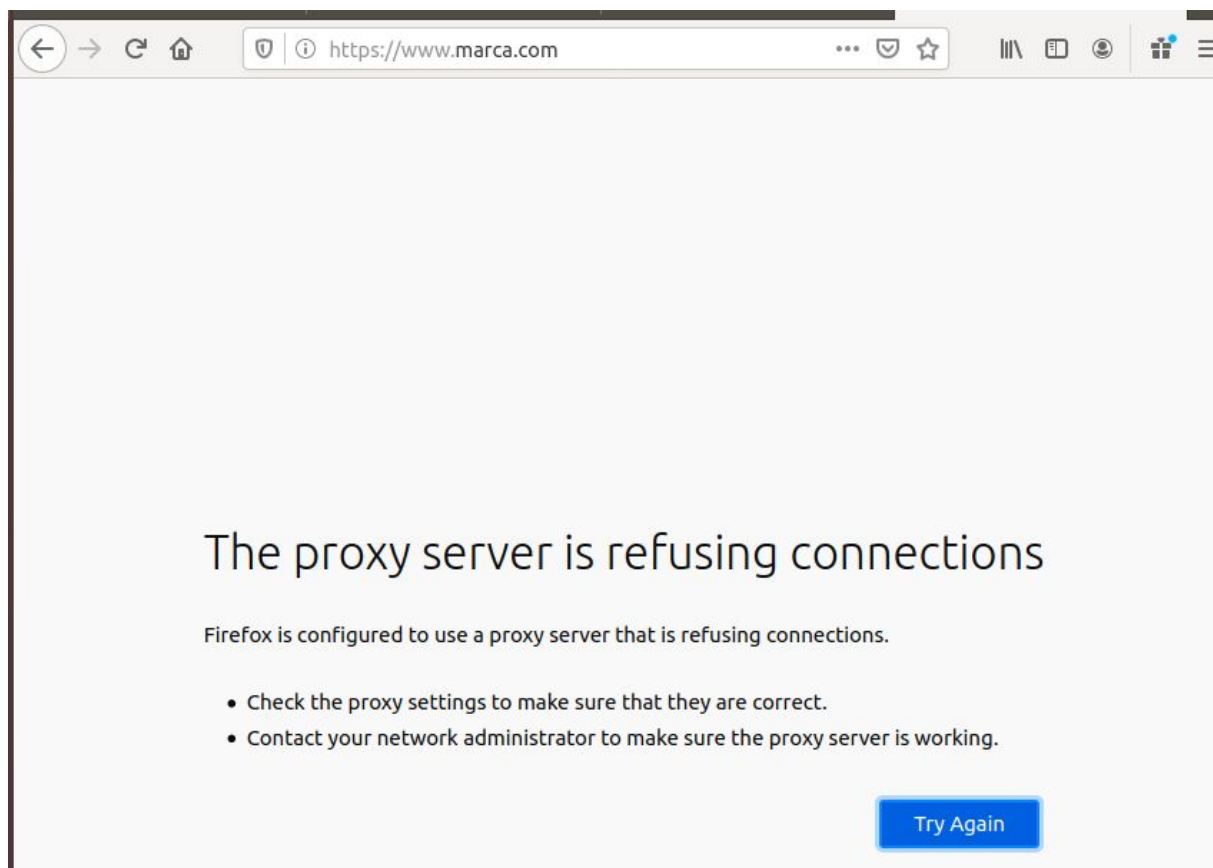


```
File Edit View Search Terminal Help
GNU nano 2.9.3 dominiosprohibidos
.marca.com
.marca.es
█
```

y una vez que tenemos el fichero creado con los dominio a los que queremos negar acceso vamos a crear el acl en nuestro fichero de configuración y aplicarlo.

```
acl dominio dstdomain "/etc/squid/dominiosprohibidos"
http_access deny dominio
```

Una vez que la hemos configurado restablecer el servicio y vemos como no nos va a dejar entrar a el dominio marca.com ni marca.es.



Podemos configurar una acl de horarios y combinarlo con los dominios de marca por tanto podemos dejar visitar marca si está dentro de un horario todos los días de diario de 12 a 3 de la tarde por ejemplo.

Para esto usamos un acl de tipo time

Algo así:

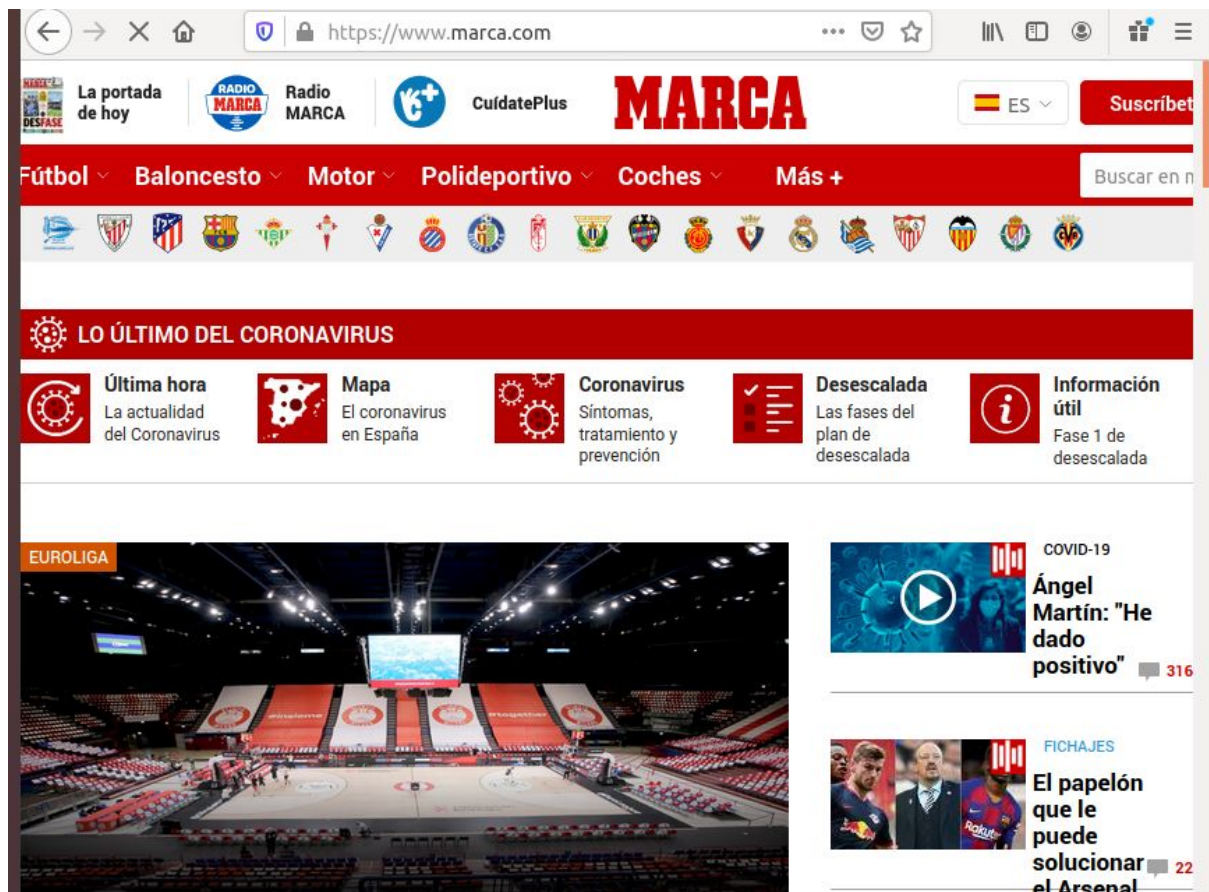
```
acl horario time MTWHF 12:00-15:00
```

http_access allow horario dominio

Vemos cómo hemos combinado en una dos acl y simplemente tenemos que poner los nombres y es como si se aplicara una operación and entre las acl si quisiéramos negar una de ellas usamos ! delante del nombre.

De nuevo una vez hechos los cambios,restablecemos el servicio y probamos a ver si podemos entrar ,.

En mi caso sí que puedo porque estoy dentro del horario que he definido.



Otro uso que se me ocurre bastante interesante es el de solicitar un usuario para poder navegar.

En este caso podemos usar htpasswd para crear los usuarios en un fichero en /etc/squid Esta utilidad no la tendremos disponible si no hemos instalada un servidor apache2,pero si no la tuviésemos con ejecutar el siguiente comando.

- ***sudo apt-get install apache2-utils***

Después es muy sencillo vamos a crear un usuario con nombre prueba.

```
ambite@ambite-VirtualBox:/etc/squid$ sudo htpasswd -c .usuarios prueba
New password:
Re-type new password:
Adding password for user prueba
ambite@ambite-VirtualBox:/etc/squid$
```

Una vez que lo hayamos ejecutado se habrá creado el fichero y la contraseña está cifrada. Como podemos ver al poner el punto delante el fichero está oculto.

```
ambite@ambite-VirtualBox:/etc/squid$ ls
dominiosprohibidos  errorpage.css  squid.conf  squid.conf.back
ambite@ambite-VirtualBox:/etc/squid$ sudo cat .usuarios
prueba:$apr1$H8kyr1IK$WbuFrLNXIai/jrch3Ntcj0
ambite@ambite-VirtualBox:/etc/squid$
```

Ahora tenemos que ir a el fichero de configuración y crear una acl de tipo auth.

Una cosa muy importante porque si no no funcionará es que el fichero de los usuarios debe pertenecer a squid o el usuario que haya creado el servicio en el sistema, porque luego será el que lo vaya a revisar y si no lo tiene no puede eso o le damos más permisos que los de root al fichero pero eso sería mala idea.

En mis sistema el usuario se llama proxy.

- ***sudo chown proxy .usuarios***

```
ambite@ambite-VirtualBox:/etc/squid$ sudo chown proxy .usuarios
```

```
ambite@ambite-VirtualBox:/etc/squid$ ls -l .usuarios
-rw-r--r-- 1 proxy root 45 may 25 12:59 .usuarios
ambite@ambite-VirtualBox:/etc/squid$
```

Ahora el fichero ya esa correctamente y tenemos un usuario creado lo siguiente es crear el acl.

Este acl tiene más parámetros de configuración que uno normal pero lo más importante es la primera línea en la cual pondremos la ubicación del módulo de autoidentificación que vamos a usar y la ruta a archivo donde están los usuarios que debe verificar.

La segunda línea especifica el número de veces que te pedirá la contraseña .

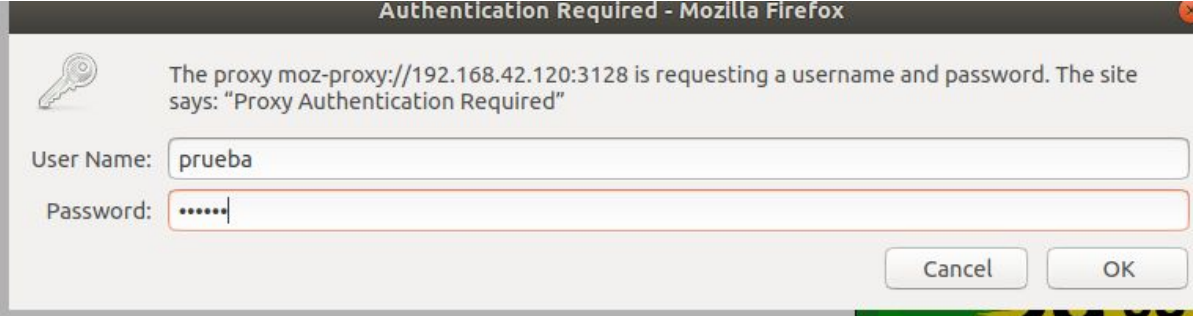
la cuarta cuánto tiempo se guarda el usuario en caché y la quinta línea se pone para no distinguir entre mayusculas o minusculas en el usuario, esto no afecta a la contraseña.

Todos estos se pueden modificar.


```
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/.usuarios
auth_param basic children 5
auth_param basic realm Proxy Authentication Required
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off

acl auth_users proxy_auth
http_access allow auth_users
```

Si lo configuramos bien al entrar al navegador nos saldrá algo así, y ahí pondremos el nombre y la clave del usuario creado.



Por último una consideración a tener en cuenta para el rendimiento del proxy es que hay acls que son slow y otras fast y para mejorar el rendimiento deberemos siempre que se prueba colocar primero las fast, por si al hacerlo ya deniega o permite el acceso será más rápido que si tiene que chequear primero las slow.

Esto no parece muy importante cuando tenemos pocas reglas o poco tráfico pero si es un proxy colocada en una red con mucho tráfico puede llegar a ralentizar mucho la navegación si no tenemos estas cosas en cuenta.