

## Enrutamiento en Ubuntu

Lo primero que hay que hacer es tener dos tarjetas de red puestas en la máquina.

Yo he puesto la primera como LAN en la red 192.168.1.1/24 y la segunda será la NAT con DHCP que será la que salga al exterior.

Nombre de la conexión: **LAN**

General Cableada Seguridad 802.1x Ajustes de IPv4 Ajustes de IPv6

Método: Manual

**Dirección**

Dirección	Máscara de red	Puerta de enlace	
192.168.1.1	255.255.255.0	0.0.0.0	<div>Añadir</div> <div>Eliminar</div>

Nombre de la conexión: **NAT**

General Cableada Seguridad 802.1x Ajustes de IPv4 Ajustes de IPv6

Método: Automático (DHCP)

**Dirección**

Dirección	Máscara de red	Puerta de enlace	
			<div>Añadir</div> <div>Eliminar</div>

Ahora hay que crear un script con el que enrutaremos.

Se le puede llamar como queramos, yo he puesto activar-enrutamiento.sh

En el script pondremos lo siguiente:

```
#!/bin/bash
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j MASQUERADE
```

```
GNU nano 2.2.6 Archivo: ...nit.d/activar-enrutamiento.sh
#!/bin/bash
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j MASQUERADE
```

A este script le pondremos los permisos 744 y el usuario y grupo serán root.

Ahora tenemos que decirle al sistema que cuando arranque lea ese fichero y para ello ponemos este comando:

```
sudo update-rc.d activar-enrutamiento.sh defaults
```

```
#sudo update-rc.d activar-enrutamiento.sh defaults
```

Ahora movemos el script a la carpeta /etc/init.d

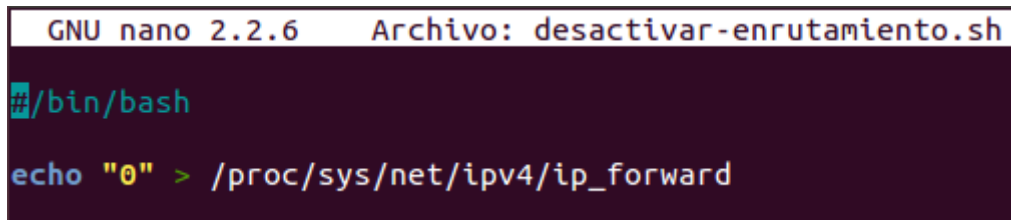
Lo ejecutamos y para ello ponemos:

```
sudo /etc/init.d/activar-enrutamiento.sh
```

Para poder desactivar el enrutamiento nos crearemos otro script que solo lo ejecutaremos cuando queramos desactivarlo. Tenemos que poner:

```
#/bin/bash
echo "0" > /proc/sys/net/ipv4/ip_forward
```

Solo hay que poner la variable ip\_forward a "0" y ya lo tendremos desactivado.

A screenshot of a terminal window with a dark background. The title bar at the top reads "GNU nano 2.2.6 Archivo: desactivar-enrutamiento.sh". The terminal content shows the following commands: 

```
/bin/bash
echo "0" > /proc/sys/net/ipv4/ip_forward
```

Ya tenemos hecho el enrutamiento y ahora pasaremos a crear el proxy.

## Crear un proxy en Ubuntu

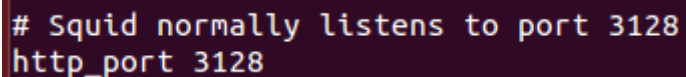
### Proxy No Transparente

Para crear un proxy hay que instalar el paquete squid

```
sudo apt-get install squid3
```

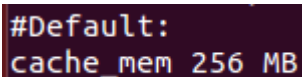
Ahora abrimos el archivo de configuración de squid: /etc/squid3/ squid.conf

Vemos el puerto que va a utilizar el squid y es el 3128. Se ve en la línea 1464

A screenshot of a text editor showing a configuration file. The visible lines are: 

```
# Squid normally listens to port 3128
http_port 3128
```

Nos vamos a la línea 2739 donde está la memoria caché y le quitamos el comentario.

A screenshot of a text editor showing a configuration file. The visible lines are: 

```
#Default:
cache_mem 256 MB
```

Nos vamos a la línea 3004 y le quitamos el comentario a cache\_dir

```
# Uncomment and adjust the following to add a d
cache_dir ufs /var/spool/squid3 100 16 256
```

Ufs es el tipo de sistema o codificación que va a utilizar para almacenar la información.

100 es el espacio en MB con el que va a trabajar squid para la cache.

16 es el número de directorios que se pueden llegar a crear en la cache.

256 es el número de directorios que se pueden crear dentro de los 16 directorios.

Nos vamos a la línea 919 donde están las ACL y añadimos dos nuevas ACL

```
acl red src 192.168.1/24
```

```
acl noway url_regex "/etc/squid3/noperm"
```

```
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
acl red src 192.168.1.0/24
acl noway url_regex "/etc/squid3/noperm"
```

El archivo noperm lo tendremos que crear y ahí pondremos las palabras que no queremos que aparezcan en el navegador.

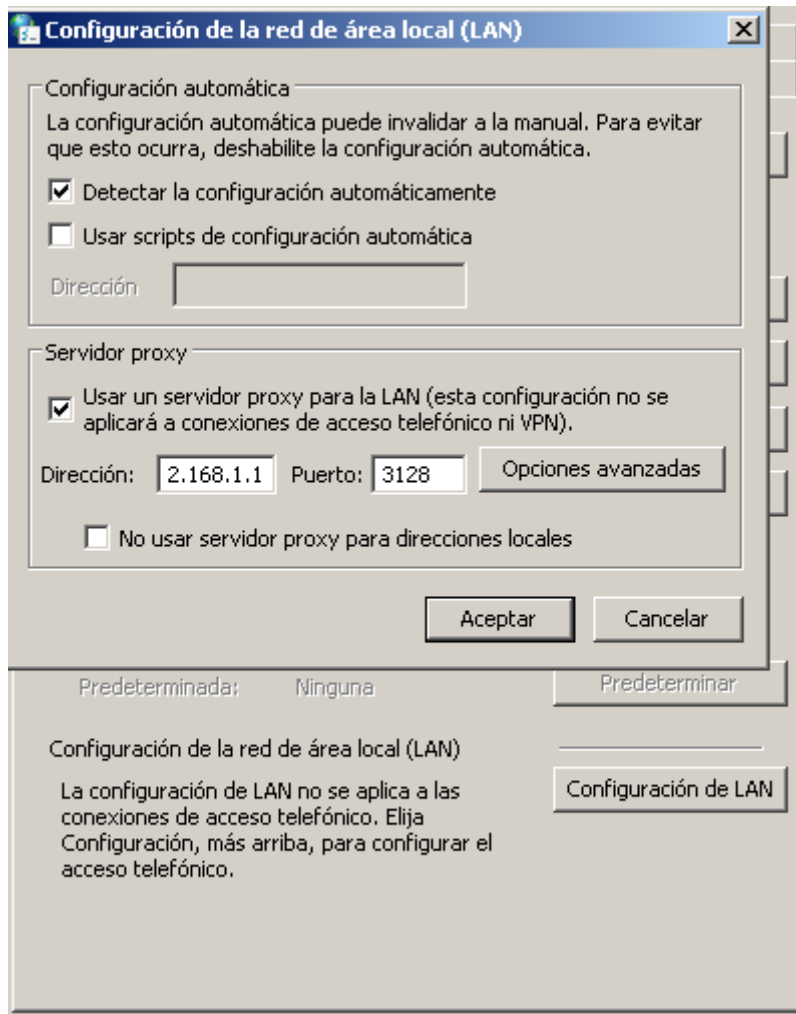
Hora nos tenemos que ir a la línea 1043 y pondremos las negaciones y los permisos en este orden.

```
# Only allow cachemgr access from localhost
http_access deny noway
http_access allow red
http_access allow localhost manager
http_access deny manager
```

Ya que tenemos hecho esto, guardamos el archivo, ponemos las palabras en el archivo noperm y reiniciamos el servicio.

```
service squid3 restart
```

Ya tenemos el servidor proxy hecho pero tenemos que configurar los navegadores para que pase por el servidor proxy.



## Proxy Transparente

En el archivo de configuración de squid3 tenemos que poner que queremos un proxy transparente y para ello lo ponemos de esta manera:

```
# Squid normally listens to port 3128
http_port 3128 transparent
```

Guardamos y salimos, obviamente tenemos que tener el archivo configurado como con el proxy no transparente.

Ahora tenemos que poner este comando para redirigir todas las peticiones del puerto 80 que provengan de la Red LAN al proxy.

```
usuario@usuario:~$ sudo iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/255.255.255.0 -d 0.0.0.0/0.0.0.0 -p tcp --dport 80 -j REDIRECT --to-port 3128
usuario@usuario:~$
```

Si queremos que cada vez que se inicie el sistema se active esta opción tenemos que hacer un script con ese mismo comando y lo guardaremos en /etc/init.d/ y le pondremos de nombre activar-proxy-transparente.sh

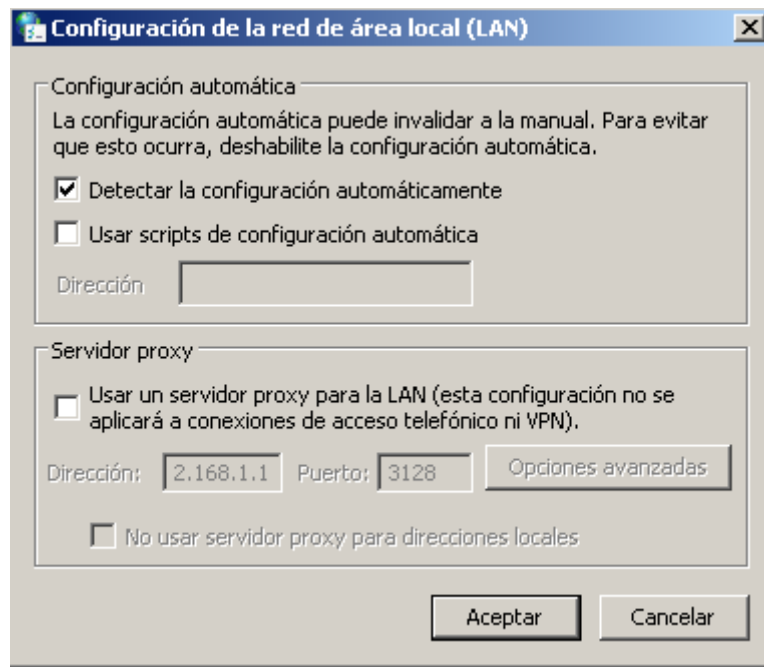
```
GNU nano 2.2.6 Archivo: /etc/init.d/activar-proxy-transparente.sh
/bin/bash
iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/255.255.255.0 -d 0.0.0.0/0.0.0.0 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

```
iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/255.255.255.0 -d 0.0.0.0/0.0.0.0
-p tcp --dport 80 -j REDIRECT --to-port 3128
```

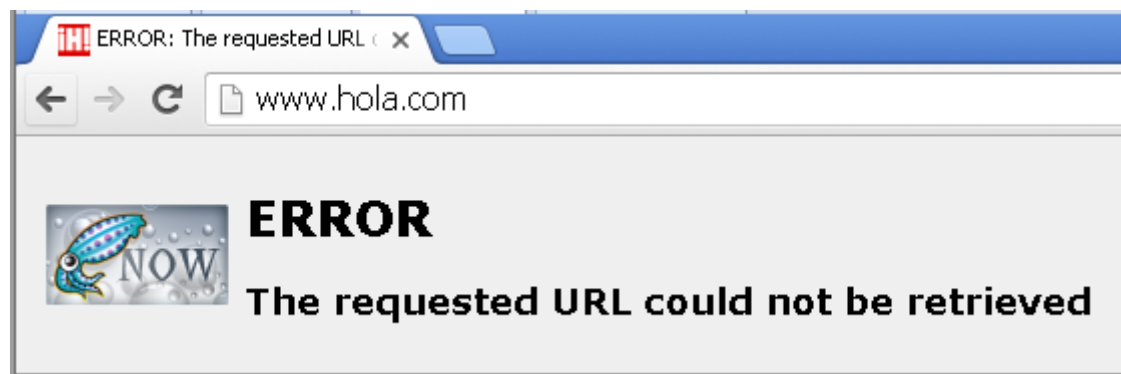
Le decimos al Sistema que cada vez que se inicie lea ese script.

```
sudo update-rc.d activar-proxy-transparente.sh defaults
```

En los clientes, tenemos que quitar de los navegadores el redireccionamiento al servidor proxy que habíamos puesto.



Probamos que no podamos entrar a ninguna página de las que habíamos bloqueado.



## ACL y Teoría de Squid

El proceso de squid está en /etc/init.d/squid3      service squid3 restart

Archivo de configuración de squid: /etc/squid3/ squid.conf

El fichero de caché está en: /ect/spool/squid3

Los archivos de información de acceso y de caché está en: /etc/log/squid3/[acces.log, cache.log]

Cambiar la información que sale al bloquear una página:

/usr/share/squid3/errors/Spanish/ERR\_ACCESS\_DENIED

### Sintaxis de la ACL

acl aclname acltipo valor

ACLTIPO: src, dst, dstdomain, url-regex, time

SRC: define a través de direcciones IP un cómputo de equipos de origen.

Ejemplo: acl mired src 192.168.1.0/24 ; acl aula1 src 192.168.2.0

DST: Genera una lista de acceso por direcciones IP de destino.

Ejemplo: acl prensadeportiva dst 193.110.128.199 194.169.201.186  
http\_access deny prensadeportiva

DSTDOMAIN:

acl nombreacl dtsdomain .dominio1 .dominio2

Ejemplo: acl prensadeportiva dtsdomain .marca.com .as.com

URL\_REGEX: bloquea las palabras para que no aparezcan en la URL.

Ejemplo: acl casijno url\_regex casino

TIME:

acl aclname time [días] [horas]

[Días] M, Lunes; T, Martes; W, Miercoles; H, Jueves; F, Viernes; A, Sábado; S, Domingo.

[Horas] 9:00-19:30

Ejemplo: acl horariolaboral time MTWHF 9:00-19:30

http\_access deny prensadepor horariolaboral

(Así se bloquea conectarse a la prensa deportiva en el horario laboral.)