

Reto Día 14: Configuración Estratégica de un Servidor Proxy con Políticas Dinámicas de Navegación en Linux

por Daniel Ariza

10/07/2025

Fase 1: Despliegue del servidor y configuración base

● Instalar Squid como Proxy HTTP/HTTPS en Linux.

PASO 1: Actualiza los paquetes del sistema

```
sudo apt update && sudo apt upgrade -y
```

PASO 2: Instala Squid

```
sudo apt install squid -y
```

Una vez instalado, el servicio Squid debería estar activo. Verifícalo con:

```
sudo systemctl status squid
```

```
daniel@ubuntucodearts:~$ sudo systemctl status squid
* squid.service - LSB: Squid HTTP Proxy version 3.x
   Loaded: loaded (/etc/init.d/squid; bad; vendor preset: enabled)
   Active: active (running) since vie 2025-07-11 10:19:06 CEST; 2min 54s ago
     Docs: man:systemd-sysv-generator(8)
   CGroup: /system.slice/squid.service
           └─ 1-3564 /usr/sbin/squid -YC -f /etc/squid/squid.conf
              1-3572 (squid-1) -YC -f /etc/squid/squid.conf
              1-3574 (logfile-daemon) /var/log/squid/access.log
              -3597 (pinger)
```

Debería mostrar algo como: **active (running)**. Si está bien, pulsa **q** para salir del estado.

PASO 3: Ubica el archivo de configuración principal

El archivo de configuración principal de Squid es:

/etc/squid/squid.conf

Haz una copia de seguridad antes de modificarlo:

sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.bak

PASO 4: Configuración básica como proxy

Editamos el archivo:

sudo nano /etc/squid/squid.conf

Busca y ajusta las siguientes líneas:

1. Puerto de escucha (por defecto es 3128, puedes dejarlo o cambiarlo):

http_port 3128

2. Permitir acceso local a tu red interna

Busca las líneas con **acl localnet** y asegúrate de añadir tu red, por ejemplo:

acl localnet src 192.168.100.1/24

Y luego busca la línea:

`http_access allow localnet`

También puedes comentar la línea que deniega todo si quieres pruebas abiertas:

`# http_access deny all`

```
GNU nano 2.5.3 Archivo: /etc/squid/squid.conf
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost
# And finally deny all other access to this proxy
#http_access deny all
```

Y en su lugar:

`http_access allow all`

(Solo para pruebas; en producción no se recomienda)

Guarda y cierra con CTRL+O, ENTER, luego CTRL+X.

PASO 5: Reinicia Squid

`sudo systemctl restart squid`

Y revisa si está escuchando en el puerto:

`sudo netstat -tulnp | grep squid`

```
daniel@ubuntucodearts:~$ sudo systemctl restart squid
daniel@ubuntucodearts:~$ sudo netstat -tulnp | grep squid
tcp        0      0 0.0.0.0:3128          0.0.0.0:*            ESCUCHAR    4460/(squid-1)
udp        0      0 0.0.0.0:39820        0.0.0.0:*            4460/(squid-1)
udp6       0      0 :::44937             :::*                  4460/(squid-1)
daniel@ubuntucodearts:~$
```

Significa que Squid está escuchando correctamente en todas las interfaces IPv4 en el puerto 3128.

● Configurar el puerto estándar (3128) y permitir solo la interfaz interna.

1. Editar el archivo de configuración de Squid

`sudo nano /etc/squid/squid.conf`

Asegúrate de que tenga lo siguiente:

Puerto y dirección de escucha (solo en la LAN):

`http_port 192.168.100.1:3128`

Esto limita Squid a escuchar solo en la IP de la LAN, no en todas.

Red interna autorizada:

Verifica o añade estas líneas:

`acl localnet src 192.168.100.0/24`

`http_access allow localnet`

`http_access deny all`

Esto permite únicamente a dispositivos de esa subred acceder al proxy, y bloquea el resto.

2. Guarda y reinicia Squid

`sudo systemctl restart squid`

3. Verifica que Squid esté escuchando solo en 192.168.100.1:3128

`sudo netstat -tulnp | grep squid`

```
daniel@ubuntu:~$ sudo systemctl restart squid
[sudo] password for daniel:
daniel@ubuntu:~$ sudo netstat -tulnp | grep squid
tcp        0      0 192.168.100.1:3128 0.0.0.0:*        LISTEN      4837/(squid-1)
udp        0      0 0.0.0.0:41396      0.0.0.0:*        4837/(squid-1)
udp6       0      0 :::37536           :::*              4837/(squid-1)
daniel@ubuntu:~$
```

Squid solo está escuchando en la IP interna 192.168.100.1.

En el puerto estándar 3128.

Ya no escucha en 0.0.0.0 ni en :::3128, así que está cerrado a conexiones externas. Perfecto para una red local segura

● Activar los logs y probar la conectividad Proxy básica desde un cliente.

1. Verificar que los logs están activados en Squid

Squid ya registra logs por defecto en:

Accesos (peticiones de clientes):

```
/var/log/squid/access.log
```

Errores o advertencias:

```
/var/log/squid/cache.log
```

Puedes verificar con:

```
ls -l /var/log/squid/
```

Y si quieres ver los últimos accesos en vivo:

```
sudo tail -f /var/log/squid/access.log
```

Déjalo abierto mientras haces la prueba desde el cliente (verás las peticiones aparecer en tiempo real).

El cliente que vamos a conectar es Windows Server Core.

En PowerShell:

```
netsh winhttp set proxy 192.168.100.1:3128
```

Luego prueba:

```
Invoke-WebRequest http://example.com
```

(Nota: Windows Server Core no tiene navegador, pero esto simula una petición HTTP)

```
PS C:\Users\daniel> netsh winhttp set proxy 192.168.100.1:3128
Configuración actual del proxy WinHTTP:

    Servidores proxy: 192.168.100.1:3128
    Lista de omisión  : (ninguna)

PS C:\Users\daniel> Invoke-WebRequest http://example.com
```

Fase 2: Creación de perfiles de navegación

- Definir 3 grupos de usuarios: desarrollo, administración y marketing.

1. Crear los grupos de usuarios

Ejecuta estos comandos en tu servidor:

```
sudo groupadd desarrollo
```

```
sudo groupadd administracion
```

```
sudo groupadd marketing
```

Puedes verificar que se han creado correctamente con:

getent group desarrollo administracion marketing

```
daniel@ubuntucodearts:~$ sudo groupadd desarrollo
daniel@ubuntucodearts:~$ sudo groupadd administracion
daniel@ubuntucodearts:~$ sudo groupadd marketing
daniel@ubuntucodearts:~$ getent group desarrollo administracion marketing
desarrollo:x:1011:
administracion:x:1012:
marketing:x:1013:
daniel@ubuntucodearts:~$ _
```

● Crear listas de control de acceso (ACL) específicas para cada grupo.

1. Crea archivos con las IPs por grupo

Vamos a guardar las IPs de cada grupo en archivos separados (esto es más limpio y escalable).

Crear directorio para las listas

```
sudo mkdir -p /etc/squid/acls
```

Crear archivo para cada grupo

```
sudo nano /etc/squid/acls/desarrollo.txt
```

Dentro de ese archivo pon, por ejemplo:

```
192.168.100.50
```

```
192.168.100.51
```

(estas serían las IPs que dnsmasq asigna al grupo desarrollo)

Repite para los otros grupos:


```
sudo nano /etc/squid/acls/administracion.txt
```

```
sudo nano /etc/squid/acls/marketing.txt
```

2. Editar el archivo de configuración de Squid

```
sudo nano /etc/squid/squid.conf
```

Añade estas líneas (puedes ponerlas al principio o cerca de las otras ACLs):

```
# ACL por grupo basado en IP
```

```
acl desarrollo src "/etc/squid/acls/desarrollo.txt"
```

```
acl administracion src
```

```
"/etc/squid/acls/administracion.txt"
```

```
acl marketing src "/etc/squid/acls/marketing.txt"
```

Ahora definimos reglas distintas. Ejemplo básico:

```
# Grupo desarrollo: acceso completo
```

```
http_access allow desarrollo
```

```
# Grupo administración: solo HTTP (bloqueamos HTTPS  
luego si queremos)
```

```
http_access allow administracion
```

Grupo marketing: solo sitios específicos (lo afinamos después)

http_access allow marketing

```
#ACL por grupo basado en IP
acl desarrollo src "/etc/squid/acls/desarrollo.txt"
acl administracion src "/etc/squid/acls/administracion.txt"
acl marketing src "/etc/squid/acls/marketing.txt"

#Grupo desarrollo acceso completo
http_access allow desarrollo

#Grupo administracion: solo HTTP (bloqueamos HTTPS luego si queremos)
http_access allow administracion

#Grupo marketing: solo sitios especifico (lo afinamos despues)
http_access allow marketing
```

Y al final, como siempre:

http_access deny all

3. Reinicia Squid

sudo systemctl restart squid

- **Configurar reglas para:**
 - **desarrollo: acceso total excepto sitios de ocio.**
 - **administración: solo navegación profesional.**
 - **marketing: acceso libre solo en horario de descanso (11:00 a 11:30 y 16:00 a 16:30).**

PASO 1: Crear listas de sitios

Creamos los archivos con dominios por categoría:

```
sudo mkdir -p /etc/squid/sites
```

Sitios de ocio (bloqueados para desarrollo)

```
sudo nano /etc/squid/sites/ocio.txt
```

Ejemplo:

```
facebook.com
```

```
youtube.com
```

```
tiktok.com
```

```
instagram.com
```

Sitios profesionales (permitidos solo a administración)

```
sudo nano /etc/squid/sites/profesionales.txt
```

Ejemplo:

linkedin.com

gov.es

ine.es

stackoverflow.com

PASO 2: Editar /etc/squid/squid.conf

```
sudo nano /etc/squid/squid.conf
```

- ♦ 1. Cargar ACLs de IP por grupo

(Ya lo habías hecho, pero por claridad):

```
acl desarrollo src "/etc/squid/acls/desarrollo.txt"
```

```
acl administracion src
```

```
"/etc/squid/acls/administracion.txt"
```

```
acl marketing src "/etc/squid/acls/marketing.txt"
```

- ♦ 2. ACLs de dominios y horarios

```
# Dominios de ocio
```

```
acl sitios_ocio dstdomain "/etc/squid/sites/ocio.txt"
```

```
# Dominios profesionales
```

```
acl sitios_profesionales dstdomain
```

```
"/etc/squid/sites/profesionales.txt"
```

Horarios permitidos para marketing

acl descanso_morning time MTWHF 11:00-11:30

acl descanso_evening time MTWHF 16:00-16:30

♦ 3. Reglas de acceso

DESARROLLO: todo menos ocio

http_access deny desarrollo sitios_ocio

http_access allow desarrollo

ADMINISTRACIÓN: solo sitios profesionales

http_access allow administracion sitios_profesionales

http_access deny administracion

MARKETING: acceso solo en horarios específicos

http_access allow marketing descanso_morning

http_access allow marketing descanso_evening

http_access deny marketing

```
#Dominios de ocio
acl sitios_ocio dstdomain "/etc/squid/sites/ocio.txt"

#Dominios profesionales
acl sitios_profesionales dstdomain "/etc/squid/sites/profesionales.txt"

#Horarios permitidos para marketing
acl descanso_morning time MTWHF 11:00-11:30
acl descanso_evening time MTWHF 16:00-16:30

#DESARROLLO: todo menos ocio
http_access deny desarrollo sitios_ocio
http_access allow desarrollo

#ADMINISTRACION: solo sitios profesionales
http_access allow administracion sitios_profesionales
http_access deny administracion

#MARKETING: acceso solo en horarios especificos
http_access allow marketing descanso_morning
http_access allow marketing descanso_evening
http_access deny marketing

#ACL por grupo basado en IP
acl desarrollo src "/etc/squid/acls/desarrollo.txt"
acl administracion src "/etc/squid/acls/administracion.txt"
acl marketing src "/etc/squid/acls/marketing.txt"

#Grupo desarrollo acceso completo
http_access allow desarrollo
```

Por defecto, denegar el resto

`http_access deny all`

PASO 3: Reinicia Squid

`sudo systemctl restart squid`

Fase 3: Control por tipo de contenido y comportamiento

- **Bloquear descargas de archivos .exe, .mp4, .zip desde cualquier perfil.**

PASO 1: Crear lista de extensiones prohibidas

Creamos un archivo con las extensiones de archivos que queremos bloquear:

```
sudo nano /etc/squid/sites/archivos_prohibidos.txt
```

Añade lo siguiente:

```
\.exe$
```

```
\.mp4$
```

```
\.zip$
```

Es importante usar las barras invertidas (\) porque se trata de expresiones regulares (regex) y el \$ indica que debe estar al final del URL.

PASO 2: Añadir la ACL al archivo squid.conf

```
sudo nano /etc/squid/squid.conf
```

Añade esta línea junto a tus otras ACLs:

```
acl archivos_prohibidos url_regex -i  
"/etc/squid/sites/archivos_prohibidos.txt"
```

- `url_regex`: busca coincidencias en la URL solicitada.
- `-i`: es para ignorar mayúsculas/minúsculas.

Y luego antes de cualquier `http_access allow`, añade:

```
http_access deny archivos_prohibidos
```

Ejemplo completo:

```
# BLOQUEO GLOBAL DE DESCARGAS
```

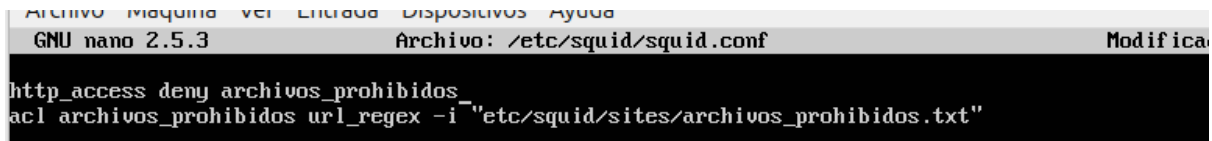
```
acl archivos_prohibidos url_regex -i  
"/etc/squid/sites/archivos_prohibidos.txt"
```

```
http_access deny archivos_prohibidos
```

```
# Luego van tus reglas por grupo...
```


`http_access deny desarrollo sitios_ocio`

`http_access allow desarrollo`



```
GNU nano 2.5.3 Archivo: /etc/squid/squid.conf
http_access deny archivos_prohibidos
acl archivos_prohibidos url_regex -i "/etc/squid/sites/archivos_prohibidos.txt"
```

Importante: Squid procesa las reglas en orden, así que la denegación debe ir antes de permitir nada.

PASO 3: Reiniciar Squid

`sudo systemctl restart squid`

● Restringir sitios con contenido multimedia (YouTube, Netflix, Twitch).

PASO 1: Crear lista de dominios multimedia

Creamos un archivo con los dominios a bloquear:

`sudo nano /etc/squid/sites/multimedia.txt`

Añade los principales dominios usados por estos servicios:

`youtube.com`

`yting.com`

`googlevideo.com`

`netflix.com`

nflxing.net

nflxvideo.net

twitch.tv

ttvnw.net

(Estos incluyen dominios auxiliares necesarios para cargar los videos)

PASO 2: Crear la ACL en squid.conf

Abre el archivo:

```
sudo nano /etc/squid/squid.conf
```

Añade la ACL y la regla antes de cualquier allow:

```
# BLOQUEO DE SITIOS MULTIMEDIA
```

```
acl multimedia dstdomain
```

```
"/etc/squid/sites/multimedia.txt"
```

```
http_access deny multimedia
```

Esta regla se aplicará a todos los perfiles (grupos) porque va antes de cualquier http_access allow.

Ejemplo ordenado:

```
acl archivos_prohibidos url_regex -i  
"/etc/squid/sites/archivos_prohibidos.txt"  
  
http_access deny archivos_prohibidos
```

```
acl multimedia dstdomain  
"/etc/squid/sites/multimedia.txt"  
  
http_access deny multimedia
```



```
GNU nano 2.5.3 Archivo: /etc/squid/squid.conf Modifica  
#Bloqueo de sitios multimedia  
acl multimedia dstdomain "/etc/squid/sites/multimedia.txt"  
http_access deny multimedia_
```

PASO 3: Reiniciar Squid

```
sudo systemctl restart squid
```

● Aplicar una política de "sitios aprobados" (whitelist) para administración.

1. Editar squid.conf

```
sudo nano /etc/squid/squid.conf
```

Asegúrate de tener (o añade si falta):

```
# ACL de IPs del grupo administración
```

```
acl administracion src
```

```
"/etc/squid/acls/administracion.txt"
```

```
# ACL de sitios aprobados (whitelist)
```

```
acl sitios_profesionales dstdomain
```

```
"/etc/squid/sites/profesionales.txt"
```

Y ahora define la política específica ANTES de cualquier `http_access` allow administración general:

```
# ADMINISTRACIÓN: solo acceso a sitios aprobados
```

```
http_access allow administracion sitios_profesionales
```

```
http_access deny administracion
```

```
#ADMINISTRACION: solo sitios profesionales
http_access allow administracion sitios_profesionales
http_access deny administracion
```

Esto funciona así:

- Permite solo si la IP está en el grupo "administracion" y el destino está en la whitelist.
- Todo lo demás que intente ese grupo será bloqueado.

2. Reiniciar Squid

```
sudo systemctl restart squid
```

Fase 4: Rendimiento y análisis

- **Configurar caché local para acelerar la carga de sitios web frecuentes.**

PASO 1: Editar el archivo de configuración de Squid

```
sudo nano /etc/squid/squid.conf
```

Busca las siguientes directivas (o añádelas si no existen):

1. Directorio de caché en disco

```
cache_dir ufs /var/spool/squid 1000 16 256
```

- **ufs:** tipo de almacenamiento.
- **/var/spool/squid:** ubicación predeterminada del caché.

- **1000**: tamaño máximo en MB (ajústalo si quieres más o menos).
- **16 y 256**: estructura de subdirectorios (déjalo así).

2. Tamaño máximo de archivos a cachear

maximum_object_size 10 MB

Evita almacenar archivos muy grandes (como .iso, .mp4, etc.).

3. Tamaño mínimo (opcional)

minimum_object_size 1 KB

Evita guardar cosas demasiado pequeñas (como cookies).

4. Caché en memoria RAM

cache_mem 256 MB

Tamaño reservado en RAM para archivos frecuentemente accedidos (sube esto si tienes RAM disponible).

```
cache_dir ufs /var/spool/squid 1000 16 256
maximum_object_size 10 MB
minimum_object_size 1 KB
cache_mem 256 MB
```

PASO 2: Crear y preparar la caché (solo una vez)

Si es la primera vez que usas caché en disco, ejecuta:

sudo squid -z

Esto crea la estructura de subdirectorios en
`/var/spool/squid`.

```
daniel@ubuntucodearts:~$ sudo squid -z
2025/07/11 14:49:47| ERROR: Can not open file etc/squid/sites/archivos_prohibidos.txt for reading
2025/07/11 14:49:47| Warning: empty ACL: acl archivos_prohibidos url_regex -i "etc/squid/sites/archivos_prohibidos.txt"
2025/07/11 14:49:47| aclIpParseIpData: WARNING: Netmask masks away part of the specified IP in '192.168.100.1/24'
2025/07/11 14:49:47| Squid is already running! Process ID 6416
daniel@ubuntucodearts:~$ sudo squid -z
2025/07/11 14:50:14| ERROR: Can not open file etc/squid/sites/archivos_prohibidos.txt for reading
2025/07/11 14:50:14| Warning: empty ACL: acl archivos_prohibidos url_regex -i "etc/squid/sites/archivos_prohibidos.txt"
2025/07/11 14:50:14| aclIpParseIpData: WARNING: Netmask masks away part of the specified IP in '192.168.100.1/24'
2025/07/11 14:50:14| Squid is already running! Process ID 6416
daniel@ubuntucodearts:~$ _
```

PASO 3: Reiniciar el servicio Squid

`sudo systemctl restart squid`

Y comprueba que se está ejecutando bien:

`sudo systemctl status squid`

● Ajustar tamaño de la memoria caché RAM y disco.

Esto lo realizamos en el paso anterior.

- **Analizar los logs de navegación** (`/var/log/squid/access.log`) e identificar patrones de tráfico.

Consultar navegación en tiempo real

`sudo tail -f /var/log/squid/access.log`

```
daniel@ubuntucodearts:~$ sudo tail -f /var/log/squid/access.log
1752230787.006 265 192.168.100.65 TCP_TUNNEL/200 4476 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230787.591 244 192.168.100.65 TCP_TUNNEL/200 4818 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230788.023 353 192.168.100.65 TCP_TUNNEL/200 4476 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230788.425 370 192.168.100.65 TCP_TUNNEL/200 4505 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230788.703 249 192.168.100.65 TCP_TUNNEL/200 4483 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230789.150 374 192.168.100.65 TCP_TUNNEL/200 4483 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230789.459 280 192.168.100.65 TCP_TUNNEL/200 4483 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230789.764 274 192.168.100.65 TCP_TUNNEL/200 4483 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230790.158 365 192.168.100.65 TCP_TUNNEL/200 4499 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
1752230790.432 245 192.168.100.65 TCP_TUNNEL/200 4498 CONNECT settings-win.data.microsoft.com:443
- HIER_DIRECT/4.231.128.59 -
```

INTERPRETACIÓN

- Se trata de tráfico **automático del sistema operativo Windows**, no navegación humana.
- El patrón es muy frecuente, con múltiples conexiones por segundo.
- Todas las conexiones son **tuneleadas (HTTPS)**, por eso Squid solo puede ver el destino **del túnel**, no el contenido.