***FASE1:***

**1. Instalación de Squid**

1. Actualizar lista de paquetes:

*sudo apt update*

2. Instalar Squid:

*sudo apt install -y squid*

**2. Copia de seguridad de la configuración original**

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

**3. Configuración básica**

Editamos  */etc/squid/squid.conf:*

http\_port 3128

acl localnet src 192.168.1.0/24

http\_access allow localhost

http\_access allow localnet

http\_access deny all

**4. Reiniciar el servicio**

*sudo systemctl restart squid*

Verificamos

*sudo ss -tunlp | grep squid*

**5. Prueba desde un cliente**

*curl -x http://192.168.1.201:3128 -I http://example.com*

Nos devuelve: HTTP/1.1 200 OK.

***FASE 2:***

## **Requisitos**

Se definieron los siguientes grupos de IPs:

- Desarrollo: 192.168.1.151–192.168.1.159

- Administración: 192.168.1.161–192.168.1.169

- Marketing: 192.168.1.171–192.168.1.179

## **Definición de ACLs**

En /etc/squid/squid.conf se añadieron las siguientes líneas:

·  *acl desarrollo src 192.168.1.151-192.168.1.159*

*· acl administracion src 192.168.1.161-192.168.1.169*

*· acl marketing src 192.168.1.171-192.168.1.179*

*· acl ocio dstdomain facebook.com twitter.com netflix.com*

*· acl sitios\_profesionales url\_regex -i \.intranet\.empresa\.com \.erp\.empresa\.local*

*· acl horario\_marketing time M T W H F 11:00-11:30 16:00-16:30*

## **Reglas de acceso**

Después de las ACL, se configuró el orden de las reglas http\_access:

· *http\_access allow desarrollo !ocio*

*· http\_access deny desarrollo*

*· http\_access allow administracion sitios\_profesionales*

*· http\_access deny administracion*

*· http\_access allow marketing horario\_marketing*

*· http\_access deny marketing*

*· http\_access allow localhost*

*· http\_access allow localnet*

*· http\_access deny all*

**Comprobamos en cliente**

# sitio permitido

***curl -x http://192.168.1.201:3128 -I http://intranet.empresa.com***

# sitio bloqueado

***curl -x http://192.168.1.201:3128 -I http://example.com***

## **Errores detectados y soluciones**

· Uso de url\_regex con subdominios:

Se usó inicialmente 'acl ocio url\_regex -i \.facebook\.com', pero Squid rechazó el patrón al considerar '.facebook.com' un subdominio redundante de 'facebook.com'. Se cambió a 'acl ocio dstdomain facebook.com' para corregirlo.

· Orden incorrecto de http\_access:

Las reglas de denegación estaban después de 'allow localnet', por lo que nunca se aplicaban. Se reordenaron para situar todas las reglas de grupo antes de 'allow localnet'.

***FASE 3:***

**Configuración**

1. Definir ACLs:

- *forbidden\_ext: extpath\_regex -i \.exe$ \.mp4$ \.zip$*

*- multimedia: dstdomain youtube.com netflix.com twitch.tv*

*- approved\_admin: dstdomain intranet.empresa.com erp.empresa.local*

2. Aplicar reglas http\_access (en este orden):

*http\_access deny forbidden\_ext*

*http\_access deny multimedia*

*http\_access deny !approved\_admin administracion*

*http\_access allow administracion approved\_admin*

*http\_access allow localhost*

*http\_access allow localnet*

*http\_access deny all*

**Pruebas realizadas**

- *curl -x http://192.168.1.201:3128 -I http://example.com/file.exe* → 403 Forbidden

- *curl -x http://192.168.1.201:3128 -I http://youtube.com* → 403 Forbidden

- *curl -x http://192.168.1.201:3128 -I http://intranet.empresa.com* → 503

ERR\_DNS\_FAIL (no existe DNS) ⚠️

- *curl -x http://192.168.1.201:3128 -I http://facebook.com* → 403 Forbidden (ocio)

**Errores y soluciones**

• Al definir la ACL multimedia con subdominios duplicados ('.youtube.com' y 'youtube.com') → Squid falla al parsear.

Solución: usar sólo 'dstdomain youtube.com netflix.com twitch.tv'.

• 503 ERR\_DNS\_FAIL al probar intranet.empresa.com → no existe resolución DNS.

Aviso: es correcto (no hay servidor DNS interno para ese nombre).

**FASE 4:**

**1. Configuración de IPs estáticas:**

- Se asignaron direcciones fijas en cada cliente

- Verificación con `*ip a*` para confirmar la IP.

2. Creación de archivo de contraseñas NCSA:

*sudo htpasswd -c /etc/squid/passwd bob*  para usuario 'bob'

- Ajustar permisos:

*sudo chown proxy:proxy /etc/squid/passwd*

3. Configuración de autenticación en Squid *(/etc/squid/squid.conf):*

*auth\_param basic program /usr/lib/squid/basic\_ncsa\_auth /etc/squid/passwd  
 auth\_param basic realm "Proxy Corporativo"  
 acl auth\_users proxy\_auth REQUIRED  
 http\_access allow auth\_users*

4. Recarga y comprobación:

*sudo squid -k reconfigure*  
   
   
**Verificación final**

En cliente:  
  
 *curl -I -x http://192.168.1.201:3128 -U bob* [*http://example.com*](http://example.com)

Pide credenciales y devuelve *HTTP/1.1 200 OK*.  
  
 Confirma acceso según usuario y rol.

***FASE 5:***

**1. Caché en memoria RAM**

• Editar el fichero de configuración:

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

*sudo nano /etc/squid/squid.conf*

• Bajo “MEMORY CACHE OPTIONS”, agregar:

*cache\_mem 256 MB*

• Guardar y reiniciar:

*sudo systemctl restart squid*

*sudo squid -k parse* # verificar sintaxis

**2. Caché en disco**

• En el mismo fichero (bajo “DISK CACHE OPTIONS”), ajustar:

cache\_dir ufs /var/spool/squid 20000 16 256 # 20 GB de caché

• Inicializar caché:

*sudo squid -z*

• Reiniciar y recargar:

*sudo systemctl restart squid*

*sudo squid -k reconfigure*

**3. Análisis de registros de tráfico**

• Ver últimas peticiones:

*sudo tail -n 15 /var/log/squid/access.log*

• Top 5 clientes (IPs) más activos:

*awk '{ print $3 }' /var/log/squid/access.log |*

*sort | uniq -c | sort -nr | head -n 5*

• Top 5 URLs más solicitadas:

*awk '{ print $7 }' /var/log/squid/access.log |*

*sort | uniq -c | sort -nr | head -n 5*

• Distribución de códigos de respuesta:

*awk '{ print $4 }' /var/log/squid/access.log |*

*sort | uniq -c | sort -nr*