

Configuración estratégica de un servidor proxy con políticas dinámicas de navegación en Linux

1. Despliegue del servidor y configuración base.

- **Instalar Squid:** Como Proxy HTTP/HTTPS en Linux
- **Configurar el puerto estándar:** El 3128 y permitir solo la interfaz interna.
- **Activar los logs y probar la conectividad proxy básica desde un cliente.**

2. Creación de perfiles de navegación.

- **Definir 3 grupos de usuarios:** Desarrollo, administración y marketing.
- **Crear listas de control de acceso (ACL):** Específicas para cada grupo.
- **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)

3. Control por tipo de contenido y comportamiento.

- **Bloquear descargas de archivos:** .exe, .mp4, .zip desde cualquier perfil.
- **Restringir sitios con contenido multimedia:** Youtube, Netflix, Twitch.
- **Aplicar una política de sitios aprobados (whitelist):** Para administración.

4. Gestión de acceso por IP y usuarios.

- **Asociar direcciones IP estáticas a los usuarios por grupo.**
- **Activar autenticación básica por usuario:** (Archivo /etc/squid/passwd).
- **Verificar que cada usuario accede según su rol y restricciones.**

5. Rendimiento y análisis.

- **Configurar caché local:** Para acelerar la carga de sitios web frecuentes.
 - **Ajustar tamaño de la memoria caché RAM y disco.**
 - **Analizar los logs de navegación:** (/var/log/squid/access.log) e identificar patrones de tráfico.
-

1. Despliegue del servidor y configuración base.

Para instalar Squid lo hacemos con el comando:

```
sudo apt update
```

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

El puerto 3128 es el estándar con el que viene configurado ya, por lo que ahora vamos a permitir solo la interfaz interna. Para ello modificamos el archivo

`/etc/squid/squid.conf` y añadimos las siguientes líneas:

```
acl red_interna src 192.168.100.0/24
```

```
http_access allow red_interna
```

```
http_access deny all
```



```
GNU nano 7.2 /etc/squid/squid.conf *
http_access deny to_linklocal

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*.conf

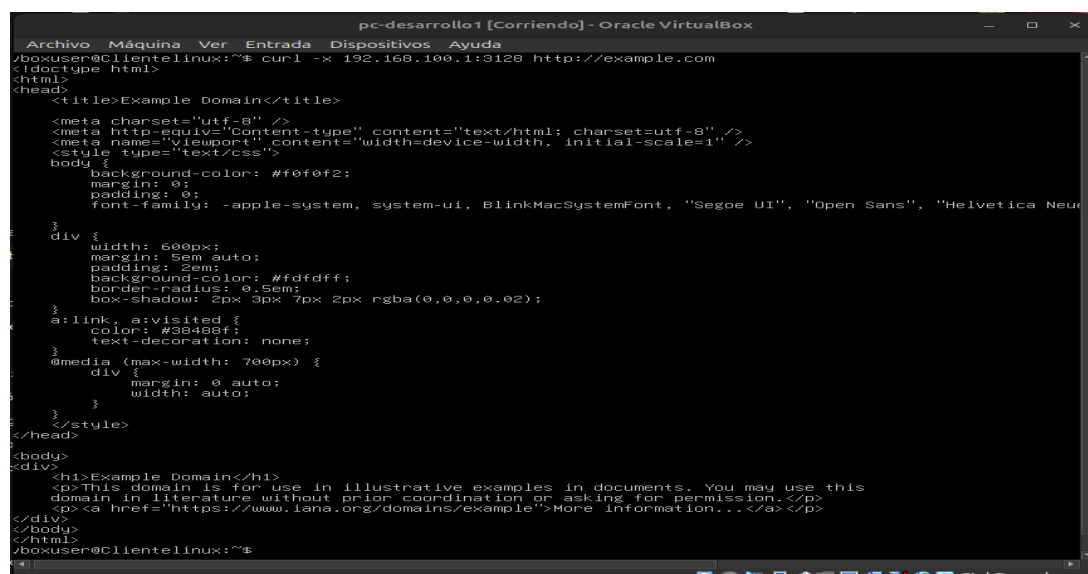
# For example, to allow access from your local networks, you may uncomment the
# following rule (and/or add rules that match your definition of "local"):
# http_access allow localnet
http_access allow red_interna
# And finally deny all other access to this proxy
http_access deny all

# TAG: adapted http access
# Allowing or Denying access based on defined access lists
#
# Essentially identical to http_access, but runs after redirectors
# and ICAP/eCAP adaptation. Allowing access control based on their
# output.

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Ahora desde una máquina cliente probaremos la conectividad proxy con el comando:

```
curl -x 192.168.100.1:3128 http://example.com
```



```
pc-desarrollo1 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
/boxuser@Clientlinux:~$ curl -x 192.168.100.1:3128 http://example.com
<doctype html>
<html>
<head>
<title>Example Domain</title>
<meta charset="utf-8" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style type="text/css">
body {
background-color: #f0f0f2;
margin: 0;
padding: 0;
font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue";
}
div {
width: 600px;
margin: 5em auto;
padding: 2em;
background-color: #fdfdff;
border-radius: 0.5em;
box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
color: #38488f;
text-decoration: none;
}
@media (max-width: 700px) {
div {
margin: 0 auto;
width: auto;
}
}
</style>
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
<p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
/boxuser@Clientlinux:~$
```

Y en el servidor verificaremos los logs con el comando:
sudo tail -f /var/log/squid/access.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo tail -f /var/log/squid/access.log
1752057380.506      355 192.168.100.32 TCP_MISS/200 1616 GET http://example.com/ - HIER_DIRECT/96.7.128.198 text/html
```

2. Creación de perfiles de navegación.

Para definir 3 grupos de usuario, primero creamos la carpeta de grupos con el comando:

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

Y ahora creamos 3 archivos dentro de la carpeta con los comandos:

```
sudo nano /etc/squid/grupos/desarrollo.txt
sudo nano /etc/squid/grupos/administracion.txt
sudo nano /etc/squid/grupos/marketing.txt
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo nano /etc/squid/grupos/desarrollo.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo nano /etc/squid/grupos/administracion.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo nano /etc/squid/grupos/marketing.txt
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

Y dentro de cada archivo le asignamos la dirección ip que asignamos anteriormente en el archivo **dnsmasq.conf**. En el grupo de marketing le asignamos la dirección que añadimos al departamento de diseño, ya que pertenecen al mismo departamento que es el de diseño y marketing.

```
GNU nano 7.2 /etc/squid/grupos/marketing.txt
192.168.100.42
```

Ahora para crear listas de control de acceso ACL, creamos una carpeta llamada listas con el comando:

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

Y dentro de ella creamos dos archivos .txt con los nombres **ocio.txt** y **no_profesionales.txt**. Ahora dentro de cada archivo introducimos lo que corresponde al mismo:

Ocio.txt	no_profesionales.txt
.facebook.com .youtube.com .netflix.com .instagram.com .tiktok.com	.facebook.com .youtube.com .netflix.com .twitch.tv .reddit.com

Lo siguiente es modificar el archivo **squid.conf** y añadir las líneas:

```
# === ACL por IP/grupo ===
acl desarrollo src "/etc/squid/grupos/desarrollo.txt"
acl administracion src "/etc/squid/grupos/administracion.txt"
acl marketing src "/etc/squid/grupos/marketing.txt"

# === ACL por dominios bloqueados ===
acl ocio dstdomain "/etc/squid/listas/ocio.txt"
acl no_profesionales dstdomain "/etc/squid/listas/no_profesionales.txt"

# === ACL por horario de descanso ===
acl descanso_morning time MTWHF 11:00-11:30
acl descanso_evening time MTWHF 16:00-16:30
```



```
GNU nano 7.2 /etc/squid/squid.conf
#
acl javascript rep_mime_type -i "application/x-javascripts"
#
#Default:
# ACLs all, manager, localhost, to_localhost, to_linklocal, and CONNECT are predefined.
# === ACL por IP/grupo ===
acl desarrollo src "/etc/squid/grupos/desarrollo.txt"
acl administracion src "/etc/squid/grupos/administracion.txt"
acl marketing src "/etc/squid/grupos/marketing.txt"
# === ACL por dominios bloqueados ===
acl ocio dstdomain "/etc/squid/listas/ocio.txt"
acl no_profesionales dstdomain "/etc/squid/listas/no_profesionales.txt"
# === ACL por horario de descanso ===
acl descanso_morning time MTWHF 11:00-11:30
acl descanso_evening time MTWHF 16:00-16:30
#
# Recommended minimum configuration:
#
#G Help      #O Write Out  #W Where Is   #K Cut        #T Execute    #C Location   #U Undo       #A Set Mark
#X Exit      #R Read File  #N Replace    #U Paste      #J Justify    #_ Go To Line #E Redo       #G Copy
```

Y las reglas de acceso final por grupo:

```
# --- Grupo desarrollo: acceso total excepto ocio ---
http_access deny ocio desarrollo
http_access allow desarrollo

# --- Grupo administracion: solo navegación profesional ---
http_access deny no_profesionales administracion
http_access allow administracion

# --- Grupo marketing: acceso solo en horario de descanso ---
http_access allow marketing descanso_morning
http_access allow marketing descanso_evening
http_access deny marketing
```



```
GNU nano 7.2 /etc/squid/squid.conf
# following rule (and/or add rules that match your definition of "local"):
# http_access allow localnet
http_access allow red_interna
# --- Grupo desarrollo: acceso total excepto ocio ---
http_access deny ocio desarrollo
http_access allow desarrollo
# --- Grupo administracion: solo navegación profesional ---
http_access deny no_profesionales administracion
http_access allow administracion
# --- Grupo marketing: acceso solo en horario de descanso ---
http_access allow marketing descanso_morning
http_access allow marketing descanso_evening
http_access deny marketing
# And finally deny all other access to this proxy
http_access deny all
#G Help      #O Write Out  #W Where Is   #K Cut        #T Execute    #C Location   #U Undo       #A Set Mark
#X Exit      #R Read File  #N Replace    #U Paste      #J Justify    #_ Go To Line #E Redo       #G Copy
```

Y el archivo cierra con: **http_access deny all** para denegar todo lo demás.

Y reiniciamos el servicio para aplicar los cambios con:

```
sudo systemctl restart squid
```

3. Control por tipo de contenido y comportamiento.

Para bloquear las descargas de archivos .exe, .mp4, .zip desde cualquier perfil, lo hacemos creando el archivo:

```
sudo nano /etc/squid/listas/archivos_bloqueados.txt
```

Y añadiendo las líneas:

```
\.exe$
```

```
\.mp4$
```

```
\.zip$
```

Ahora para el bloqueo de sitios multimedia, lo hacemos parecido, creando el archivo:

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

Y añadiendo las líneas:

```
.youtube.com
```

```
.netflix.com
```

```
.twitch.tv
```

```
.vimeo.com
```

```
.soundcloud.com
```

Lo siguiente es crear una **Whitelist** para administración. Se hace igual, creando el archivo:

```
sudo nano /etc/squid/listas/whitelist_administracion.txt
```

Y añadiendo las líneas:

```
.intranet.empresa.com
```

```
.google.com
```

```
.wikipedia.org
```

```
.stackoverflow.com
```

```
.ubuntu.com
```

Ahora que ya tenemos las listas creadas, modificamos el archivo **squid.conf** y añadimos las siguientes líneas:

```
# === ACL de tipo de archivo prohibido ===
```

```
acl archivos_prohibidos url_regex "/etc/squid/listas/archivos_bloqueados.txt"
```

```
# === ACL de sitios multimedia ===
```

```
acl multimedia dstdomain "/etc/squid/listas/multimedia.txt"
```

```
# === Whitelist para administración ===
```

```
acl whitelist_admin dstdomain "/etc/squid/listas/whitelist_administracion.txt"
```

Y ahora le añadimos las siguientes reglas de control, que deben de ir antes del **http_access allow administracion** que añadimos anteriormente ya que es una regla general y permitiría el tráfico por encima de las reglas específicas. Las líneas son:

```
# Bloquear archivos peligrosos para todos
http_access deny archivos_prohibidos
```

```
# Bloquear multimedia para todos los perfiles
http_access deny multimedia
```

```
# Aplicar whitelist a administración (bloquea todo lo demás)
http_access allow administracion whitelist_admin
http_access deny administracion
```

Y reiniciamos el servicio para que se apliquen las reglas.

4. Gestión de acceso por IP y usuarios.

Ya hemos asociado las direcciones ip estáticas a los usuarios por grupos anteriormente. Ahora vamos a activar la autenticación básica por usuario. Para ello creamos el archivo de contraseñas con cada usuario y le asignamos una contraseña a cada uno:

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$ sudo htpasswd -c /etc/squid/passwd user_dev
sudo htpasswd /etc/squid/passwd user_admin
sudo htpasswd /etc/squid/passwd user_mkt
New password:
Re-type new password:
Adding password for user user_dev
New password:
Re-type new password:
Adding password for user user_admin
New password:
Re-type new password:
Adding password for user user_mkt
AZDINFARISSE@SVR-BASE-AZDINFARISSE:~$
```

Ahora modificamos el archivo **squid.conf** y añadimos las siguientes líneas:

```
# Autenticación básica
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic realm ProxyUsuarios
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

```
# ACLs de usuarios por nombre
acl usuario_dev proxy_auth user_dev
acl usuario_admin proxy_auth user_admin
acl usuario_mkt proxy_auth user_mkt
```

Y añadimos también las reglas por usuario con IP:

```
# ACLs de IPs por grupo
acl ip_dev src 192.168.100.32
acl ip_admin src 192.168.100.52
acl ip_mkt src 192.168.100.42
```

```
# Combinar IP + usuario
http_access allow usuario_dev ip_dev
http_access allow usuario_admin ip_admin
http_access allow usuario_mkt ip_mkt
```

Ahora desde el cliente administración intentamos acceder a google.com con el usuario `user_admin` y la contraseña asignada, y lo hacemos con el comando:

```
curl -x 192.168.100.1:3128 -U user_admin:Contraseña http://youtube.com
```

y en el cliente con el comando `sudo tail -f /var/log/squid/access.log` podemos ver los logs de acceso y saber si se deniegan o se admiten.



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/squid/listas$ sudo tail -f /var/log/squid/access.log
1752057380.506 355 192.168.100.32 TCP_MISS/200 1616 GET http://example.com/ - HIER_DIRECT/96.7.128.198 text/html
1752135600.103 0 192.168.100.32 TCP_DENIED/407 3953 GET http://google.com/ - HIER_NONE/- text/html
1752135654.201 0 192.168.100.32 TCP_DENIED/407 3957 GET http://youtube.com/ - HIER_NONE/- text/html
1752135661.047 0 192.168.100.32 TCP_DENIED/407 3953 GET http://google.com/ - HIER_NONE/- text/html
1752135674.260 0 192.168.100.32 TCP_DENIED/407 3981 GET http://stackoverflow.com/ - HIER_NONE/- text/html
1752135883.800 153 192.168.100.32 TCP_MISS/301 895 GET http://google.com/ user_dev HIER_DIRECT/172.217.17.14 text/html
```

En la imagen se puede ver que en las líneas 2, 3, 4 y 5 me está denegando el acceso debido a que no me había logueado en el cliente con el usuario correspondiente. Una vez me loguee correctamente me permitió el acceso.

5. Rendimiento y análisis.

Para aumentar el tamaño de la caché en el archivo **squid.conf** añadimos las siguientes líneas:

```
cache_mem 512 MB
cache_dir ufs /var/spool/squid 2000 16 256
```

Y reiniciamos el servicio para que se aplique.

Ahora para analizar los logs de navegación lo hacemos con el comando:
sudo tail -f /var/log/squid/access.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/etc/squid/grupos$ sudo tail -f /var/log/squid/access.log
1752220372.034      0 192.168.100.1 NONE NONE/400 3866 - / - HIER NONE/- text/html
1752220372.034      17 192.168.100.52 TCP_MISS/400 3964 GET http://192.168.100.1:3128/ user_admin HIER_DIRECT/192.168.100.1 text/html
1752220372.220     106 192.168.100.52 TCP_MISS/301 495 GET http://youtube.com/ user_admin HIER_DIRECT/142.250.200.78 application/binary
1752220651.730      0 192.168.100.1 NONE NONE/400 3866 - / - HIER NONE/- text/html
1752220651.730      17 192.168.100.52 TCP_MISS/400 3964 GET http://192.168.100.1:3128/ user_admin HIER_DIRECT/192.168.100.1 text/html
1752220651.880      72 192.168.100.52 TCP_MISS/301 495 GET http://youtube.com/ user_admin HIER_DIRECT/142.250.200.78 application/binary
1752220691.472      0 192.168.100.42 NONE NONE/400 3867 - / - HIER NONE/- text/html
```

Como se puede observar en la foto, el usuario admin ha accedido a <http://youtube.com> y el acceso le ha sido concedido. Por otro lado podemos observar en la última línea que la máquina cliente con ip terminada en .42, que es la máquina del usuario mkt ha intentado acceder a internet, pero se le ha denegado la conexión por no estar logueado con sus credenciales.