SAD - U4.2. Seguridad en aplicaciones

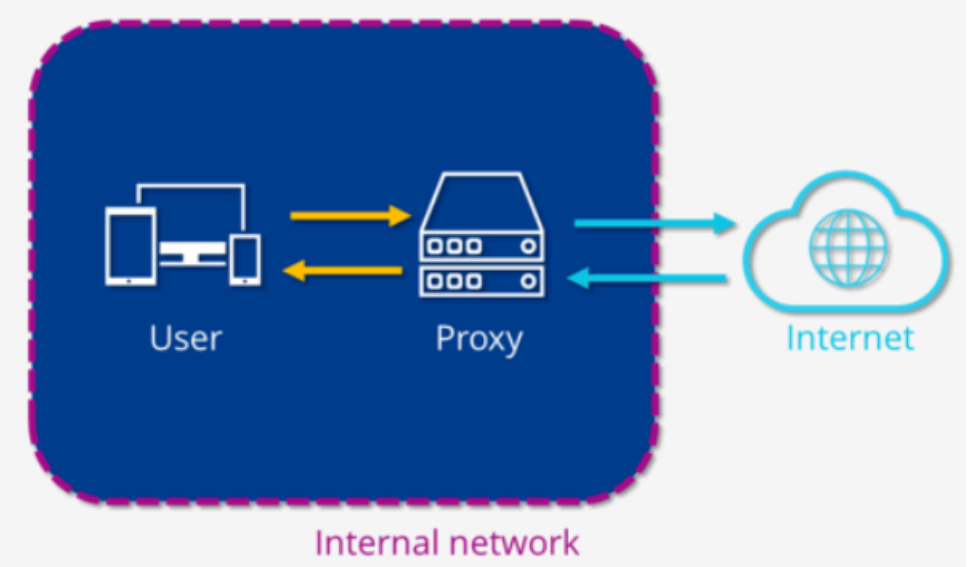
Descargar estos apuntes

Índice

1. ¿Qué es un servidor proxy?

Es un **agente intermediario** que realiza una acción en representación de otro. Intercepta las conexiones de red hechas desde un cliente a un servidor de destino.

- **Funcionamiento**: A solicita un recurso de C mediante B



https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-proxy-inverso/.

1.1. Proxy como firewall de nivel de aplicación

Los servidores proxy actúan sobre la **capa de aplicación (nivel 7)** del modelo OSI.

- Puede entender ciertas aplicaciones y protocolos de nivel 7 (por ejemplo, protocolo de transferencia de archivos o **FTP**, **DNS** o **HTTP**).
- **Permite detectar si un protocolo no deseado está funcionando por un puerto no estándar** o si se abusa de un protocolo de forma perjudicial, gracias a que entiende los protocolos de nivel de aplicación.
- Es mucho **más seguro y fiable** comparado con cortafuegos de filtrado de paquetes, ya que repercute sobre las 7 capas del modelo OSI.
- Es similar a un cortafuegos de filtrado de paquetes, pero también permite **filtrar por el contenido del paquete**, por ejemplo, para encontrar un archivo de malware en una respuesta http.

1.2 Ventajas

Mejora la seguridad

- Puede **bloquear tráfico malicioso**, ya que analiza el contenido del paquete, por ejemplo, las peticiones HTTP, para observar si se hace la descarga de un archivo ejecutable.
- **Bloquea** el acceso a sitios no deseados, por ejemplo, examinando la URL de destino.
- Puede hacer **Filtrado de contenidos/Control de contenidos** que no queremos que los usuarios accedan, como por ejemplo, contenido
- **Restringe el acceso a Internet** para solo unos determinados puertos (el resto quedarán cerrados).
- **Oculta la identidad** y da privacidad de los equipos de la red interna a nivel de red (solo un equipo se comunica con Internet).
    - Puede dar servicio a muchos usuarios para solicitar a través de él contenidos web (Ocultación IP / anonimato).
    - La petición del host NO va destinada al servidor final si no que se hace a la IP del proxy. Este será el que hará la petición a Internet.
    - Tampoco hay enrutamiento ni acceso directo como en el caso del firewall de nivel 3 y 4.

Acelera la navegación por Internet

- Los usuarios de la red local navegan más rápido, ya que puede guardar en **memoria caché** los recursos que se han descargado previamente de Internet.
- Utilización más eficiente de la conexión a Internet.
- Aumento y mejora de la velocidad de respuesta a peticiones.

Registra la actividad de la red

Se guarda en los logs la actividad hacia Internet u otras redes. Así podemos extraer estadísticas de uso del acceso a Internet, y tomar decisiones basadas en datos, por ejemplo:

- Saber qué sitios son más visitados, así guardar en caché todo lo que se pueda
- Saber qué sitios se visitan y no se quiere permitir
- Saber cuál es el periodo que se utiliza más el acceso a Internet
- Saber el ancho de banda usado por cada usuario y si hay abuso...

1.3 Inconvenientes

Posibles contenidos desactualizados

- **Puede servir a los clientes contenidos desactualizados de la memoria caché**. Esto se mejora comprobando la fecha de caducidad de los recursos en la memoria caché.

Puede impedir la realización de operaciones de red avanzadas

- Algunos protocolos no soportan el uso de proxy, por ejemplo el protocolo SIP de VoIP.

Posible violación de la intimidad

El proxy actúa como un "man-in-the-middle" de los usuarios de la red local.

- Almacenar las páginas y objetos que los usuarios solicitan puede suponer una **violación de la intimidad** para algunas personas.
- Hay que establecer en las políticas de la empresa que se está monitorizando el tráfico de los usuarios.

Posible punto de congestión

Como tiene que gestionar mucho tráfico de red hacia el exterior, esto puede hacer que en determinados momentos sea un cuello de botella.

- Entonces, hay que dimensionar bien este equipo en memoria RAM y disco.
- Utilizar el balanceo de carga, que sería un clúster de servidores proxy repartiéndose el trabajo.

1.4. Tipos de servidores proxy

Podemos clasificar los tipos de proxy según su función en:

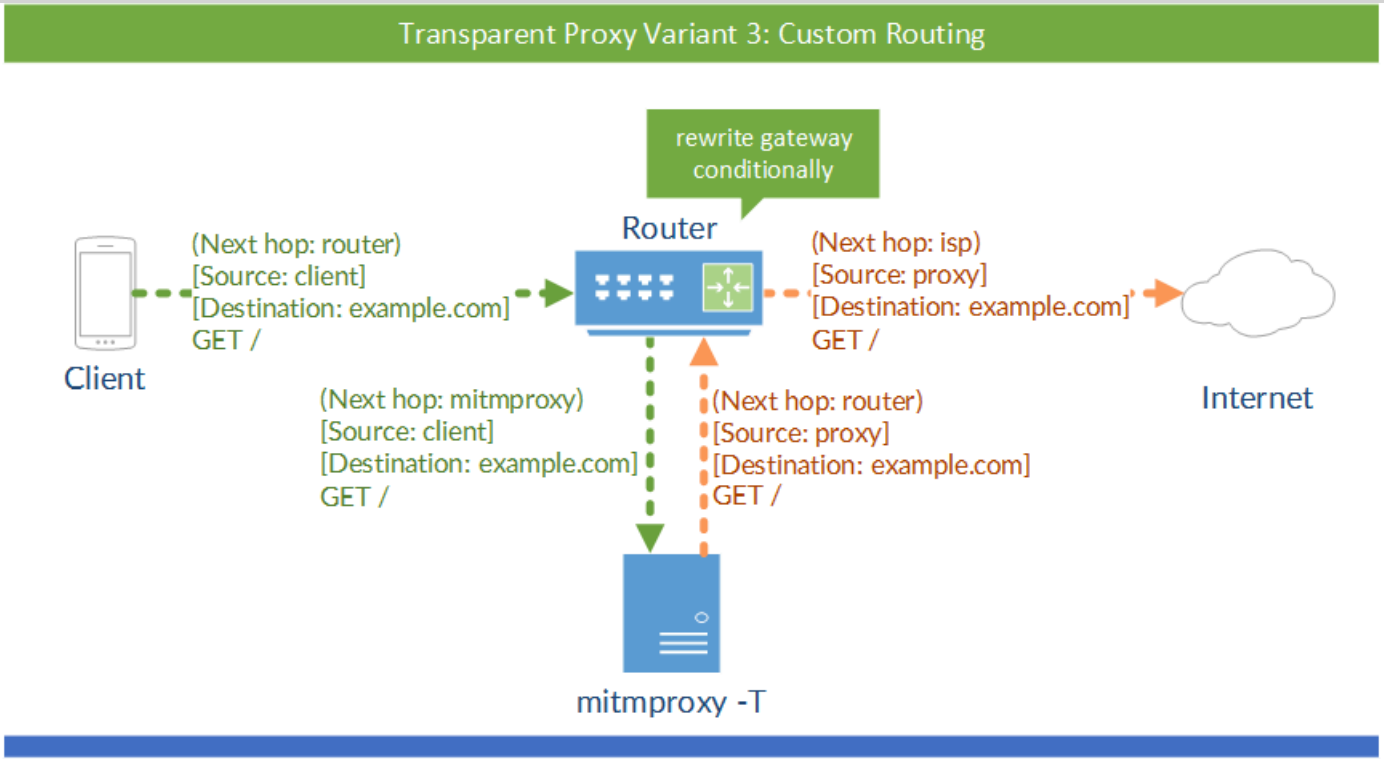
Proxy cache y proxy NAT

- **Proxy cache** web: Aplicación específica de acceso a la web, para mejorar su rendimiento. Mantienen copias locales de los archivos más solicitados.
- **Proxy NAT**: Integra los servicios de traducción de direcciones de red privada-pública en el Proxy.

Proxy transparente

Es un proxy normal, no es necesario configurar las aplicaciones clientes, como el navegador web, cliente ftp, cliente de correo, etc. para indicar que deben interactuar con un servidor proxy (dirección proxy), ya que las peticiones siempre van destinadas al proxy.

Pero el proxy transparente combina un **servidor proxy** con un **enrutador** o firewall. Por simplicidad se puede hacer que se ejecuten en la misma máquina.



1. Los clientes envían las peticiones a los servidores de destino originales, por ejemplo Google, teniendo como gateway el enrutador-firewall que trabaja en conjunto con el proxy.
2. El enrutador-firewall reenvía las conexiones de los clientes destinadas al puerto 80 hacia el puerto del servicio proxy (3128 habitualmente).
3. El proxy filtra la petición, tal como esté definido en sus políticas, y reenvía la petición si es necesario hacia el enrutador-firewall para que la reenvíe a Internet.

Esto elimina la necesidad de configurar los clientes para ir a través de proxy.

- Las aplicaciones NO son conscientes de tener un proxy por el medio y por eso se llama transparente (como un vidrio que permite ver Internet a través de él).
- Este tipo de proxy no permite hacer autenticación de usuarios como hacen los proxys convencionales.

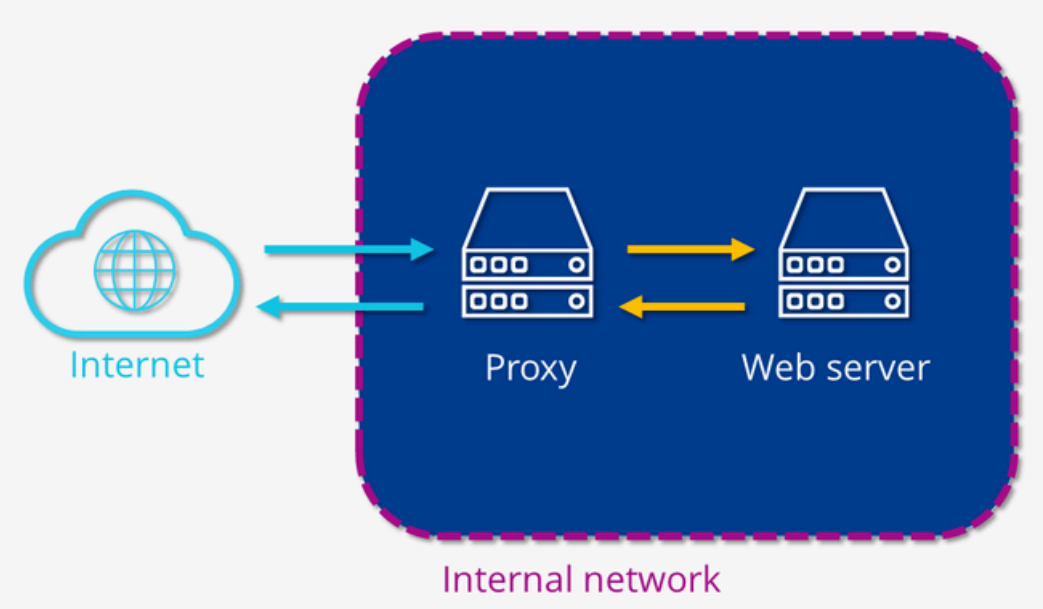
Squid también lo permite

Proxy anónimo

Es un proxy público que aumenta la privacidad y el anonimato de los clientes proxy mediante la eliminación de características identificativas.

- De hecho, como las peticiones reales parten del proxy, es su dirección IP de origen la que aparece en los paquetes de tráfico.
- Los servidores web a los que se accede no pueden ver la dirección IP real del cliente, sino la del proxy.

Proxy inverso (reverse proxy)



https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-proxy-inverso/

El proxy inverso **se sitúa en la red de destino**, donde se encuentran los servidores, y no en la red origen, donde están los clientes, como los proxy-caché.

El proxy inverso es un proxy instalado en una red con diferentes servidores web (o de otros tipos), sirviendo de intermediario a las peticiones externas.

- **Da "servicio" a servidores** web, de bases de datos, etc. y **no a los clientes**.
- Mejora la seguridad de los servidores que tiene detrás, ya que recibe todas las peticiones y las gestiona.
- Mejora el rendimiento ya que **hace de servidor de balanceo de carga**, repartiendo las peticiones entre los servidores que tiene detrás.
- También **guarda en memoria caché el contenido estático**, por ejemplo archivos html de los servidores que tiene detrás.

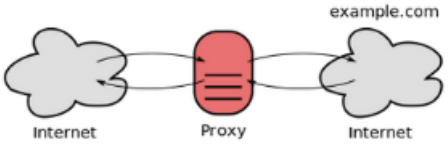
Ejemplos pueden ser **HA Proxy, NGINX, Traefik**.

Exemple de configuració de NGINX com a proxy invers

Proxy abierto o público

Este proxy acepta peticiones de cualquier ordenador, esté o no conectado a su red. Suelen situarse en Internet como proxy público.

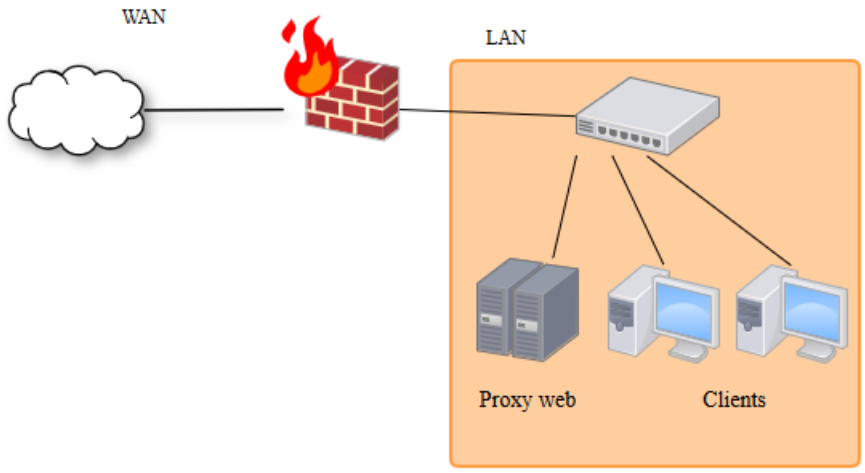
Es un servidor proxy público, susceptible de abuso pero también puede ser usado por espías.



1.5. Posible ubicación del proxy

No intercepción

En este caso, el proxy es un equipo más de la red local.

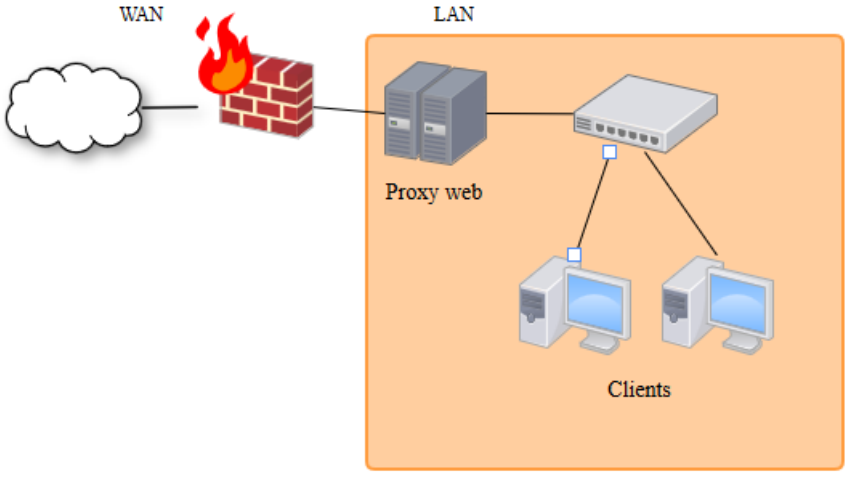


Medidas adicionales:

- Hay que configurar los equipos para que utilicen el proxy (en la aplicación cliente web) pero los usuarios con conocimientos podrían “saltarse” el proxy.
- Hay que configurar el firewall para que sólo acepte tráfico proveniente del proxy y no de los clientes de la red local.
- Alguien podría suplantar la IP del proxy si se encuentra en la misma red.

Intercepción

El proxy es un equipo que intercepta físicamente las conexiones de red (está en medio), y está dotado de dos interfaces de red.



- En este caso, si cae el proxy, quedamos sin conexión a internet y habríamos de tener soluciones alternativas.

2. Proxy con Squid

Squid http://www.squid-cache.org/ es un popular programa libre que implementa un servidor proxy y memoria caché de páginas web, con licencia GPL.

- [x] Permite acelerar un servidor web, guardar en memoria caché peticiones repetidas a DNS y otras búsquedas, memoria caché de web y seguridad filtrando el tráfico.
- [x] Está diseñado para ejecutarse bajo entornos Unix (pero funciona con Windows).
- [x] Se le considera muy completo y robusto.
- [x] Orientado a HTTP y FTP pero compatible con otros protocolos.
- [x] Implementa diversas modalidades de cifrado como TLS, SSL, y HTTPS.

Características de Squid

Algunas de sus características son:

- [x] Proxy y Memoria caché HTTP, FTP, HTTPS.
- [x] Proxy para SSL
- [x] Proxy transparente
- [x] Soporta WCCP (Web Cache Control Protocol), control de múltiples proxys para balancear carga.
- [x] Reglas de Control de acceso
- [x] Aceleración de navegación HTTP.

2.1. Instalación y configuración de Squid

Para instalar Squid en Ubuntu, ejecutamos:

```
sudo apt-get update
sudo apt-get install squid
```

Para comprobar que Squid está en ejecución, ejecutamos:

```
sudo systemctl status squid
# o bien
sudo service squid status
```

También podemos comprobar que Squid está escuchando en el puerto 3128 con:

```
sudo netstat -tuln | grep 3128
```

Y por último, podemos ver los losg de squid en `/var/log/squid/access.log` y `/var/log/squid/cache.log`.

Una vez instalado, podemos configurar Squid editando el archivo de configuración `/etc/squid/squid.conf`.

Este fichero está dividido en secciones, de las cuales, las más importantes son:

- **OPTiONS FOR AUTHENTICATION**: Para configurar la validación de usuarios. Vàries directives "auth...".
- **ACCESS CONTROLS**: Directivas "acl" y "http_access".
- **NETWORK OPTIONS**: Directiva "http_port".

- **OPTIONS WHICH AFFECT THE NEIGHBOR SELECTION ALGORITHM**: Directiva "cache_peer"
- **MEMORY CACHE OPTIONS**: "cache_mem", "maximum_object_size_in_memory",
- **DISK CACHE OPTIONS:** "cache_dir", "minimum_object_size", "maximum_object_size"
- **OPTIONS FOR URL REWRITING**: "url_rewrite_program", "url_rewrite_children", "url_rewrite_access"

Algunas de las directivas más importantes de Squid son:

- **http_port**: Define el puerto en el que Squid escucha las peticiones HTTP.
- **cache_dir**: Define el directorio de la memoria caché de Squid.
- **cache_mem**: Define la cantidad de memoria caché de Squid.
- **acl**: Define las listas de control de acceso de Squid.
- **http_access**: Define las reglas de acceso de Squid.
- **http_reply_access**: Define las reglas de respuesta de Squid.
- **cache_peer**: Define los servidores de memoria caché de Squid.
- **never_direct**: Define las excepciones de memoria caché de Squid.
- **refresh_pattern**: Define las reglas de actualización de la memoria caché de Squid.
- **visible_hostname**: Define el nombre visible de Squid.
- **logformat**: Define el formato de los logs de Squid.
- **access_log**: Define el archivo de logs de Squid.
- **cache_log**: Define el archivo de logs de la memoria caché de Squid.

Antes de aplicar los cambios realizados en el archivo de configuración de Squid, es recomendable comprobar que la configuración es correcta con:

```
sudo squid -k reconfigure
# o bien
sudo service -k parse
```

Para aplicar los cambios en la configuración de Squid, reiniciamos el servicio con:

```
sudo systemctl restart squid
# o bien
sudo service squid reload
```

2.2. Condiciones de las reglas ACL (Access control list) en SQUID

Primero se definen las condiciones a comprobar para cada petición de los clientes. Se hace con la directiva acl.

> acl [nombre de la lista] [tipo] [componentes de la lista]

Las condiciones definidas se usan después en las reglas de control como **"http_access"**. El orden en que definimos las condiciones "acl" **no tiene importancia**.

```
acl localnet src 172.16,100.0/28
acl hosts_permesos src -i "/etc/squid/listas/permitidos"
acl hosts_profes 192.168.0.4 192.168.0.7
acl sites_prohibits url_regex facebook.com twitter.com
```

El parámetro -i permite ignorar mayúsculas y minúsculas.

Si la lista de parámetros es muy grande, es preferible usar un fichero para poner los parámetros (uno por cada línea). Habrá que poner el nombre de fichero con la ruta entre comillas.

```
acl hosts_permitidos src -i "/etc/squid/listas/permitidos"
```

En el fichero permisos habría una lista de ip's, una por cada línea, con los hosts permitidos:

```
92.168.0.25
192.168.0.35
192.168.0.70
```

> ⚠ **Ficheros de ACL**
>
> Nos debemos asegurar que el fichero existe en la ruta indicada y que el usuario "proxy" lo puede leer.

2.3. Reglas de control de acceso (acceso a http)

Después se definen las acciones (deny/allow) que se debe ejecutar con la petición del cliente si la/las condición/es que ponemos como parámetros se cumplen.

```
http_access [deny o allow] [acl implicadas]
```

> ⚠ **Orden de las reglas**
>
> El orden en que escribimos las HTTP_ACCESS es **MUY importante**. No se pueden escribir en cualquier orden (similar a las reglas iptables). Normalmente haremos:
>
> - Primero pondremos las reglas más específicas (afectan a menos hosts)
> - Después las menos específicas (afectan a más hosts).
>
> Esto no quiere decir que no haya casos en que se deba hacer de otra manera.

Ejemplo1: Regla genérica (afecta a muchos hosts). La segunda regla se tiene que poner siempre como política por defecto que afecta a todos los hosts, y cambiar el orden hará que nadie pueda navegar por Internet.

```
http_access allow hosts_permitidos
http_access deny all
```

Ejemplo2: Regla más específica, tiene dos condiciones definidas antes en las "acl". Afecta a las mismas IP's origen que antes pero ahora sólo a las peticiones que NO van destinadas a hosts prohibidos. Se tienen que cumplir las dos condiciones (AND). Sólo con que falle una, no se ejecuta la regla (que en este caso es allow).

- Si la regla se ejecuta, no se mira ninguna regla más (como en iptables).
- Se puede negar una acl con el símbolo !

```
// Si la IP es de un host permitido (AND) la URL NO contiene una palabra prohibida
http_access allow hosts_permitidos !sites_prohibidos
http_access deny all
```

- Si se quiere hacer un "OR lógico" (o sea, una condición o la otra) se tiene que duplicar la regla (en el ejemplo, tres acl de tipo src):

```
http_access allow red_profesores
http_access allow red_alumnos
```

- Si se quiere hacer AND (las dos acl se cumplen) se ponen una al lado de la otra.

```
http_access allow red_alumnos red_profesores
```

ATENCIÓN: Esta última regla es imposible que se cumpla nunca ya que las dos redes son diferentes y un paquete sólo puede tener ip origen en una de las dos.

2.4 Tipos de ACL

IP's origen i destino

Para mirar direcciones IP de origen o destino (src, dst).

```
acl aclname src ip-address/netmask ...  # Dirección IP del cliente (src=source=origen)
acl aclname src addr1-addr2/netmask ... # Igual pero con un rango de direcciones
acl aclname dst ip-address/netmask ...    # Dirección IP obtenida de la URL de destino (dst=destino)
```

Ejemplos:

```
acl pc-jefe src 192.168.0.22
acl pcs-alumnos src 192.168.0.50-192.168.0.60
acl ip-facebook dst 157.240.243.35
acl red-prohibida dst 192.168.10.0/24
```

Dominios de origen y destino

Para mirar dominios de origen o destino (srcdomain, dstdomain).

```
acl aclname srcdomain .foo.com ...          # Dominio origen de la petición
acl aclname dstdomain .foo.com ...          # Dominio de destino
```

Horarios

Para comprobar un horario:

```
acl aclname time [day-abbrevs] [h1:m1-h2:m2]   # Días y hora a comprobar
```

Abreviaturas:
S - Sunday, M - Monday, T - Tuesday, W - Wednesday H - Thursday, F - Friday, A - Saturday
h1:m1 ha de ser menor que h2:m2

```
acl horari_laboral_mati time MTWHF 09:00-14:30
```

Expresiones regulares en la URL y el PATH

Para buscar una expresión dentro de la URL:

```
acl aclname url_regex [-i] ^http:// ...   # Expresión regular que se busca dentro de la URL completa
acl aclname urlpath_regex [-i] \.gif$ ... # Expresión regular que se busca sólo dentro del camino de la URL
```

Para más información de las expresiones regulares (RegEx) y prueba: https://regexr.com/

Ejemplos

```
// Al principio de la URL, buscar la cadena seguida de cualquier carácter
acl aclname url_regex serv_publicidad ^http://adserver.*

// Buscar al final de la URL las letras mp3
acl aclname url_regex ficheros_mp3 -i mp3$
```

Otros tipos de ACL

```
acl aclname port 80 70 21 0-1024... # Puerto de destino de la petición
acl aclname proto HTTP FTP ...      # Protocolo de la petición
acl aclname method GET POST ...     # Método de la petición HTTP (GET, POST, HEADER...)
acl aclname browser [-i] regexp ... # Comprueba el nombre del navegador cliente
```

2.5. Ejemplos de configuración de Squid

Ejemplo 1

Condiciones a cumplir por la configuración

- Permitir acceder si "eres el jefe" O (OR) "vas a webs_permitidas"
- Si vas a una web que no esté en las permitidas y no eres jefe, se te deniega (última regla).
- En cuanto se cumple una, ya se hace el allow

```
http_access allow ip_jefe
http_access allow web_permesa
http_access deny all
```

> ¿El jefe podrá acceder a webs no permitidas?

Ejemplo 2 de uso de acl's y http_access

Condiciones a cumplir para el caso

- Permitir acceder si "eres el jefe" Y (AND) "vas a una web permitida"
- Has de ser jefe y ir a web permitida para navegar.
- ¡Se tienen que cumplir las dos condiciones!

```
http_access allow ip_jefe web_permesa
http_access deny all
```

> ¿El jefe podrá acceder a webs no permitidas?

Ejemplo completo: caso concreto

Condiciones a cumplir para el caso

- Se quiere que el jefe pueda acceder sin restricciones (ninguna) a internet.
- Hay unos cuantos sitios web permitidos para cualquier persona y a cualquier hora.
- No se permite el acceso a los empleados a webs prohibidas durante el horario de trabajo, pero si en horario de pausa.

```
acl pc_jefe src 192.168.0.10
acl palabras_prohibidas url_regex sex porn culo weapon chat
// Palabras excepciones escritas en el fichero con la ruta indicada
acl palabras_excepciones url_regex "/etc/squid3/acl/palabras_excepciones.acl"
// Identificador de red local
acl red_local src 192.168.0.10/255.255.255.0
acl pcs_empleados src 192.168.0.15-192.168.0.20
// Horario de la pausa
acl horario_pausa time MTWHF 14:00-15:00
```

Completa el fichero de configuración con las reglas de acceso.

Comprueba si se cumplen las condiciones con la siguiente tabla:

Horario de trabajo	Webs permitidas	Webs prohibidas
Jefe	ALLOW	ALLOW
Empleado	ALLOW	DENY

Horario de pausa	Webs permitidas	Webs prohibidas	Resto de webs
Jefe	ALLOW	ALLOW	ALLOW
Empleado	ALLOW	ALLOW	ALLOW

Actividades

1. Puedes indicar qué hacen las siguientes reglas?

- acl hora time MTWHF 08.00-15.05
  http_access deny !hora
- acl aula src
  http_access deny all
  http_access allow aula
- acl webs dstdomain www.google.es www.iesdoctorbalmis.com
  http_access deny webs

2. Configura un servicio proxy con Squid que proporcione acceso a Internet con la siguiente configuración:

- Hay dos tipos de usuarios, A y B.
- Los usuarios de tipo A pueden acceder a todas las direcciones, excepto algunas excepciones.
- Los usuarios de tipo B no pueden acceder a ninguna dirección, excepto algunas excepciones.
- Utiliza solo el puerto 8080/tcp (en lugar de 3128/tcp que es el que usualmente utiliza).