SERVIDOR PROXY

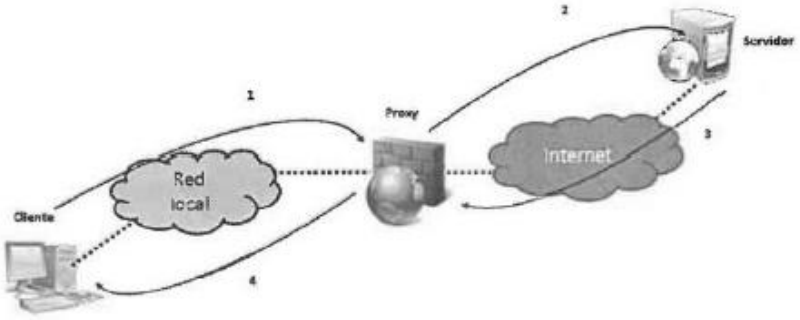
Es un software o dispositivo que realiza una función en nombre de otro sistema o aplicación denominado cliente proxy. El proxy aglutina todas las peticiones de los clientes de una red para "representarlos" frente a un servidor externo. Hay varios tipos: Proxy transparente y Proxy no transparente

Proxy transparente, intercepting proxy o forced proxy

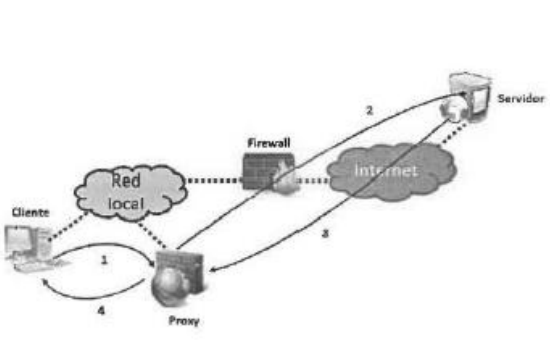
- Requiere que el equipo cliente tenga dirigida su ruta por defecto hacia él y examinará el tráfico y capturará las peticiones de los clientes
- La puerta de enlace por defecto del equipo cliente debe apuntar al proxy transparente
- Es frecuente que el proxy tenga habilitado el protocolo NAT para la traducción de direcciones IP internas en las IP externas del proxy
- El cliente ignora que sus peticiones son desviadas o capturadas por lo que no tiene que hacer ninguna operación de configuración adicional. Por este motivo es muy utilizado por los ISP (Internet Service Providers)
- Tiene algunos inconvenientes:
    - Problemas de autenticación ya que los protocolos que admiten no siempre permiten gestionar autenticación de cliente
    - Permite ocultar las actividades de los usuarios de redes de navegación anónima

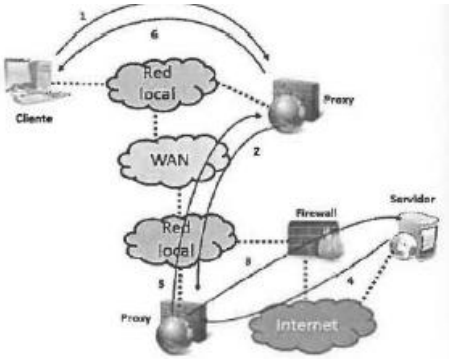
Fases a realizar en el entorno proxy para el acceso a un servidor remoto:

1. El cliente solicita el recurso al servidor adecuado haciendo llegar la petición primero al servidor proxy
2. El servidor proxy puede trasladar la petición como le llega o modificarla
3. El proxy envía la petición del cliente al servidor remoto en su nombre
4. El servidor remoto no sabe de quien viene la petición inicial, pero devuelve la respuesta al proxy que fue quien le preguntó
5. El servidor proxy cambia de nuevo las cabeceras y reenvía la respuesta al cliente.



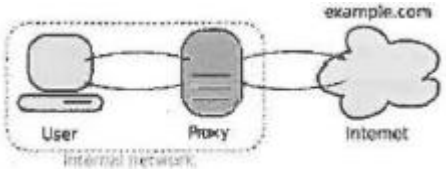
Los servidores proxy se pueden encadenar integrando en cada salto nuevas funciones pero como desventaja tiene el aumento de la latencia.



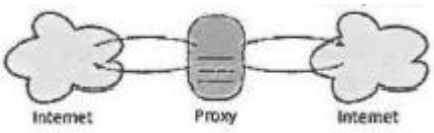


TIPOS DE SERVIDORES PROXY SEGÚN LA RELACIÓN CON SUS CLIENTES

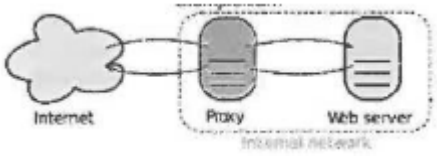
- Forward Proxy: El cliente debe invocar el nombre del servidor destino para realizar la conexión.



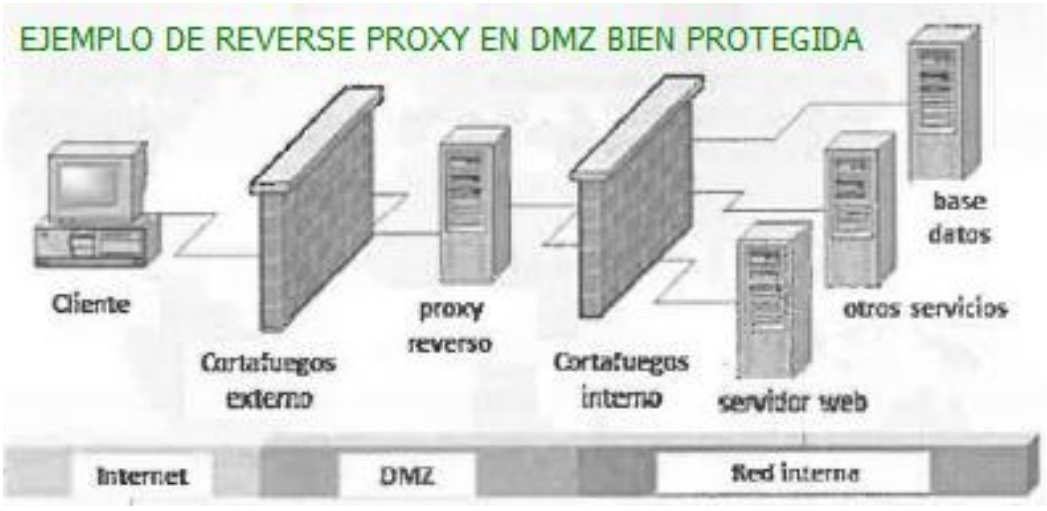
- Forward Proxy abierto: Es un proxy de tipo directo, es decir es accesible por cualquier equipo desde cualquier lugar de la red. (Suelen usarse como proxys anónimos)



- Reverse proxy: Recupera recursos de uno o mas servidores en nombre del cliente, los recursos son devueltos al cliente como si vinieran del proxy inverso (reverse) en vez del servidor. (Son muy útiles para asegurar los servicios de los servidores públicos)



  - Ocultan la existencia de características del servidor al que representan
  - Dificultan la penetración de malware a la LAN del servidor
  - Finaliza los túneles de cifrado SSL para liberar de esta carga al servidor quedando este tramo entre proxy y servidor en la red LAN sin cifrar.
  - También aligeran la carga del servidor mediante técnicas de caching.
  - Ahorran ancho de banda comprimiendo datos en la conexión.
  - Pueden hacer balanceo de carga con otros servidores equivalentes en la misma LAN (load balancing)



Comandos del servicio. Cuando hagamos cambios en la configuración del proxy, se debe parar o reiniciar el servicio.

- Parar el servicio /etc/init.d/squid3 stop
- Reiniciar el servicio /etc/init.d/squid3 restart
- Crear enlaces simbólicos para el autoarranque update-rc.d squid3 defaults
- Eliminar los enlaces para el autoarranque update-rc.d -f squid3 remove

Directivas de caché. Squid no solo es un servidor proxy web, En su fichero de configuración también encontramos algunas directivas de gestión de caché para almacenar páginas web.

- Define el usuario con el que Squid operará en la caché cache_effective_user proxy
- Define el e-mail de notificación de errores de Squid cache_mgr fulanito@hotmail.com
- Adjudicación de memoria RAM para la caché cache_mem 32 MB
- Tamaño máximo de objetos guardados en caché máximum_object_size 4096 KB
- Cache_dir Type Directory-Name Mbytes Level1 [options]
    - Type → Define el Sistema de almacenamiento: ufs, aufs…
    - Directory-name → Es el directorio de la caché por defecto, en Ubuntu /var/spool/squid3
    - Mbytes → Es la memoria reservada para la caché
    - Level1 → Es el número de directorios de primer nivel, por defecto 16
    - Level2 → Es el número de subdirectorios de segundo nivel, por defecto 256

  Se pueden habilitar varias cachés en distintas ubicaciones escribiendo más de una directiva

Reglas/listas de acceso. Una vez definidas las ACL se pueden crear reglas que permitan o denieguen el acceso en función de si cumplen o no las ACL asociadas a dicha regla.

http_access http_reply_access reply_body_max icp_access always_direct never_direct

http_access allow | deny acl1 acl2…
http_access allow | deny acl3 acl4…
http_access deny all

← **QUE SE INTERPRETAN COMO** →

http_access allow | deny acl1 AND acl2… OR
http_access allow | deny acl3 AND acl4… OR
http_access deny all

Métodos de autenticación en un proxy Squid

#Tomando la web oficial de Squid, declaramos una lista blanca de dominios

Acl whitelist dstdomain .whitelist.com .goodsite.com .partnerssite.com

#Declaramos el protocol de navegación y dos ACL para los puertos seguros e inseguros

Acl http proto http

Acl port_443 port 443

Acl port_80 port 80

#Declaración de los métodos http que podrían crear túneles SSL

Acl CONNECT method CONNECT

#Obligación para los usuarios de autenticarse para formar parte de autenticated_users

Acl authenticated_users proxy_auth REQUIRED

#Reglas para permitir el acceso por el puerto 80 hacia la lista blanca de dominios a usuarios no autenticados.

http_access allow http port_80 whitelist

#Se permite la creación de túneles SSL solo para el puerto 443 y hacia la lista blanca

http_access allow CONNECT port_443 whitelist

Reglas solo para usuarios autenticados que pueden navegar por el puerto 80 y a cualquier destino

http_access allow http port_80 authenticated_users

Regla para que los usuarios autenticados puedan crear túneles SSL por el puerto 443 a cualquier destino.

http_access allow CONNECT port_443 authenticated_users