Teoria sobre els diferents tipus de proxy HTTP/S possibles

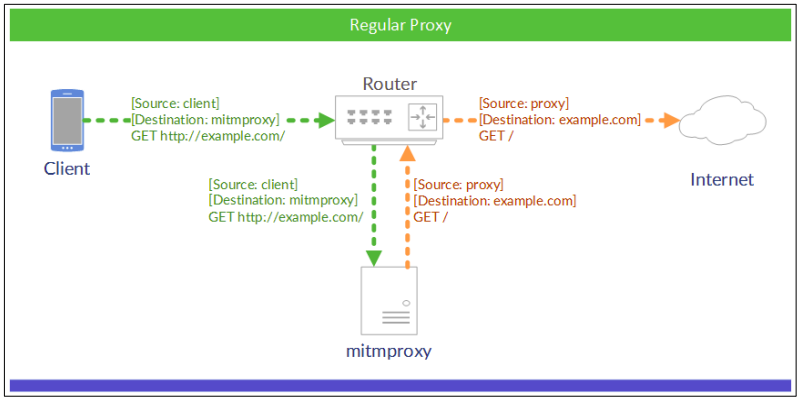
Si volem interceptar el trànsit HTTP o HTTPS dels clients web presents a la nostra LAN podem escollir entre diverses opcions

***Proxy HTTP explícit** : En aquest mode, cada client web, les sol·licituds HTTP del qual volem interceptar, (és a dir, els navegadors) s'ha de configurar per utilitzar explícitament el nostre servidor intermediari. A Firefox, per exemple, això es pot fer des de "Paràmetres->General->Paràmetres de xarxa->Paràmetres"; allà, la IP i el port del nostre Mitmproxy es poden especificar a l'opció "Configuració manual->Servidor d'HTTP".

> NOTA: Una altra opció és "Utilitza els paràmetres de servidor intermediari del sistema" perquè delega la especificació de la IP i el port del nostre proxy a la configuració general del sistema (accessible en Gnome a través de l'opció "Xarxa"->"Servidor intermediari de xarxa"->"Manual"->"Servidor d'HTTP" del panell de control). Fent-ho així, aquesta configuració pot ser general per tots els clients web del sistema així que ja no serà necessari configurar el proxy a cadascun d'ells de forma individual

Un proxy (o "servidor intermediari") en aquest mode accepta una connexió del client HTTP i reenvia la sol·licitud al servidor web de destinació (potser després d'inspeccionar-la i/o d'editar eventualment la càrrega útil de la sol·licitud sense cap problema, ja que els paquets es transmeten sense xifrar). Per al client, sembla com si el proxy simplement estigués transmetent la seva connexió (com ho fan els routers o els servidors del nostre ISP). I per al servidor, sembla que el proxy és el client.

> NOTA: Un programa força habitual per implementar un proxy HTTP (ja sigui explícit o transparent, i fins i tot HTTPS) és **Squid** (http://www.squid-cache.org, el qual està especialitzat, més enllà d'aspectes com la censura de destins (via inspecció de dominis DNS/URLs/IPs/ports/...) i del registre dels events (peticions, respostes, etc) en logs per fer-ne estadístiques, en gestionar el contingut "catxejat" de peticions prèvies de múltiples clients. Un altre software que pot funcionar com a proxy HTTP és **TrafficServer** (https://trafficserver.apache.org)



***Proxy HTTPS explícit**: En aquest mode, cada client web, les sol·licituds HTTPS del qual volem interceptar, (és a dir, els navegadors) s'ha de configurar per utilitzar també explícitament el nostre servidor intermediari. A Firefox, per exemple, es pot fer des de "Paràmetres>General->Paràmetres de xarxa->Paràmetres"; allà, la IP i el port del nostre proxy es poden especificar a l'opció "Configuració manual->Servidor d'HTTPS".

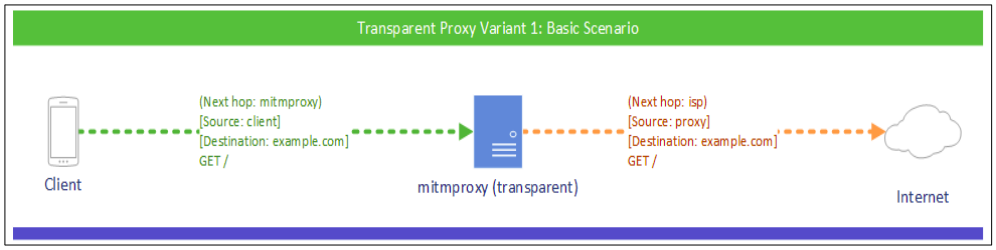
> NOTA: Una altra opció és "Utilitza els paràmetres de servidor intermediari del sistema" perquè delega la especificació de la IP i el port del nostre proxy a la configuració general del sistema (accessible en Gnome a través de l'opció "Xarxa"->"Servidor intermediari de xarxa"->"Manual"->"Servidor d'HTTPS" del panell de control). Fent-ho així, aquesta configuració pot ser general per tots els clients web del sistema així que ja no serà necessari configurar el proxy a cadascun d'ells de forma individual

En aquest mode, el client es connecta al proxy i fa una sol·licitud de tipus CONNECT (com per exemple aquesta *CONNECT example.com:443 HTTP/1.1*) Un proxy convencional no pot visualitzar ni manipular un flux de dades xifrat amb TLS ja que una sol·licitud CONNECT el que fa és simplement demanar al proxy que obri una "canonada" TCP entre el client i el servidor de manera que el proxy aquí només actuï com un "facilitador" en reenviar a cegues dades en ambdues direccions sense saber res del contingut.

Per poder inspeccionar i manipular les sol·licituds HTTPS, (que estan xifrades per definició, tal com hem dit) el truc és convertir el proxy en una autoritat de certificació de confiança per poder generar així certificats d'intercepció sobre la marxa que suplantaran als certificats de servidor legítims (vegeu la NOTA a continuació); fent això sí que s'exposaran llavors totes les sol·licituds HTTPS interceptades. Però per aconseguir que el client confiï en aquests certificats d'intercepció, el proxy s'ha de registrar prèviament com a CA de confiança en el client web les sol·licituds del qual es volen estudiar

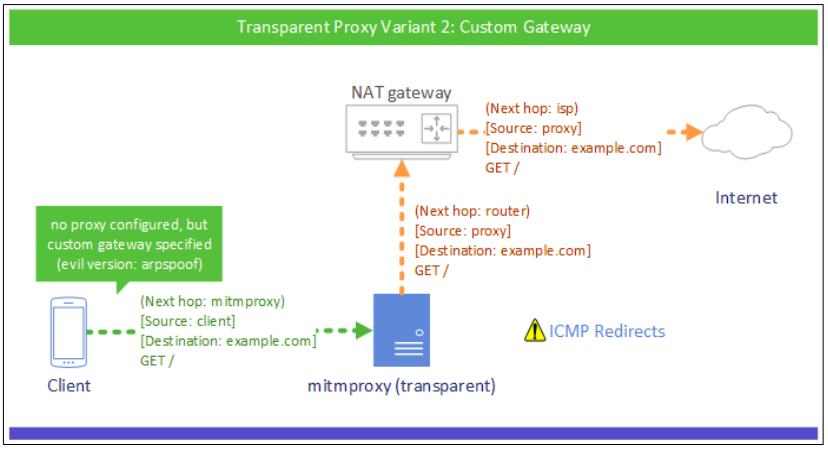
NOTA: Les connexions HTTPS xifren cada sol·licitud i resposta entre el client i el servidor amb un secret compartit d'extrem a extrem, de manera que un servidor intermediari HTTPS normal no pot desxifrar els paquets de dades intercanviats. Quan el client obre una connexió TLS al servidor web segur, abans de transferir qualsevol dada verifica la identitat del servidor mitjançant la comprovació de dues condicions: en primer lloc, verifica si el seu certificat ha estat signat per una CA coneguda pel client i en segon lloc, s'assegura que el "common name" (també anomenat "CN", que normalment correspon al seu nom DNS totalment qualificat) del servidor coincideix amb el que es connecta. Si les dues condicions són certes, el client assumeix que la connexió és segura. Per poder detectar la connexió, el proxy hauria d'actuar com a autoritat de certificació (com Verisign, Letsencrypt o similar). Tanmateix, en comptes d'emetre certificats a persones o organitzacions reals, el proxy genera de forma dinàmica certificats amb qualsevol nom d'amfitrió que sigui necessari per a una connexió. Si, per exemple, un client vol connectar-se a https://www.facebook.com, el proxy genera un certificat per a "www.facebook.com" i el signa amb la seva pròpia CA. Sempre que el client confiï en aquesta CA, les dues condicions esmentades anteriorment són certes (CA de confiança, mateix CN), el que significa que el client creu que el servidor intermediari és de fet "www.facebook.com".

***Proxy HTTP transparent**: En aquest mode, els clients web no tenen cap proxy configurat explícitament, però igualment les seves sol·licituds HTTP sempre passaran, de totes maneres, per un proxy "transparent". Per aconseguir això, el truc està en què s'ha de configurar el proxy transparent com a porta d'enllaç per defecte dels clients, tal com es mostra a la figura següent; d'aquesta manera, totes les peticions que van a Internet passaran necessàriament pel proxy.

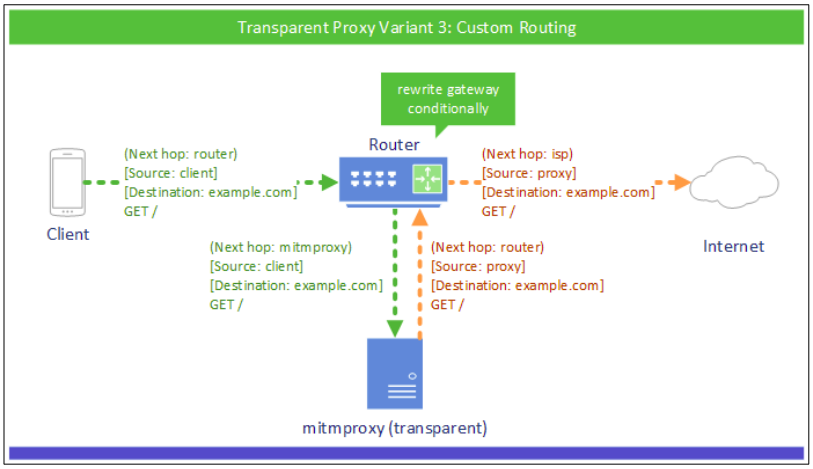


Idealment, la mateixa màquina hauria de fer les dues tasques: ser un encaminador i ser un servidor intermediari HTTP/S. Si féssim servir un ordinador amb Linux per fer les dues tasques, seria relativament fàcil. Però, ¿què passa si ja tenim un encaminador funcionant com a porta d'enllaç a la nostra LAN? Podríem fer una d'aquestes dues opcions:

1.-Obligar a establir el proxy com a porta d'enllaç per defecte de tots els clients de LAN (mentre es manté el router com a porta d'enllaç per defecte només del proxy). La manera més senzilla de fer-ho és enviant aquesta configuració als clients mitjançant DHCP. Si no és possible, una altra manera és fent un atac "ARP Spoofing".



2.-Implementar l'encaminament personalitzat a la configuració real del router. Aquesta opció és útil quan es necessita un control més detallat de quin trànsit arriba al proxy i quin no. Es pot, per exemple, triar desviar al servidor intermediari transparent només el trànsit cap a determinats destins. Hi ha un gran nombre de maneres d'aconseguir-ho, i tot dependrà de la marca i el model del router. En la majoria dels casos, però, es recomana la primera opció per la seva facilitat d'ús.

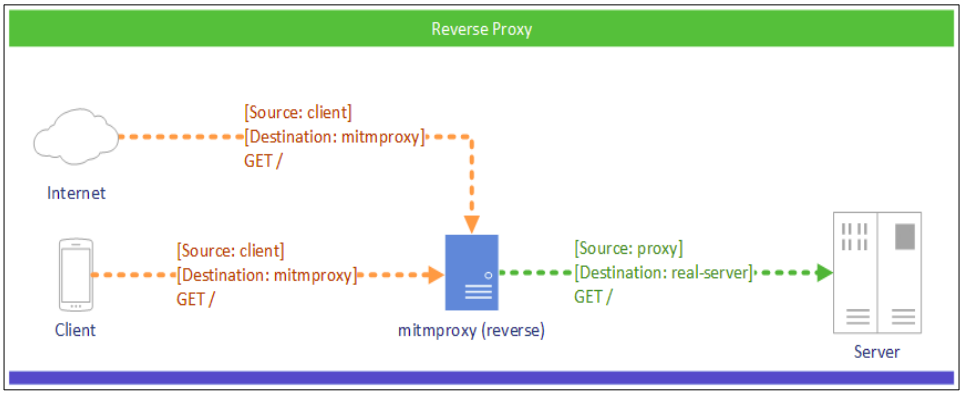


*__Proxy HTTPS transparent__: La seva implementació és bàsicament la del proxy HTTP transparent amb l'ús addicional del certificats CA descrits al mode "HTTPS explícit"

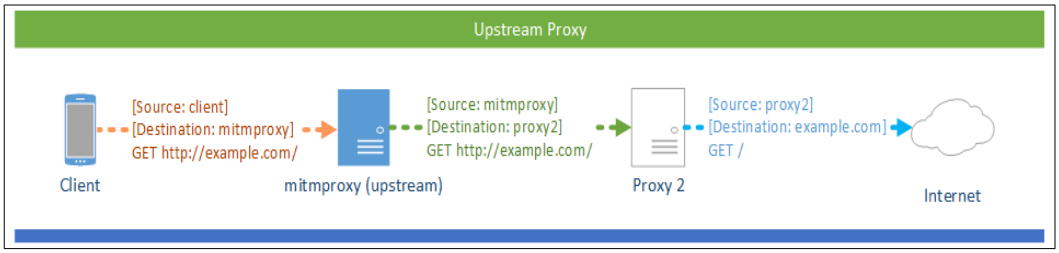
***Proxy HTTP/S invers:** Un proxy invers pren sol·licituds provinents d'Internet i les reenvia als servidors web presents a la nostra xarxa local interna. En aquesta configuració, aquestes sol·licituds al servidor intermediari no són conscients de l'existència dels servidors de la xarxa interna, protegint-los així dels atacs comuns basats en la web. Altres motius per implementar "proxys inversos" són utilitzar-los com a servidors de contingut "catxejat" i/o com a punts d'autenticació centralitzats i/o per fer un equilibri de càrrega entre servidors interns (si n'hi ha més d'un) i/o com a "terminadors TLS" (és a dir, servidors que realitzen el (des)xifratge TLS, descarregant d'aquesta tasca als servidors interns, els quals podran ser llavors simples servidors HTTP plans), etc.

NOTA: Programes que poden actuar com a "proxy invers HTTP/S" són **Nginx** (https://www.nginx.com), **HAProxy** (http://www.haproxy.org , el qual també pot actuar com a proxy invers TCP genèric), el propi **Apache** (https://httpd.apache.org) sempre que tingui el mòdul "mod_proxy" activat, **Varnish** (https://varnish-cache.org) o també **TrafficServer** (https://trafficserver.apache.org)

NOTA: Una manera molt fàcil de distingir un proxy de tipus "invers" d'un proxy que sigui "directe" és fixar-se en qui tria el servidor HTTP/S final: si és el propi proxy, llavors aquest és "invers", si és el client web, llavors és "directe". És per això que, per exemple, per anar a un servidor web fent servir un proxy invers només ens cal especificar aquest -perquè serà ell qui s'hi connecti al servidor web en qüestió sense poder intervenir-hi des del client- (*curl https://reverse.example.com*) però si fem servir un proxy directe hem d'especificar tant aquest com el servidor web remot concret al qual volem connectar (*curl -x https://forward.example.com https://web.server.com*). En tot cas, teniu més informació a https://kinsta.com/blog/reverse-proxy



***Proxy HTTP/S "upstream":** Si es volen encadenar servidors intermediaris afegint Mitmproxy davant d'un altre proxy, es pot utilitzar el mode "upstream" de Mitmproxy. En aquest mode, totes les sol·licituds rebudes per Mitmproxy es transfereixen incondicionalment a l'altre servidor intermediari prèviament establert.



NOTA: Mitmproxy en particular també pot ser utilitzat com un proxy de tipus **SOCKS5**

En tot cas, si el lector vol saber-ne més, a https://docs.mitmproxy.org/stable/concepts-modes i a https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks hi ha molta més informació sobre els detalls tant de la teoria com de la implementació dels modes aquí explicats. De fet, d'aquí s'ha obtingut aquest diagrama: