Ocultación de tráfico con Obsfproxy y OpenVPN

enero 17, 2017 adastra Deja un comentario Go to comments

OpenVPN es una muy buena alternativa a la hora navegar de forma privada y segura, crear una VPN es algo relativamente fácil de configurar, dependiendo evidentemente de tus necesidades y de las complejidades que pueda tener tu entorno/configuración, no obstante, en algunas ocasiones no se puede utilizar de forma directa debido a restricciones y medidas de censura que se aplican en ciertos lugares del mundo, por ejemplo, si te encuentras en China continental te vas a encontrar con que tu cliente de OpenVPN no se podrá conectar con el servidor debido a los bloqueos impuestos por "el gran firewall" chino, las medidas de censura instauradas por éste y otros gobiernos impiden que se puedan completar las conexiones utilizando un canal cifrado. En este caso concreto, para conseguir el filtrado de dichas conexiones, los ISPs junto con los "entes censores" implementan técnicas para el análisis de tráfico y detección de patrones "sospechosos", dichas técnicas son conocidas como DPI (Deep Packet Inspection) permitiendoles saber que aunque el tráfico se encuentre cifrado, utiliza algún tipo de protocolo concreto, el cual pueden encontrarse restringido o no y evidentemente, partiendo de dicha información, se toma la decisión de permitir o bloquear el tráfico de forma automática. Este problema ya lo han enfrentado soluciones de anonimato como TOR y tal como os comentaba en otro articulo, la mejor forma a día de hoy de "saltarse" esas restricciones consiste en utilizar implementaciones de "Pluggable Transports" los cuales se encargan de alterar el tráfico generado desde el cliente y enmascararlo de tal forma que de cara al ISP o censor se trata de paquetes de datos que no cumplen con ninguno de los patrones de bloqueo y en apariencia, tiene el mismo comportamiento de una petición habitual, como una búsqueda en Baidu (http://www.baidu.com/) o Sogou (http://www.sogou.com/). Existen múltiples implementaciones de Pluggable Transports y aunque las más robustas y extendidas han sido desarrolladas por el equipo de TorProject, existen otras implementaciones que buscan la interoperabilidad de los "Pluggable Transports" en soluciones de privacidad y anonimato distintas a TOR, como es el caso de uProxy, Lantern o Psiphon.

Para utilizar cualquier Pluggable Transport es importante instalar los componentes adecuados tanto en el cliente como en el destino (servidor). Dichos elementos se encargan de enmascarar y desenmascarar los paquetes de datos enviados y recibidos, de tal forma que desde el cliente se aplica el Pluggable Transport adecuado, el cual se encargará de aplicar las modificaciones correspondientes sobre los paquetes de datos y posteriormente se envían al destino. En el lado del cliente funciona como un proxy local, el cual intercepta las peticiones y las procesa antes de salir hacia el gateway de la red. En el lado del destino o servidor, funciona exactamente igual pero en sentido inverso, es decir, se reciben los paquetes de datos por parte de un listener o servicio que "desenmascarará" las peticiones y recompondrá el mensaje enviado originalmente por el cliente. Un mecanismo sencillo y muy robusto que es altamente resistente a la censura.

Una vez comprendido el funcionamiento de los Pluggable Transports y ver que realmente no es demasiado complejo, podemos intentar llevarlo a la practica utilizando alguna implementación de PT disponible, en este caso, se hará con una red VPN implementada con OpenVPN y la implementación de PT Obfsproxy, la cual ha sido desarrollada por el equipo de TorProject y funciona bastante bien.

Configuración en el lado del cliente.

En primer lugar, el cliente de la red VPN debe configurarse de tal forma que las conexiones no se lleven a cabo de forma directa contra el servidor OpenVPN, en su lugar, la conexión se realizará contra el proceso de Obfsproxy en el lado del servidor y éste a su vez, se encargará de reenviar los paquetes hacia el servidor OpenVPN correspondiente. Para hacer esto, basta simplemente con cambiar el parámetro “remote” en el fichero de configuración del cliente o directamente desde consola.

remote <IPSERVIDOR_OBFSPROXY> 21194

A continuación, se debe instalar Obfsproxy en el sistema del cliente y luego, levantar el proceso que se encargará de ofuscar el tráfico enviado desde la máquina del cliente. Dicho proceso, como se ha mencionado anteriormente, funcionará simplemente como un proxy local que se encargará de tratar los paquetes y enviarlos al servidor Obfsproxy indicado.
Para instalar obfsproxy sobre un sistema basado en Debian, basta simplemente con ejecutar el comando “apt-get install obfsproxy”, aunque también se puede instalar directamente con pip ejecutando el comando “pip install obfsproxy”.
Después de instalar, el siguiente comando permitirá iniciar el proceso en el lado del cliente en el puerto “10194”

obfsproxy –log-file=obfsproxy.log –log-min-severity=info obfs3 –shared-secret=<32 caracteres> socks 127.0.0.1:10194

Como se puede apreciar, se especifica un fichero de logs, nivel de log, el PT a utilizar con obfsproxy que en este caso será “obfs3” y finalmente, la opción “–shared-secret” corresponde a una cadena de texto que debe ser la misma en el cliente y servidor, además es recomendable que dicha cadena de texto tenga por lo menos 32 caracteres para generar una clave de 256bits. El parámetro “socks” le indica a obfsproxy que debe levantar un servidor proxy socks en el puerto 10194 en la máquina local, el cual deberá ser utilizado por el cliente de OpenVPN. Ahora bien, antes de continuar, se recomienda generar la clave compartida utilizando OpenSSL, tan sencillo como ejecutar el siguiente comando:

openssl rand -base64 32

Recordar que la salida del comando anterior debe ser incluida en el parámetro “–shared-secret” y además, se trata de un valor que debe ser igual en el proceso obfsproxy en el lado del cliente y en el lado del servidor.

El último paso para que la configuración de OpenVPN+Obfsproxy quede completada en el lado del cliente, consiste nuevamente en modificar el fichero de configuración del cliente openvpn e incluir las siguientes instrucciones “socks-proxy-retry” y “socks-proxy”. En el fichero de configuración estas instrucciones quedarían así:

socks-proxy-retry
socks-proxy 127.0.0.1 10194

El puerto “10194” evidentemente tiene corresponder con el valor indicado en obfsproxy cuando se ha iniciado el proceso cliente.

Configuración en el lado del servidor.

En el lado del servidor OpenVPN hay que seguir unos pasos muy similares a los que se han llevado a cabo en el lado del cliente, solamente que en este caso, es incluso más sencillo ya que en primer lugar, únicamente hay que editar el fichero de configuración “server.conf” y asegurarse de que el puerto indicado en el parámetro “port” coincide con con el que se va a utilizar posteriormente en el proceso servidor de Obfsproxy. En este caso, se asume que dicho puerto será el 1194 (valor por defecto en un servidor OpenVPN).
A continuación, se debe iniciar el proceso servidor de Obfsproxy utilizando el siguiente comando.

obfsproxy –log-file=obfsproxy.log –log-min-severity=info obfs3 –dest=127.0.0.1:1194 –shared-secret=<32 caracteres> server 0.0.0.0:21194

Como se puede apreciar, el comando ejecutado en el lado del servidor es muy similar al que se ha lanzado desde el lado del cliente, con la diferencia de que se ha indicado el interruptor “–dest” que permite apuntar al puerto donde se encuentra ejecutándose OpenVPN, lo cual a efectos prácticos, significa que todo el tráfico que obfsproxy consigue desenmascarar será reenviado al puerto especificado, es decir, al puerto en donde se encuentra ejecutándose el servidor OpenVPN. Por otro lado, tal como se ha indicado anteriormente, es necesario que el valor de “–shared-secret” sea el mismo tanto para el cliente como para el servidor. Se trata de la clave utilizada para descifrar los paquetes (cifrado end-to-end). Por último, se indica también que Obfsproxy se ejecutará en modo servidor en el puerto “21194” debido al valor “server” indicado por parámetro.

Ahora que todo está preparado, se debe arrancar el servidor Openvpn y si todo va bien, se puede arrancar el cliente Openvpn con el fichero de configuración correspondiente, esperar a que la conexión con el servidor se establezca correctamente y ver los logs. También resulta muy interesante capturar el tráfico tanto en cliente y servidor para ver qué es lo que hace obfsproxy y cómo altera los paquetes de datos.
Como has podido ver, ha sido fácil de configurar y los resultados son muy interesantes. Intenta probarlo en tus servidores y si estás interesado, aporta a la red de Tor levantando Bridges con OBFS

Un saludo y happy hack!