1. **Proxy**

El término en ingles «Proxy» tiene un significado muy general y al mismo tiempo ambiguo, aunque invariablemente se considera un sinónimo del concepto de «Intermediario». Se suele traducir, en el sentido estricto, como delegado o apoderado (el que tiene el poder sobre otro). Su finalidad es interceptar las conexiones de red que un cliente hace a un servidor de destino.

Los motivos por los que se introduce un proxy en la red son: seguridad, rendimiento y anonimato...

Estos son los pasos que se siguen cuando un cliente realiza una petición, y tenemos un proxy de red:

* Un cliente se conecta al Servidor Intermediario (Proxy).
* El cliente solicita al proxy una conexión, fichero u otro recurso disponible en un servidor distinto.
* El servidor Intermediario (Proxy) proporciona el recurso ya sea conectándose al servidor especificado o sirviéndose de su caché.

Conceptualmente, los proxys se distinguen de los cortafuegos en que los proxys trabajan a nivel de aplicación, mientras que los cortafuegos operan en el nivel de red. Los firewalls deniegan el acceso a ciertas IP y puertos TCP o UDP, y los proxys deniegan el acceso a ciertas páginas web y/o a usuarios de la red.

Una aplicación común de los servidores proxs es funcionar como caché de contenido de red, proporcionando en la proximidad de los clientes una caché de páginas y ficheros disponibles a través de la red en servidores remotos, permitiendo a los clientes de la red local acceder a ellos de forma más rápida y confiable.

Cuando se recibe una petición para un recurso de Red especificado en una URL, el servidor intermediario busca el resultado de la URL dentro de la caché. Si éste es encontrado, el proxy responde al cliente proporcionando inmediatamente el contenido solicitado. Si el contenido solicitado no estuviera disponible en la caché, el proxy lo traería desde el servidor remoto, entregándolo al cliente que lo solicitó y guardando una copia en la caché.

El contenido de la caché va siendo sustituido de acuerdo con el algoritmo de expiración elegido. Éste podrá basarse en la antigüedad de la información almacenada, en su tamaño o en el historial de solicitudes recibidas.

Los algoritmos de expiración más habituales son:

LRU (Least Recently Used o Menos Recientemente Utilizado). En este algoritmo los objetos que no han sido accedidos en mucho tiempo son los primeros en ser eliminados, manteniendo siempre en la caché los objetos más recientemente solicitados. Esta política es la utilizada por Squid de modo predeterminado.

LFUDA (Least Frequently Used with Dynamic Aging o Menos Frecuentemente Utilizado con Envejecimiento Dinámico). En este algoritmo los objetos más veces solicitados permanecen en la caché, eliminándose los que menos veces se han solicitado.

GDSF (GreedyDual Size Frequency o Frecuencia de tamaño GreedyDual).

Este algoritmo, además de tener en cuenta la frecuencia con la que se reciben solicitudes, también tendrá en cuenta el tamaño de los objetos solicitados. De este modo, se da prioridad a los objetos pequeños (que ocupan menos memoria) con muchas solicitudes.

1. **Servidor proxy en Linux: Squid**

Squid es un proxy muy popular y ampliamente utilizado entre los sistemas operativos GNU/Linux. Es muy confiable, robusto y versátil, y se distribuye bajo los términos de la Licencia Pública General GNU (GNU/GPL).

* 1. **Instalación y arranque de Squid**

Para instalar squid, una vez comprobada la configuración de red, tan sólo tendremos que abrir el terminal y escribir:

**sudo apt-get install squid**

* 1. **Inicio y parada del servicio**

Para arrancar *squid*, escribe:

**service squid start**

Para pararlo:

**service squid stop**

Y para reiniciarlo:

**service squid restart**

Reiniciar o iniciar la configuración de Squid permite verificar si hay errores en el fichero de configuración; pero ten cuidado: si no devuelve mensajes de depuración, pero detectas algún error, mira en el fichero /var/log/syslog:

**sudo gedit /var/log/syslog**

1. **Configuración básica de Squid**

El archivo de configuración se encuentra en ***/etc/squid/*** y se llama ***squid.conf***. Antes de modificarlo, conviene realizar una copia de seguridad.

Una configuración básica debe incluir, al menos, los parámetros que se indican a continuación:

* **http\_port**: Establece el puerto de escucha para squid (por defecto: puerto 3128).
* **visible\_hostname**: nombre del equipo que queremos que aparezca en los mensajes de error.
* **acl**: a cada ACL o lista de control de acceso se le hace corresponder una (o más) regla de control de acceso (http\_access) que es la que permite o deniega las conexiones desde o hacia los recursos definidos en aquélla.

Otros parámetros importantes son:

* **cache\_dir**. Establece la localización y el tamaño de la caché en el disco duro. Ejemplo: **cache\_dir ufs /var/spool/squid 100 16 256**. ufs es el sistema de almacenamiento que utiliza squid. 100 el tamaño en megas de la caché, 16 el nivel de subdirectorios de primer nivel y 256 el segundo nivel de subdirectorios por cada directorio de primer nivel.
* **cache\_replacement\_policy**: se indica el algoritmo de sustitución que va a utilizar la memoria caché cuando se llene.
  1. **Las listas de control de acceso o ACL**

Mediante las listas de control de acceso se hacen agrupaciones de diferentes recursos (IPs de origen y destino, URLs, franja horaria…) y a otros conceptos para luego permitir o denegar el acceso mediante las reglas de control de acceso.

* **acl nombre src [IP] [IPRED/MASCARA] [ARCHIVO]**

**acl equipo1 src 192.168.1.23**

**acl redlocal src 192.168.1.0/255.255.255.0**

**acl varios src “/etc/squid/varios”**

Ejemplo de contenido del archivo */etc/squid/varios*

**192.168.2.45**

**192.168.2.78**

**192.168.2.46**

* **acl nombre dstdomain [nombre de dominio] [archivo]**

**acl prohibidas dstdomain** [www.google.es](http://www.google.es/)[www.hotmail.co](http://www.hotmail.com/)**m**

**acl prohibidas dstdomain “/etc/squid/prohibidas”**

Ejemplo de contenido del archivo */etc/squid/prohibidas*

[www.google.es](http://www.google.es/)

[www.hotmail.es](http://www.hotmail.es/)

* **acl nombre time [días][rango horario] (S-domingo, M-lunes, T-martes, W-miercoles, H-jueves, F-viernes, A-sabado)**

**acl horario1 time AS 08:00-24:00 (sabados y domingos de 8h a 24h)**

* **acl nombre url\_regex [caracteres] [archivo]**

Permite especificar expresiones regulares; puede ser interesante, por ejemplo, para agrupar sitios con diferentes extensiones:

**acl extensiones url\_regex .com .es**

* **acl nombre urlpath\_regex [caracteres] [archivo]**

Permite especificar la extensión de ficheros a descargar:

**acl descargas urlpath\_regex \.pdf$ \.doc$**

* **ACLs predefinidas**

Existen unas ACLs ya configuradas que podemos usar directamente: **localhost**, **all** (cualquier equipo), **localnet** (red a la que pertenece el servidor).

Una vez hemos creado los elementos que intervienen en la red, vamos a definir las reglas que van a determinar el funcionamiento del proxy; en concreto las reglas que van a controlar las solicitudes HTTP a servidores Web. El proxy analizará estas reglas y decidirá si responde a la solicitud del cliente o no.

* 1. **Las reglas de acceso http\_access**

**http\_access allow/deny !nombreAcl**

Allow: permite el acceso a las solicitudes que cumplan las condiciones de las acl (si hay varias acl-s, comprueba que se cumpla las condiciones de todas)

Deny: deniega el acceso a las solicitudes que cumplan las condiciones de las acl (si hay varias acl-s, comprueba que se cumpla las condiciones de todas

Se puede utilizar el operador ¡ delante del nombreAcl; con ello, negamos las condiciones de esa acl.

**http\_access allow redlocal (permiso de acceso a todos los equipos de la red local)**

**http\_access deny equipo1 (se deniega el permiso al *equipo1*)**

**http\_access deny prohibidas (se deniega el permiso a las prohibidas desde cualquier equipo)**

**http\_access deny horario (se deniega el permiso en ese *horario1* desde cualquier equipo)**

**http\_access deny equipo1 prohibidos (se deniega el acceso a las páginas incluidas en *prohibidos* al *equipo1*)**

Hay que tener muy en cuenta el orden de las reglas de acceso. Se deben colocar de más concretas a mas generales. Cuando al proxy le llega una petición, se comprueba en orden cada regla hasta que alguna sea aplicable; a partir de ahí, no se comprueban más reglas. Conviene poner como última regla de acceso la que queramos que establezca la política por defecto (aceptar todo o denegar todo).

**http\_access allow equipo1**

**http\_access deny prohibidos redlocal**

**http\_access allow horario redlocal**

**http\_access deny all !localhost**

El *equipo1* no tiene ninguna restricción, los equipos de la red local no pueden acceder a las paginas prohibidas y sólo pueden acceder en el horario indicado.

Con la última restricción se deniega el acceso en cualquier otro caso (excepto a localhost)

* 1. **Mensajes de error y registro**

Por lo general, Squid viene preconfigurado con mensajes en inglés, pero podemos modificarlo para que estos mensajes de error salgan en español automáticamente o poner los nuestros propios. Si queremos que aparezcan en español, en el archivo de configuración de *squid* pondremos:

**error\_directory /usr/share/squid/errors/Spanish**

Si queremos modificar por ejemplo el mensaje de error que aparece al impedir el acceso a una determinada página, editaremos el archivo */usr/share/squid/errors/Spanish/ERR\_ACCESS\_DENIED*.

Los ficheros donde se guarda el registro de incidencias son:

***/var/log/syslog***: almacena mensajes de error

/***var/log/squid/access.log***: almacena información sobre los accesos

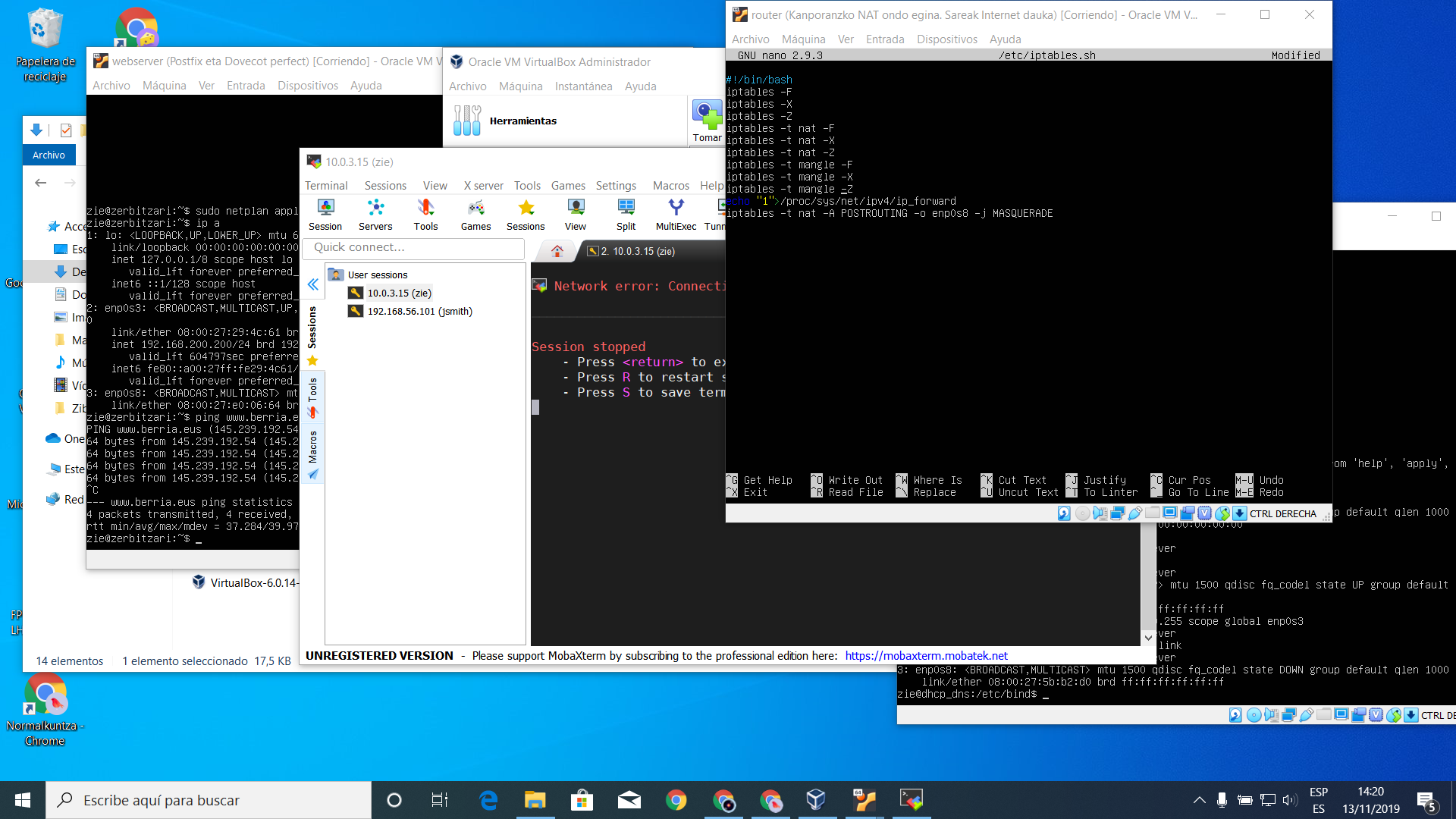
***/var/log/squid/cache.log***: almacena información sobre la caché.

1. **Nuestro primer proxy**
   1. **Preparación de la red local**

Supongamos que tenemos una red local con dos servidores: *dhcp-dns* (servicios: dhcp y dns) y *zerbitzari* (servicios: *http* y *mail*). Entre el exterior y la red interna se encuentra una máquina *router* que hace de intermediaria. Ésta tiene dos interfaces de red (una es la puerta de enlace de los equipos de la red local, y la otra es la que está conectada al exterior).

1. Deberemos comprobar que hay conectividad entre las máquinas de la red local.
   1. Haz PING entre ellas.
   2. Usando *w3m*, comprueba que se accede a la página web desde cualquiera.
2. Crearemos un script de nombre */etc/iptables.sh*

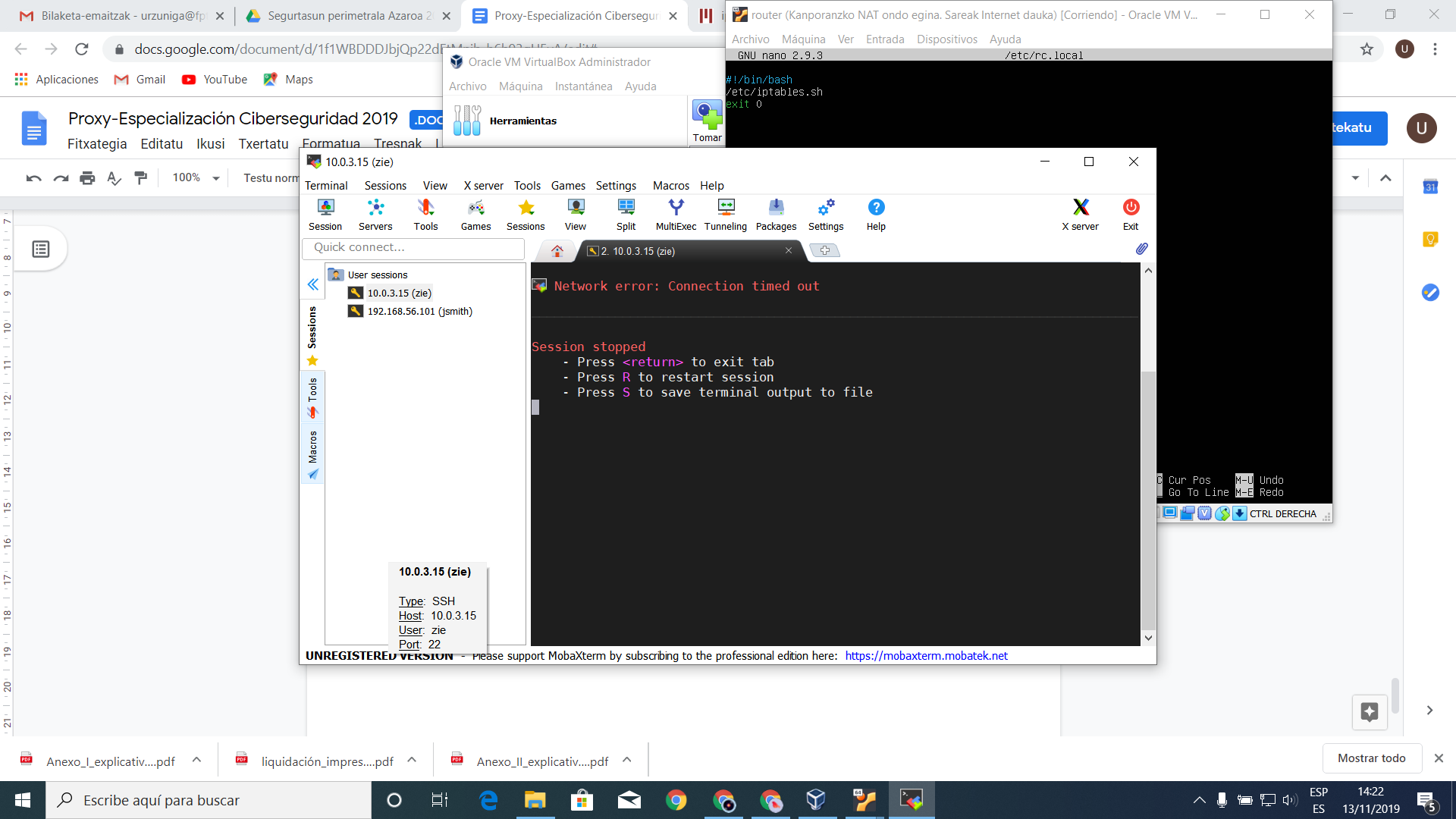
* Vaciaremos las tablas.
* Activaremos el bit de Forwarding
* Haz que la IP privada se cambie por la pública cuando el tráfico viaje desde la red local hacia el exterior.



1. Comprobaremos que el script se ejecuta sin errores.

*sh /etc/iptables.sh*

1. Automatizaremos la ejecución de ese script al iniciarse el sistema:



1. Asignaremos permisos de escritura a los dos scripts.

chmod +x /etc/rc.local

chmod +x /etc/iptables.sh

1. Reiniciaremos la máquina, y comprobaremos que desde las máquinas de la red local se puede acceder a Internet. Si queremos simular una navegación, podremos utilizar la aplicación *w3m*.

w3m [www.berria.eus](http://www.berria.eus)

* 1. **Proxy transparente**
     1. **Puesta en marcha**

Un proxy transparente es un proxy cuya existencia es desconocida para el resto de equipos de su red. En nuestro caso, lo lógico es que el proxy transparente lo tuviéramos en la máquina *router*, que, como hemos dicho, hace de intermediaria entre la red local y el exterior; por lo tanto, es el lugar oportuno para filtrar las navegaciones desde el interior hacia Internert.

Lo primero que haremos será redirigir esas navegaciones al puerto 8080, que será donde esté escuchando el servicio *squid*. Para ello, añade lo siguiente al fichero */etc/iptables.sh*.

iptables -t nat -A PREROUTING –s 192.168.200.0/24 -p tcp --dport 80 -j REDIRECT --to-port 8080

Instala el servicio *squid*.

apt-get install squid

Ahora, rellenaremos de manera muy básica el fichero de configuración de squid, de tal manera que se pueda navegar a cualquier sitio siempre:

**http\_port 3128**

**http\_port 8080 transparent**

**http\_access allow all**

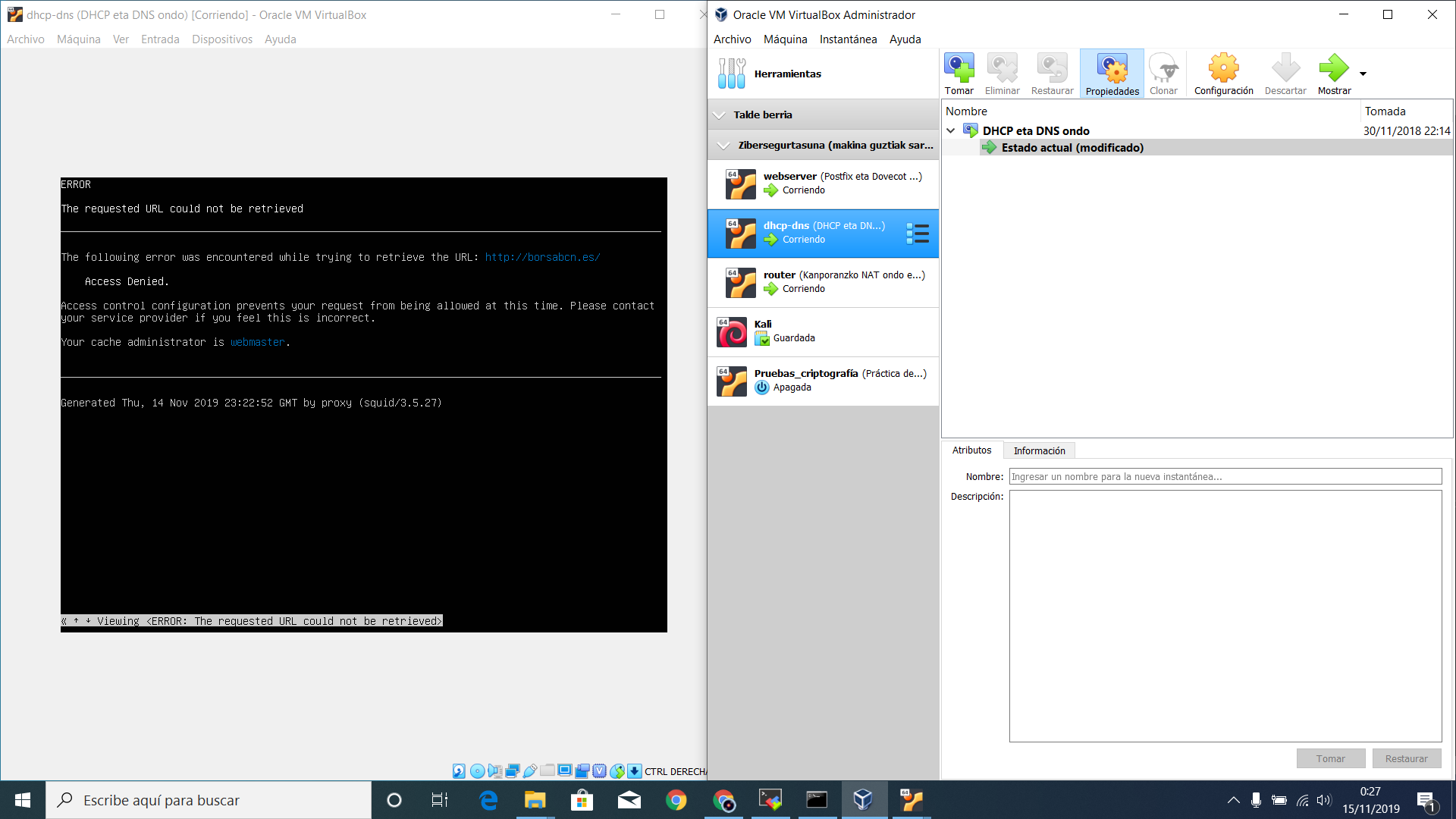
Reiniciaremos el servicio.

sudo service squid restart

Si no aparecen errores, iremos a la máquina *dhcp-dns* e intenta navegar a [www.borsabcn.es](http://www.borsabcn.es). Deberá permitirnos navegar.

Ahora, cambiaremos el fichero de configuración para que siempre se impida la navegación. Reiniciaremos el servicio.

Si no aparecen errores, iremos a la máquina *dhcp-dns* e intentaremos navegar a [www.borsabcn.es](http://www.borsabcn.es). No deberá permitirnos navegar:



* + 1. **Medidas de seguridad básicas**

Posibles problemas del proxy transparente: nuestra máquina proxy hace un man in the middle entre los clientes y el router, de tal manera que no se le exige al navegador del cliente que use TLS.

Una medida de seguridad muy oportuna es filtrar el tráfico según el puerto de destino de la petición. Así, es conveniente crear una acl para los puertos que están a salvo, y denegar todo el tráfico para el resto. En nuestro caso, mantendremos de momento a salvo sólo los puertos 80 y 443, ya que el proxy lo usaremos exclusivamente para permitir o no navegar.

