

Bloqueando Facebook por http y https

Índice de contenido

Introducción.....	1
Configurando el firewall (iptables).....	2
Configurando el servidor de nombres (bind9 - DNS).....	3
Configurando el servidor web (apache2).....	5

Introducción

Este tutorial explica como denegar el acceso a facebook (ya sea por http o por https) utilizando el firewall (iptables), el servidor de nombres (bind9) y el servidor web (apache2).

A grandes rasgos, lo que haremos será evitar la consultas a otro DNS que no sea el nuestro, utilizando algunas reglas en nuestro firewall. Luego, haremos un poco de DNS Spoofing (lea http://es.wikipedia.org/wiki/Spoofing#DNS_Spoofing), suplantando la identidad de facebook.com en nuestra red (le haremos crear a los clientes de nuestra red que este DNS responde por facebook.com). Luego crearemos un **virtual host** en apache para nuestro dominio facebook.com.

NOTA: En el tutorial se utiliza la consola o terminal para configurar todo.

Vamos! No muerde.

Cuando se especifica un # al principio, quiere decir que debemos ser root, a excepción que estemos hablando de un archivo de configuración, en ese caso se está hablando de un comentario. Cuando la linea empieza con \$ es un usuario común (en los server ubuntu de exo es el usuario admsrv).



Configurando el firewall (iptables)

En el firewall de la escuela utilizamos una versión modificada de la configuración compartida por [@mhoyos](#) en la lista de Referentes Tecnicos en NTICs de Argentina, que puedes descargar desde [acá](#).

En esa configuración, tenemos DROP como política de FORWARD

:FORWARD DROP [339345:17392875]

“Pateamos” los paquetes que no estén específicamente permitidos. Las reglas que permiten pasar los paquetes por ciertos puertos son (se encuentran un poco más abajo):

```
#Permite ssh
-A FORWARD -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -p tcp -m tcp --sport 22 -j ACCEPT
#Permite la navegación web
-A FORWARD -p tcp -m tcp --sport 80 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 80 -j ACCEPT
#permite la navegación web por https
-A FORWARD -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -p tcp -m tcp --sport 443 -j ACCEPT
#Permite la consulta de dns
-A FORWARD -p udp -m udp --sport 53 -j ACCEPT
-A FORWARD -p udp -m udp --dport 53 -j ACCEPT
#Navegación por puerto 4040 (el sitio de lapampa.edu.ar )
-A FORWARD -p tcp -m tcp --sport 4040 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 4040 -j ACCEPT
#Abro los puertos para los clientes de correo
-A FORWARD -p tcp -m tcp --sport 25 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 25 -j ACCEPT
-A FORWARD -p tcp -m tcp --sport 143 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 143 -j ACCEPT
-A FORWARD -p tcp -m tcp --sport 993 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 993 -j ACCEPT
-A FORWARD -p tcp -m tcp --sport 465 -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 465 -j ACCEPT
```

Si lo dejamos así, basta con que los alumnos cambien a mano su dns para que este método



no funcione, por lo tanto NO permitimos el forward de consultas DNS. Luego veremos como configuramos nuestro dns para que él consulte por dominios no resueltos por él.

Por lo tanto comentamos las lineas donde abríamos el puerto 53, quedando así:

```
#-A FORWARD -p udp -m udp --sport 53 -j ACCEPT
#-A FORWARD -p udp -m udp --dport 53 -j ACCEPT
```

Recuerden que el archivo con la definición de las reglas está en **/etc/iptables.up.rules** y que en **/etc/network/interfaces** debe especificar que establezca las reglas de firewall al levantar las interfaces. Esto se logra con la linea **post-up iptables-restore < /etc/iptables.up.rules**.

El archivo completo de configuración es [éste](#).

Editamos nuestro **/etc/iptables.up.rules** y cuando este listo, actualizamos las reglas:

```
# iptables-restore < /etc/iptables.up.rules
```

Configurando el servidor de nombres (bind9 - DNS)

Los cambios que haremos no afectan a tu configuración actual (si ya resuelves otros nombres).

Primero editaremos el archivo **/etc/bind/named.conf.options**, debe quedar con el siguiente contenido:

```
options {
    directory "/var/cache/bind";
    forwarders {
        ip_del_dns_del_ISP;
        otra_ip_DNS;
    };
    auth-nxdomain no;
    listen-on-v6 { any; };
};
```

La parte que nos interesa es el forwarders, donde especificaremos los DNS de nuestro proveedor, de google, de openDNS, o alguno externo que resuelve el resto de los nombres no resueltos por este.

También es importante que la opción `directory "/var/cache/bind"`; no este comentada, ya que será en esa ubicación donde estarán los archivos de las zonas.

Ahora editamos `/etc/bind/named.conf.local`. Quizás ya tengas otras zonas configuradas, no hay problema, agregamos la siguiente zona al final:

```
zone "facebook.com" {
    type master;
    file "db.facebook.com";
};
```

No se olviden del `;` al final. Observen que en `file` utilizamos un path relativo. Bind9 buscará ese archivo en `/var/cache/bind`.

Bien, por último crearemos el archivo `/var/cache/bind/db.facebook.com` con el siguiente contenido:

```
$ORIGIN facebook.com.
$TTL 86400; 1 día
@ IN SOA server1 admin (
    2011100311; serie
    6H; refresco (6 hs)
    1H; reintentos (1 hs)
    2W; expira (2 semanas)
    3H; mínimo (3 hs)
)
@ IN NS ns1.facebook.com.
ns1 IN A 172.16.0.1
www IN A 172.16.0.1
```

Recuerden respetar todos los puntos al final de los nombres, sino se concatena el contenido de `$ORIGIN`.

Lo que esta en **negrita**, debes cambiarlo según tus requerimientos (nombre del servidor, IP, etc).

Una vez finalizada la edición de estos archivos, procedemos a reiniciar el servicio:

```
# service bind9 restart
```

Si salió todo bien, deberíamos probar si resuelve bien los nombres:

```
$ dig facebook.com ns
; <<>> DiG 9.8.1 <<>> facebook.com ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43153
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;facebook.com.                IN      NS

;; ANSWER SECTION:
facebook.com.                86400   IN      NS      ns1.facebook.com.

;; ADDITIONAL SECTION:
ns1.facebook.com. 86400   IN      A       172.16.0.1

;; Query time: 3 msec
;; SERVER: 172.16.0.1#53(172.16.0.1)
;; WHEN: Tue Oct  4 10:09:44 2011
;; MSG SIZE rcvd: 64
```

Efectivamente, facebook.com apunta a nuestro servidor. Ahora a configurar apache.

Configurando el servidor web (apache2)

Primero algunas consideraciones iniciales. La raíz de apache, de forma predeterminada es `/var/www`. En esa ubicación debemos tener un enlace simbólico a `/home/admsvr/html_public` llamado `public`, sería:

```
/var/www/public → /home/admsrv/html_public
```

En una terminal hacemos:

```
# ln -s /home/admsrv/html_public /var/www/public
```

De esta forma, armamos nuestro sitio en el home del usuario, con la comodidad que esto implica sin disminuir o modificar los permisos de la carpeta /var/www.

Comento esto porque en los virtual host que configuraremos aparecen estas ubicaciones.

Bien, desde una consola nos ubicamos en /etc/apache2/sites-available

```
# cd /etc/apache2/sites-available
```

Copiamos el archivo default

```
# cp default facebook
```

Editamos facebook, quedando así:

```
<VirtualHost *:80>
    ServerAdmin admin@cep.edu.ar
    ServerName facebook.com
    ServerAlias www.facebook.com
    DocumentRoot /var/www/public/facebook
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/public/facebook/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
</VirtualHost>
```

En ServerAdmin ponen su correo. Básicamente, tanto facebook.com como www.facebook.com apuntan al sitio alojado en /var/www/public/facebook.

Debemos crear este directorio, pero recuerden que su ubicación real es en el /home/admsrv/html_public; hacemos:

```
$ mkdir /home/admsrv/html_public/facebook
```

Y colocamos dentro de este directorio un sitio web avisando que facebook no esta

disponible dentro de la escuela. Les comparto [la página que hice para esto](#).

Ahora editaremos el archivo default-ssl contenido dentro del mismo directorio (/etc/apache2/sites-available). Acá solo editaremos algunas líneas, ya que hay opciones específicas para el TDServidor al final del archivo. Las líneas que cambiaremos son (deben quedar así):

```
ServerName facebook.com
ServerAlias www.facebook.com
DocumentRoot /var/www/public/facebook
<Directory /var/www/public/facebook/>
```

El resto queda igual.

Bien, una vez creado el documentRoot en /home/admsrv/html_public/facebook procedemos a habilitar los sitios. Hacemos

```
# a2ensite facebook
```

Nos dirá que debemos recargar la configuración de apache.

```
# service apache2 restart
```

para reiniciar o

```
# service apache2 reload
```

para recargar la configuración.

Si todo salió bien, ya contamos con nuestra propia versión de facebook =)

Cuando pongamos en nuestro navegador <https://facebook.com> nos saldrá un aviso de conexión no confiable. Es porque el certificado que entrega no es el de facebook, sino el nuestro. Muchos no entrarán, otros agregarán la excepción y oups...



Espero les sirva, y cualquier cosa no duden en comunicarse conmigo, ya sea por algo que no se entendió o algo que haya que corregir.

Pueden hacerlo en la [lista de referente tics de argentina](#), a través de los comentarios en <http://elscriptdelsysadmin.com.ar>, por Twitter [@matuvarela](#) o Identi.ca [@matuu](#)

Saludos!