

Dominios no deseados: cómo manejar los anfitriones virtuales

Introducción

Hace muchos años, décadas, uno deseaba tener, al menos, un dominio web para uso personal o de nuestra empresa, y los más aventureros de temáticas, de noticias, entretenimiento y demás. Lo cierto del caso es que con la delegación de dominios regionales (en nuestro caso, Venezuela con las extensiones **.ve**) hemos podido tener varios, vender varios, montar varios... En este 2019 llegamos al punto de que hasta tenemos que lidiar con varios... **¡que no son nuestros!**

¿Qué son los anfitriones virtuales?

Cuando Tim Berners-Lee montó el primer servidor web utilizó una máquina Next con una gigantesca cantidad de memoria RAM para la época: ¡1 gigabyte! esto fue posible gracias a que Steve Jobs fue despedido de Apple y se propuso realizar una amplia gama de potentes máquinas de escritorio (más que todo para diseño gráfico y películas, recuerden «Toy History» de 1995).

No me consideren ignorante, se muy bien que los *mainframes* de IBM eran máquinas colosales pero la misma burocracia corporativa no hubiera permitido el uso de equipos para estas aventuras, **recordemos que las primeras redes eran militares y luego académicas** (universidades y centros de investigación como el CERN donde aún trabaja Tim Berners). Un servidor web era para un solo dominio web, no daba para más (chistoso el aviso que le colocaron para que no apagarán el primer servidor web, los ordenadores eran tan caros que los cuidaban como oro en paño).

Pero hoy en día las computadoras se han vuelto poderosas y nos permiten ahorrar direcciones IPv4, muy limitadas ellas. ¿Cómo se logra esto?

El programa servidor web **Apache** fue el primero en aprovechar esta características: de manera resumida y simplificada consiste en que según la petición web de cabecera HTTP determina a qué sitio web se refiere y retribuye o devuelve el contenido de la carpeta especificada a cada dirección web, sirviendo una misma dirección IP para varios dominios. Pero se pone mejor el ejemplo, lo normal es que un servidor web (para evitar fallos por medio de la redundancia) tenga varias interfaces de red, cada una con una dirección IP distinta, ya sea pública o privada (este caso lo analizaremos otro día).

Los anfitriones virtuales son los diferentes dominios que sirve un ordenador con **Apache** o **nginx** instalado y que de acuerdo a la petición del cliente lee las carpetas correspondientes (cada una con el mismo nombre del dominio) y las devuelve a cada usuario.

¿Que son los Sistemas de Nombres de Dominio?

Ya hemos hablado de ellos en otros artículos ([lean este enlace](#) y este [otro enlace](#)): esencialmente son máquinas a las cuales uno pregunta el nombre de un dominio y devuelve una dirección IP, hasta acá todo bien, por ello no explico más este asunto.

Dominios no solicitados

Por diferentes razones un servidor DNS puede indicar o apuntar que nuestro servidor web alberga un dominio o subdominio cualquiera:

- Puede suceder que una dirección IPv4 (más bien un bloque de direcciones) sean vendidas de una empresa a otra sin haber eliminado todos los DNS que apuntan a los dominios de la empresa anterior.
- En el caso de los que alojan máquinas virtuales como la empresa DigitalOcean, una dirección IPv4 la rentan a cualquier máquina virtual que la necesite, así uno destruye una máquina virtual pero los DNS a nivel mundial no se enteran de ello y pueden durar un buen tiempo apuntando los dominios a esa dirección IPv4 en particular y que de paso nosotros alquilamos... *El equivalente a esto es como cuando uno compra un teléfono móvil celular cuyo número era de otra persona: lo más probable es que recibamos un montón de llamadas desconocidas ¡y hasta nos insultan cuando contestamos! (por eso es mejor no atender números conocidos).*
- La otra opción es que alguien utilice su propio servidor web y configure, **quién sabe con cuáles intenciones**, uno o varios dominios hacia nuestra dirección IPv4. Esto se puede hacer, por ejemplo, montando un [dnsmasq](#).

Solución práctica

En los países industrializados esto lo manejan simplemente haciendo un rastreo o seguimiento para indagar lo que causa esos dominios no deseados apuntados hacia nuestro servidor web y ya con pruebas en mano ir al ISP, empresa que venda dominios web o incluso gobiernos (acá en Venezuela se tiene que acudir a la CONATEL) a exponer su caso.

Eso, mis estimadas lectoras y estimados lectores, es un buen gasto de tiempo, dinero y recursos, hay batallas que sin combatir uno las gana.

Por eso es que proponemos (e implementamos, esto es diciendo y haciendo) configurar que *todos y cada uno de los dominios y/o subdominios que recibamos en nuestro servidor web sean dirigidos al anfitrión virtual cero.*

Apache web server

En este popular servidor web el truco es el siguiente: configuramos el anfitrión virtual por defecto, el denominado **000-default.conf** configurado para atender la dirección ficticia **example.com**: (en idioma inglés *example* significa "ejemplo") sí, como saben (o ahora lo saben) [ese dominio web nadie tiene asignado ese dominio](#), o mejor dicho, la [IANA se reserva su uso](#), en este caso para propósitos puramente didácticos.

Como pueden imaginar nunca recibiremos solicitud alguna de ese dominio, por lo que todo lo que llegue al anfitrión virtual por defecto lo enviaremos a una página de error 404 por medio del siguiente fichero

```
ServerAdmin webmaster@example.com
ServerName example.com
ServerAlias www.example.com
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Dicho fichero en Ubuntu 18.04 está ubicado en el siguiente sitio:

```
/etc/apache2/sites-available/000-default.conf
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Luego debemos activarlo y recargar Apache (de nuevo en Ubuntu 18.04):

```
sudo a2ensite 000-default.conf
sudo systemctl reload apache2
```

Agregando el resto de los dominios y subdominios

Para agregar el resto de los dominios, deberemos crear un directorio y establecer derechos de lectura y escritura:

```
$ sudo mkdir -p /var/www/otrodominio.com.ve/public_html
$ sudo chown -R $USER:$USER /var/www/otrodominio.com.ve/public_html
$ sudo chmod -R 755 /var/www
```

Creamos el fichero de configuración:

```
$ sudo nano /etc/apache2/sites-available/otrodominio.com.ve.conf
```

Y le agregamos este contenido:

```
ServerAdmin webmaster@otrodominio.com.ve
ServerName otrodominio.com.ve
ServerAlias www.otrodominio.com.ve
DocumentRoot /var/www/otrodominio.com.ve/public_html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.
<https://www.ks7000.net.ve>

De nuevo grabamos y recargamos:

```
sudo a2ensite otrodominio.com.ve.conf
sudo systemctl reload apache2
```

Y repetiremos tantas veces como sean necesarias (en este punto podremos escribir un guion automatizado para esto).

Comprobando el orden establecido

Bastará con ejecutar **apache2ctl -S** para listar por pantalla el orden de prioridad con la que están listados y atendidos todos nuestros dominios, y cuando el servidor web llegue al final de la lista y no encuentre el dominio solicitado por el visitante volverá al principio de la lista y entregará lo correspondiente al dominio **example.com** pero como sabemos que jamás ni nunca habrá coincidencia, pues devolverá una página de error 404.

Para estas -y otras páginas de respuestas HTTP- tenemos una plantilla creada por el [señor Andi Dittrich y que nosotros contribuimos traduciendo al castellano](#) que mostrará nuestro correo electrónico en la página correspondiente (en este caso, error 404). La idea con este trabajo extra es que puedan contactar con nosotros, si es que es necesario tomar mayores acciones acerca del erróneo encaminamiento web hacia nuestra dirección IPv4.

nginx

No, no estamos muy prácticos con nginx (*nadie nace aprendido*) sin embargo es bastante parecido y en este [artículo de DigitalOcean en idioma inglés](#) (que está publicado bajo licencia Creative Commons que permite su traducción a cualquier idioma, anímense) está muy bien detallado y es muy parecido al Apache, solo que no los llaman anfitriones virtuales sino *bloques de servidor* ("server blocks").

Fuentes consultadas

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

En idioma inglés

Tutoriales en DigitalOcean

- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-arch-linux>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-debian-8>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-debian-7>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-13-10>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-16-04>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-centos-6>
- <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-centos-7>

Otras fuentes

- <https://httpd.apache.org/docs/current/vhosts/examples.html>
- <https://serverfault.com/questions/82306/apache-default-catch-all-virtual-host>
- https://subscription.packtpub.com/book/networking_and_servers/9781849514965/1/ch01lvl1sec14/setting-up-a-default-catch-all-virtual-host
- <https://www.jamescoyle.net/how-to/107-apache-2-catch-all-virtualhost>
- <https://www.digitalocean.com/community/questions/how-to-set-the-default-virtual-host-on-apache-2>
- <https://security.stackexchange.com/questions/83362/how-can-i-stop-someone-from-displaying-my-website-on-his-domain>

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>
