

## Cómo de crítico era el fallo en los servidores DNS

**Autor:** Iñaki Úcar (16 de julio de 2008)

<http://www.enchufa2.es/archives/como-de-critico-era-el-fallo-en-los-servidores-dns-i.html>

**Fecha Publicación:** 20 de julio de 2008

### Introducción

Ocurría la semana pasada: [Fallo crítico en DNS obliga a parchear toda Internet](#). No es que el fallo sucediera la semana pasada, ya que es un error tan viejo como Internet, pues proviene de la implementación del protocolo DNS. De hecho, mucho antes ya habían sido descubiertos por distintas personas pequeños errores en dicho protocolo. Sin embargo, fue la semana pasada cuando [Dan Kaminsky](#) descubrió que la unión de todos estos pequeños errores daba como resultado un enorme agujero de seguridad que afecta a toda Internet. Pero, ¿por qué? ¿Tan grave es? Pues sí, mucho. Y para entenderlo, pasaré a explicar por encima el funcionamiento del protocolo DNS.

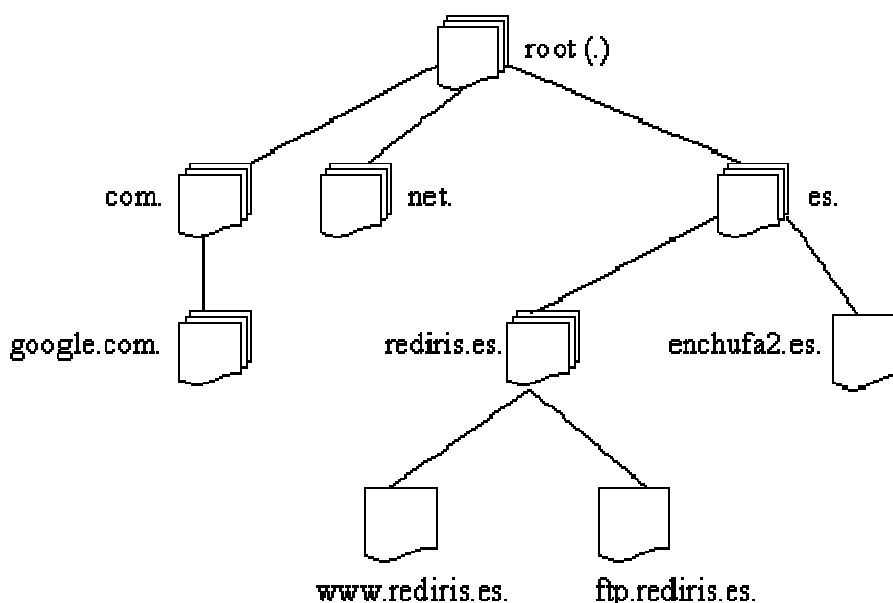
Cada computador conectado a Internet tiene asociado a él idealmente una **dirección IP** única (técnicamente [no es así](#), pero eso ya es otra historia) del tipo A.B.C.D, donde A, B, C y D son un número entre 0 y 255 (ej. 192.168.1.1). Esta IP es, digamos, como la dirección de nuestra casa para Correos. Entonces, cuando queremos acceder a Google en español, por ejemplo, tendremos que llamar a la dirección [66.102.9.99](#) (pinchad, y comprobad que os lleva a [www.google.es](#)). Pero, ¿cómo vamos a recordar las direcciones de todas las páginas que visitamos? Es mucho más fácil acordarse de “www.google.es”, ¿verdad? Pues es ahí donde entra en juego el protocolo DNS de resolución de nombres de dominio.

Nunca introducimos una IP directamente en el navegador: en lugar de eso, introducimos un nombre de dominio como el de “www.google.es”. Ante esta situación, es nuestro ordenador el encargado de buscar la IP correcta (66.102.9.99) para “llamar” a Google. Para eso se inventaron los servidores DNS, que son unas bases de datos repartidas a lo largo y ancho de Internet que albergan la traducción entre nombres de dominio y direcciones IP. Así que nuestro ordenador solicitará a uno de estos servidores de nombres (uno cercano que viene pre-configurado) que nos devuelva la dirección IP de “www.google.es”. Como hay millones y millones de máquinas conectadas a Internet, tener una traducción de todas ellas en un mismo lugar sería muy ineficiente por el volumen de datos a almacenar, por el coste que tendría cada búsqueda y porque cada máquina de Internet realiza miles de solicitudes DNS al día. Por ello, la escalabilidad es el gran triunfo del protocolo DNS.

### Jerarquización de los nombres de dominio

Esta inmensa base de datos de la que hablamos está partida en trozos y dividida entre montones de servidores DNS de manera jerárquica. En la cima del árbol, existen más de 130<sup>1</sup> **servidores DNS raíz** (root servers) repartidos por todo el mundo que albergan las direcciones IP de los servidores DNS encargados de los pocos cientos de **dominios de primer nivel** existentes, como por ejemplo los ‘.es’,

‘.com’, ‘.net’, etc. Éstos, a su vez, tienen recogidas las direcciones de los subdominios, como por ejemplo ‘rediris.es’, ‘google.es’, etc. Los últimos, a su vez, engloban más subdominios, como por ejemplo ‘www.rediris.es’, ‘ftp.rediris.es’, etc. Lo podemos ver mejor en el siguiente gráfico:



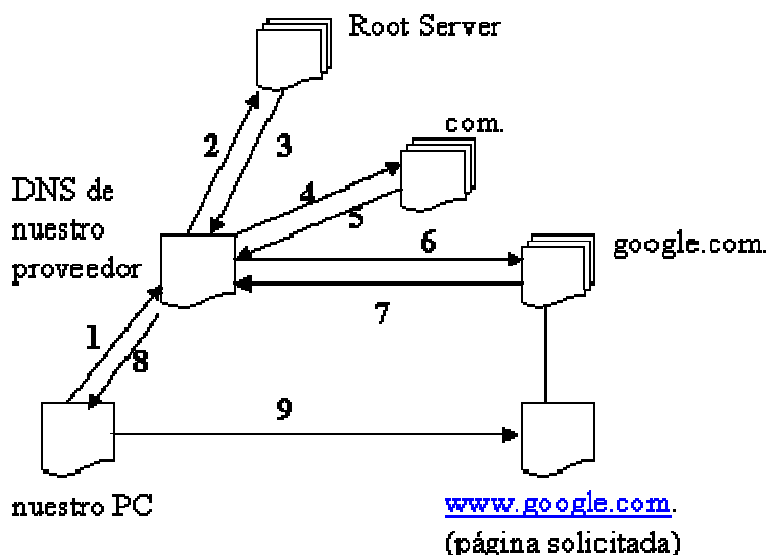
Como veo que esto va a quedar un poco largo, mejor lo divido en entregas. En esta primera anotación ya hay suficiente información por hoy. En futuras entregas repasaremos cómo funciona el protocolo DNS para llegar al meollo de la cuestión: la caché de los servidores DNS, en qué consiste el famoso agujero de seguridad y qué han podido hacer para paliarlo.

---

<sup>1</sup> Habitualmente se dice que hay 13 root servers identificados de la A a la M, y [esto da lugar a confusiones](#). En realidad, hay más de 130 localizaciones de estos root servers, más de 130 máquinas físicas que albergan root servers. Esto sería como decir que Google sólo tiene un servidor web. Evidentemente, para el volumen de tráfico que tiene que soportar, tiene réplicas de la misma web en máquinas diferentes por todo el mundo. Para comprobarlo, haremos lo siguiente: (desde Windows) vamos a **Inicio**, y le damos a **Ejecutar...** Tecleamos “nslookup” (sin las comillas) y presionamos **Enter**. En la nueva ventana, tecleamos “www.google.es” y volvemos a presionar **Enter**. Lo que hace este programa es solicitar al servidor DNS la IP de “www.google.es” y mostrarla por pantalla. En el campo “Addresses:” nos muestra las IP de la página solicitada. Como comprobaréis, hay más de una.

## Funcionamiento del protocolo DNS

En el siguiente gráfico se detalla el proceso de una resolución de dominio completa, aunque no siempre se requieren todos los pasos, ya veremos por qué:



En primer lugar, desde nuestro PC se genera una petición DNS hacia el servidor de nuestro proveedor de conexión. Dicha petición va en un paquete [UDP](#) y en él se pide la IP para la dirección “www.google.com” (en nuestro ejemplo).

Como nuestro servidor DNS no tiene conexión directa con la página solicitada, en principio, no sabe su dirección, así que tendrá que averiguarla. Para ello, realiza la misma consulta a uno de los Root Servers de los que sí sabe la dirección, puesto que son fijos.

El Root Server tampoco conoce la dirección de “www.google.com”, pero sí que conoce la dirección del DNS que alberga los dominios “.com”, así que se la entrega a nuestro servidor DNS, con lo cual ya está un poquito más cerca de lo que busca.

Nuestro DNS de nuevo realiza la misma consulta, pero esta vez al DNS encargado de los dominios “.com”. Éste último tampoco conoce la respuesta final, pero sí que conoce la dirección del DNS que alberga los dominios “google.com”...

... así que se la entrega.

Una vez más, nuestro DNS realiza la misma consulta al DNS ubicado en “google.com”. Y, por fin, él sí conoce la respuesta, puesto que se encuentra en su misma red.

Se dice que es “autoritativo” para esa dirección. Le entrega la respuesta final a nuestro servidor DNS.

Tras estos pasos que, aunque largos, se realizan en un lapso de tiempo pequeño, nuestro servidor ya conoce la respuesta a nuestra pregunta y nos la entrega.

Ahora ya podemos comunicarnos directamente con “www.google.es” puesto que conocemos su dirección IP.

Fácil, ¿no? Y diréis, ¿siempre se realizan todos estos pasos? Evidentemente, no es necesario. Una vez que un servidor DNS ha realizado una consulta y conoce una IP más, **la guarda en su memoria caché** para agilizar futuras peticiones, ya que es esperable que las direcciones IP no estén variando todo el tiempo. Como sí pueden variar, estas entradas de la memoria caché se guardan un tiempo máximo y luego se borran, por lo que habrá que volver a averiguarlas con los pasos anteriores.

### ***Buscando el agujero de seguridad***

Ahora bien, cuando recibimos la respuesta de un servidor DNS, nuestro ordenador confía en esos datos ciegamente. ¿Qué ocurriría si un atacante consiguiera hacerle creer al servidor DNS que la dirección IP para “www.google.com” (u otra página) es otra diferente a la real? Pues bien, en ese caso el servidor DNS guardaría en su caché esa asociación falsa para futuras consultas, y cuando nosotros solicitáramos la dirección de Google, nos haría entrega de esa dirección falsa en la cual nuestro ordenador confiaría ciegamente (porque no tiene manera de comprobar su validez).

En consecuencia, **un atacante lograría llevarnos a donde él quisiera de manera indetectable** mediante este método, denominado *cache poisoning* o envenenamiento de la caché. Por poner un ejemplo ilustrativo, imaginemos que alguien realiza una réplica de la página web de nuestro banco y la instala en su ordenador. Mediante envenenamiento de la caché del servidor DNS, logra que éste asocie el dominio de nuestro banco con la IP de su ordenador. Al ingresar nosotros en la web de nuestro banco, en realidad estaremos cargando la página alojada en el ordenador del atacante, y al enviar nuestros datos de conexión se los estaremos dando en bandeja de plata. Y lo peor de todo: ¡es imposible enterarse!

En la próxima entrega, profundizaremos en las causas de este agujero de seguridad y qué han podido hacer para solucionarlo.

### ***Envenenamiento de la caché***

Las consultas DNS no requieren conexión porque utilizan el protocolo UDP. Esto significa que se envía un único paquete con la consulta y se espera la respuesta en otro paquete UDP. Entonces, ¿nos fiamos de cualquier paquete de respuesta que llega? Obviamente no. Cada paquete entrante tiene una serie de indicadores que sirven para comprobar su validez: la dirección IP de quien envía el paquete, el puerto utilizado para la comunicación, un identificador (ID) aleatorio de 16 bits y el propio contenido del paquete. El servidor DNS se realiza las siguientes preguntas al recibir un paquete de respuesta:

¿Le acabo de preguntar algo a la IP que me envía este paquete?

¿Es una respuesta a la pregunta que yo he realizado?

¿El ID coincide con el que yo he enviado?

¿El puerto de llegada es el mismo desde el que yo pregunté?

Si las respuestas a todas estas preguntas son afirmativas, el servidor DNS acepta la respuesta como válida y la almacena en la caché. La primera respuesta válida que llega se acepta.

¿Qué tendría que hacer un atacante para explotar el agujero de seguridad? Para empezar, ser más rápido contestando que el servidor al que preguntó el DNS atacado. Esto es fácil: bastaría con realizar un [ataque DDoS](#) para inutilizar ese servidor y enviar muchos paquetes iguales al DNS atacado para asegurar el éxito.

¿Qué más? Cambiar la IP del que manda el paquete por otra cualquiera es algo trivial. Los únicos impedimentos que quedan por salvar son el ID y el puerto de destino, y aquí están los fallos graves. Me explico. Resulta que el ID aleatorio es de 16 bits, lo que nos da 65.536 posibilidades (muy pocas, porque 65.536 paquetes distintos se envían muy rápido), y lo que es peor: un servidor DNS **utiliza siempre el mismo puerto** para las consultas. Por lo tanto, es relativamente sencillo para un atacante averiguar qué puerto utiliza un servidor DNS y predecir el ID que utilizará para las consultas. Es decir, debido a las peculiaridades en la implementación del protocolo DNS, **es fácil para un atacante confundir a un servidor y envenenar su caché con datos falsos**, lo cual nos deja a los usuarios indefensos.

### ***Los grandes fabricantes solucionan el problema***

Desconozco qué medidas se habrán tomado al respecto, pero hago notar que tan sólo con permitir que cada consulta se haga desde un puerto aleatorio, las posibilidades aumentan notablemente -65.536 puertos posibles a los que hay que restar 1.024 puertos restringidos, y 65.536 ID posibles por cada puerto; **en total 4.227.858.432 posibilidades**-, y por consiguiente la dificultad para el atacante es muchísimo mayor.

En cualquier caso, **no hay por qué preocuparse**. Los grandes fabricantes - como Debian, Cisco, Infoblox, ISC, Juniper, Microsoft, Red Hat, Sun (Solaris)...- se pusieron manos a la obra y sacaron los parches pertinentes para sus servidores con gran rapidez. A día de hoy, seguro que todos los administradores del mundo han actualizado sus equipos y ya no hay nada que temer.

Más información en [esta nota técnica](#) del US CERT que describe la vulnerabilidad o en [esta otra nota](#) del SANS Institute.