# La vulnerabilidad del sistema de nombres (DNS) de Dan Kaminsky

Por Gabriel.Verdejo.Alvarez@gmail.com - Noviembre 2008

Este documento está orientado a la divulgación pública de conocimiento por lo que los aspectos técnicos, si bien deberían ser rigurosos, son tratados de forma accesible para cualquier persona con un mínimo de interés.

En este escrito se explica la vulnerabilidad del sistema de nombres (DNS) que Dan Kaminsky [www1] dio a conocer durante el verano de 2008 y se realiza un breve análisis de su impacto en Internet.

La estructura de este documento se divide en dos apartados claramente diferenciados, que todo y estar relacionados entre sí, permiten al lector ir directamente a la parte que más se ajuste a su nivel o interés en el tema.

- **Análisis técnico**: En este apartado se realizará una explicación amena del protocolo de resolución de nombres orientado a la famosa vulnerabilidad. Se comentarán los aspectos técnicos más relevantes y se complementará con un ejemplo práctico.

- **Impacto social**: En la segunda parte de este documento se describirá la cronología seguida por Dan Kaminsky en la gestión de esta vulnerabilidad y se reflexionará brevemente sobre el impacto teórico y real así como la gestión (¿modélica?) realizada por parte de la comunidad.

*"To learn, read. To know, write. To master, teach" – Yogi tea bag*

## 1. Análisis técnico de la vulnerabilidad de Kaminsky

### 1.1 Convenciones y definiciones

Siempre que se realiza un documento en un idioma distinto a los que originalmente incluyen la información, se produce una pérdida por el cambio de contexto. También hay que notar que muchas veces las traducciones de una misma expresión varían en función del traductor.

De esta forma y para mantener el espíritu riguroso de este manuscrito referenciaremos los vocablos y expresiones anglófonas sin traducirlos. Dejamos como trabajo para el lector la elección de la traducción que considere más adecuada.

Por cuestiones de eficiencia Internet se organiza jerárquicamente mediante direcciones IP, estas direcciones pueden ser vistas como los números de teléfono dónde el prefijo, por ejemplo, nos suele indicar a qué provincia estamos llamando.

Para facilitar el uso de Internet a las personas, generalmente es más fácil recordar un nombre que una ristra de números, se facilitó un servicio que permitiera el uso indistinto de las direcciones IP o sus nombres asociados. Al igual que pasa con el listín telefónico únicamente necesitamos recordar el nombre de la persona para poder obtener su número de teléfono.

El sistema de **DNS** (*Domain Name System*) [Stevens94][AL01][Liu02][www2][www3] es el encargado, entre otras funciones importantes, de establecer las correspondencias entre los nombres y las direcciones IP utilizadas en Internet.

A continuación enumeraremos una serie de conceptos vitales para entender cómo funciona este sistema de traducción de nombres a direcciones:

**Resolver** Es la parte cliente del servicio DNS. Su función es la de realizar las peticiones solicitando la dirección IP de un nombre.

**Nameserver** Es la parte servidora del DNS. Su función es la de contestar a las peticiones de nombres recibidas.

**Zone** Es una lista que contiene las parejas *<nombre>* / *<dirección IP>*. En realidad una zona puede contener más información, pero para nuestra explicación esta simplificación es suficientemente buena.

**Delegation** Cuando un *nameserver* no conoce el contenido de una *zone* pero sabe a quien preguntar, utiliza un mecanismo de delegación que consiste en contestar a la petición recibida indicando dónde se ha de buscar.

**GTLD** El *Global Top Level Domain* es el servidor que contiene toda la información referente a un dominio de último nivel.

En el caso del nombre www.google.com el GTLD correspondiente sería el de "*.com"*, en el caso de www.rediris.es sería el "*.es*" y así sucesivamente.

**Authoritative nameserver** Para cada una de las *zone* que puedan existir debe haber un servidor que se encargue de esta traducción.

**Recursive nameserver** Si un *nameserver* recibe una petición de resolución y no conoce la respuesta, realiza a su vez las peticiones que sean necesarias para acceder a la *zone* correspondiente que contenga la respuesta.

Esta capacidad recursiva es configurable, de forma que no siempre todos los servidores responden a todas las consultas. Por ejemplo un proveedor de Internet (ISP) que únicamente contesta peticiones de sus clientes.

**Resource record**

Además de la traducción *<nombre> / <dirección IP>* que ya hemos comentado, los *nameservers* también pueden proporcionar otros tipos de información útil al sistema [www60].

Ejemplos de este aspecto puede ser la obtención de los servidores de correo de un dominio (registros MX).

**Root server**

El sistema de DNS proporciona un conjunto básico de servidores de forma que a partir de ellos se pueda resolver cualquier petición. Esta lista de servidores básicos es "fija[1]" y está permanentemente almacenada en todos los *Recursive nameservers*.

Para investigar más y ver el formato y función que tienen estos servidores se puede consultar la bibliografía adjunta [www4][www5].

## 1.2 Seguimiento de una petición de resolución de nombre para www.rediris.es

Para ejemplificar el funcionamiento del DNS realizaremos una petición de "ping" a RedIris [www29]. Este comando se suele utilizar para la comprobación de la conectividad entre dos ordenadores.

Este ejemplo es del todo inofensivo y consiste simplemente en enviar una petición a www.rediris.es para que nos la retorne. En el caso de utilizar Windows debemos abrir una ventana de MS-DOS (figura 1) y para sistemas *Unix simplemente tener acceso a un shell o x-terminal (figura 2).

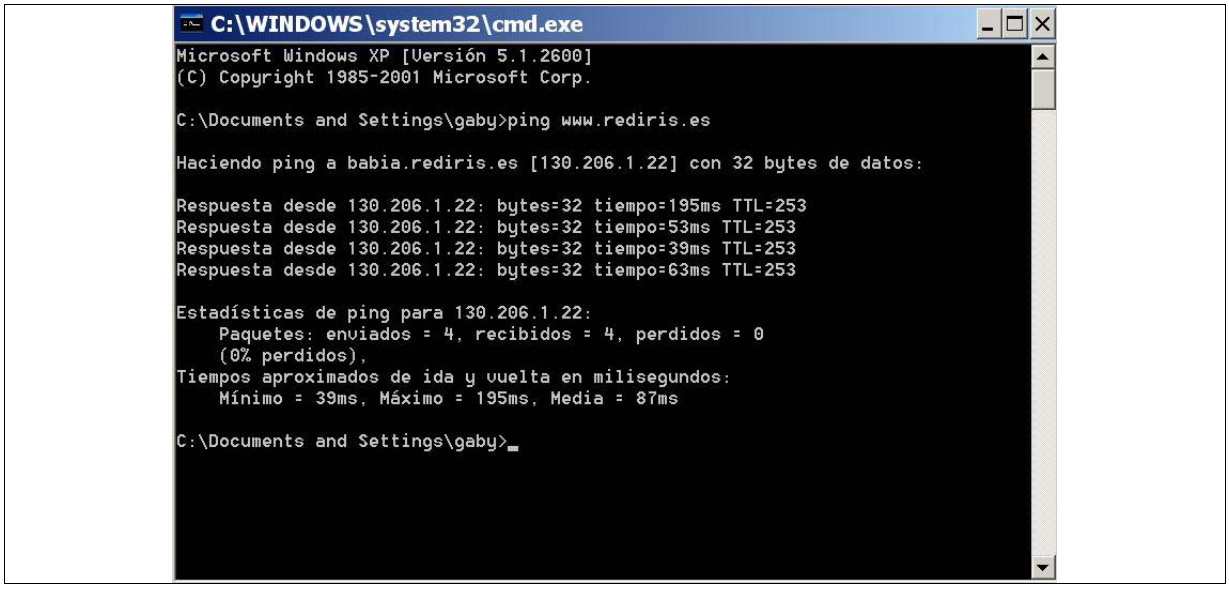


Fig. - 1: Sesión MS-DOS en Windows.

[1] Todo y que no es estrictamente cierto ya que algún *Root server* podría alguna vez cambiar su dirección IP, podemos asumir perfectamente para nuestros propósitos que estos son siempre los mismos con las mismas direcciones.

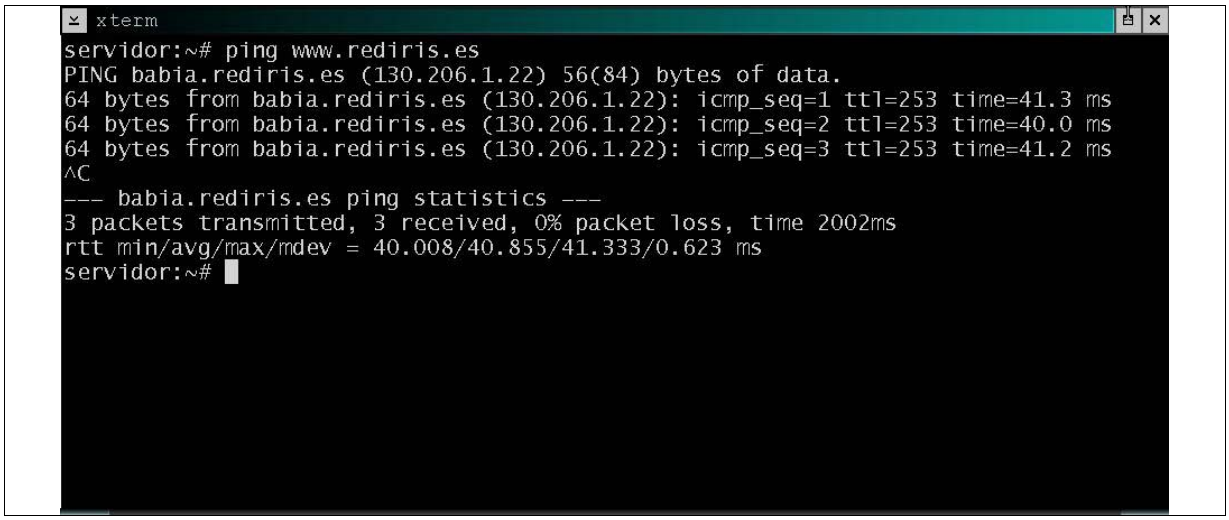


Fig. - 2: Xterminal en *Unix.

Lo primero que realizará nuestro ordenador es enviar una petición a nuestro servidor de nombres solicitándole la dirección IP de www.rediris.es (figura 3).

Obviamente para nuestro ejemplo debemos asumir que ninguna petición se encuentra en la memoria caché (lugar dónde se guardan las peticiones ya realizadas) de los servidores. Esto nos permitirá analizar en detalle todo el proceso.


Fig. - 3: Petición del cliente al servidor DNS.

Una vez solicitada la petición a nuestro servidor de nombres, este comprueba que no conoce la respuesta (figura 4) y consulta entonces su base de datos interna de *Root servers* para elegir uno al azar. Este mecanismo de selección aleatorio tiene como objetivo evitar el colapso de un único servidor balanceando las peticiones entre los distintos elementos del conjunto.

Una vez seleccionado uno de los servidores genera una petición para reenviar la consulta sobre el nombre de www.rediris.es. El *Root server* que recibe la petición comprueba que no conoce la respuesta y entonces responde proporcionando el *GTLD* que gestiona el dominio solicitado ("*.es*" en nuestro caso).

**NOTA**: Efectivamente un ataque exitoso a todos los *Root servers* pondría en serios aprietos a Internet. No profundizaremos más en este tema por desviarnos de nuestro objetivo inicial, pero sí incluimos en la bibliografía los detalles de varios intentos de ataques al servicio de DNS [www12][www13].

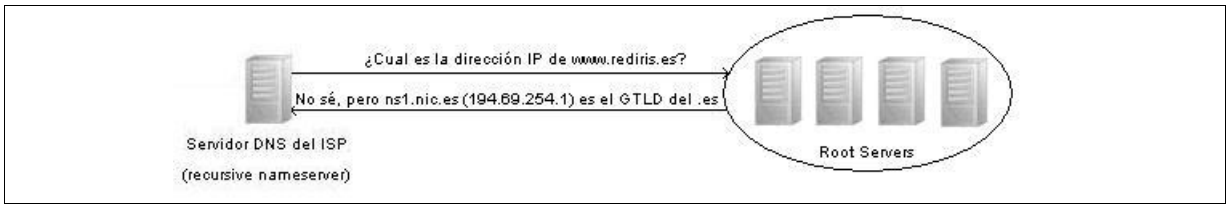


Fig. - 4: Petición del *Recursive nameserser* (ISP) al *Root server*.

Una vez recibido el *GTLD* del dominio "*.es*" nuestro *Recursive nameserver* le envía una nueva petición de solicitud (figura 5) de resolución de nombre para www.rediris.es. El servidor *GTLD* comprueba que no dispone de la respuesta y entonces busca en su base de datos qué servidores se encargan del dominio de segundo nivel de la petición (*rediris.es* en nuestro caso).

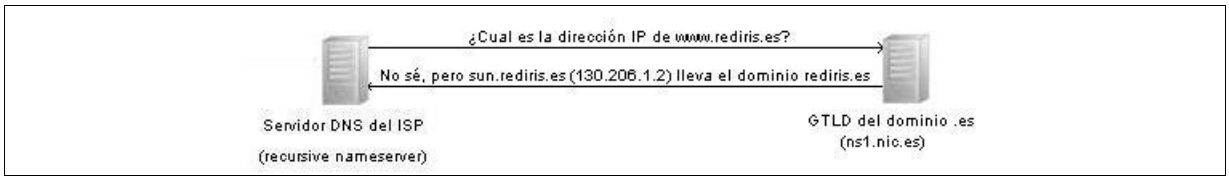


Fig. - 5: *Petición del Recursive nameserser* (ISP) al GTLD.

Una vez más nuestro *Recursive nameserver* recibe una contestación negativa a su pregunta junto con un nuevo servidor al que debe preguntar. De esta forma vuelve a realizar la petición de resolución de nombre (figura 6) para www.rediris.es y la envía al servidor encargado de gestionar el dominio "*rediris.es"*.

Finalmente recibe una contestación afirmativa dónde se explicita la dirección IP asociada al nombre www.rediris.es (130.206.1.22).

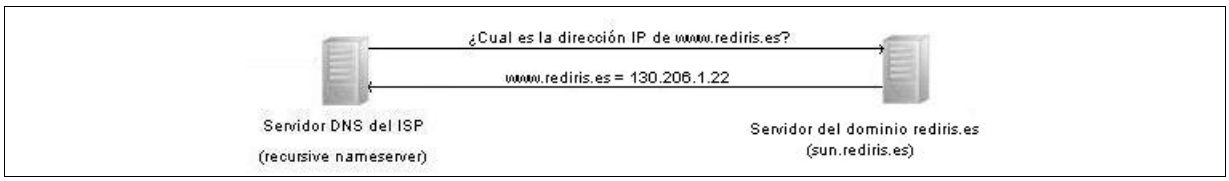


Fig. - 6: Petición del *Recursive nameserser* (ISP) al servidor del dominio "*rediris.es*".

Una vez resuelta la petición original por el *Recursive nameserver* este envía la respuesta (www.rediris.es = 130.206.1.22) al usuario inicial, que realizará a su vez una conexión a la dirección IP proporcionada (figura 7).

Cabe destacar que el sistema de caché de DNS se encarga de almacenar las últimas peticiones que se realizan por los usuarios de forma que no sea necesario realizar este proceso incontables veces cada segundo para cada usuario.

Por otro lado, el sistema de DNS permite marcar cada una de estas respuestas con un período de validez de forma que los nuevos cambios que se realicen sean "visibles" en un período de tiempo finito. El **TTL** (*Time To Live*) [www6][www7] es el mecanismo que gestiona la validez de los registros del sistema de DNS.

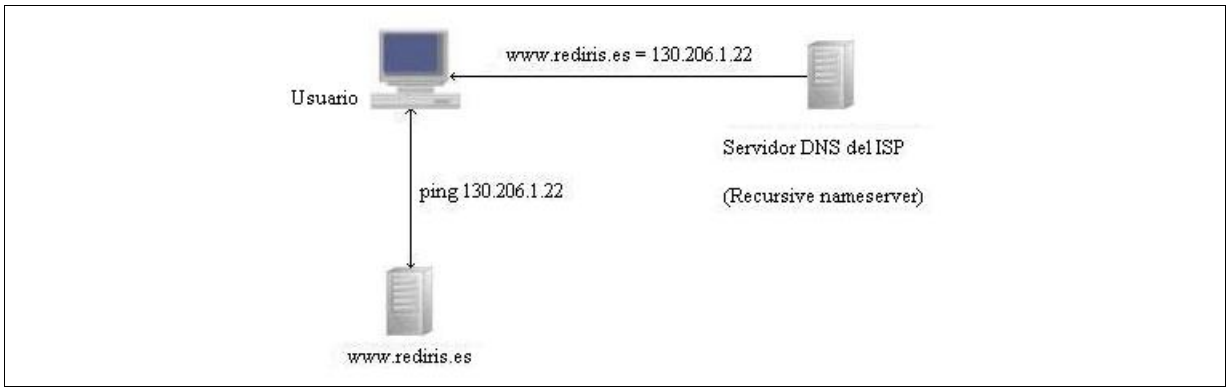


Fig. - 7: Respuesta del Servidor DNS del ISP a nuestra petición de www.rediris.es

### 1.3 Análisis de las respuestas generadas en la resolución de nombre de www.rediris.es

Una vez realizada la explicación simplificada del proceso de resolución de nombres, debemos profundizar un poco más para entender cómo funciona esta vulnerabilidad y cómo afecta al servicio de DNS. Para seguir nuestro análisis deberemos asumir que el lector está mínimamente familiarizado con el protocolo TCP/IP [Stevens94][www14][www15].

Para los ejemplos que utilizaremos necesitaremos la herramienta **DIG** (*Domain Information Groper*) [www8][www9] que nos permitirá simular el comportamiento de un *Resolver*. En caso de que no se tenga acceso a este programa se pueden utilizar versiones on-line como [www10] o [www11].

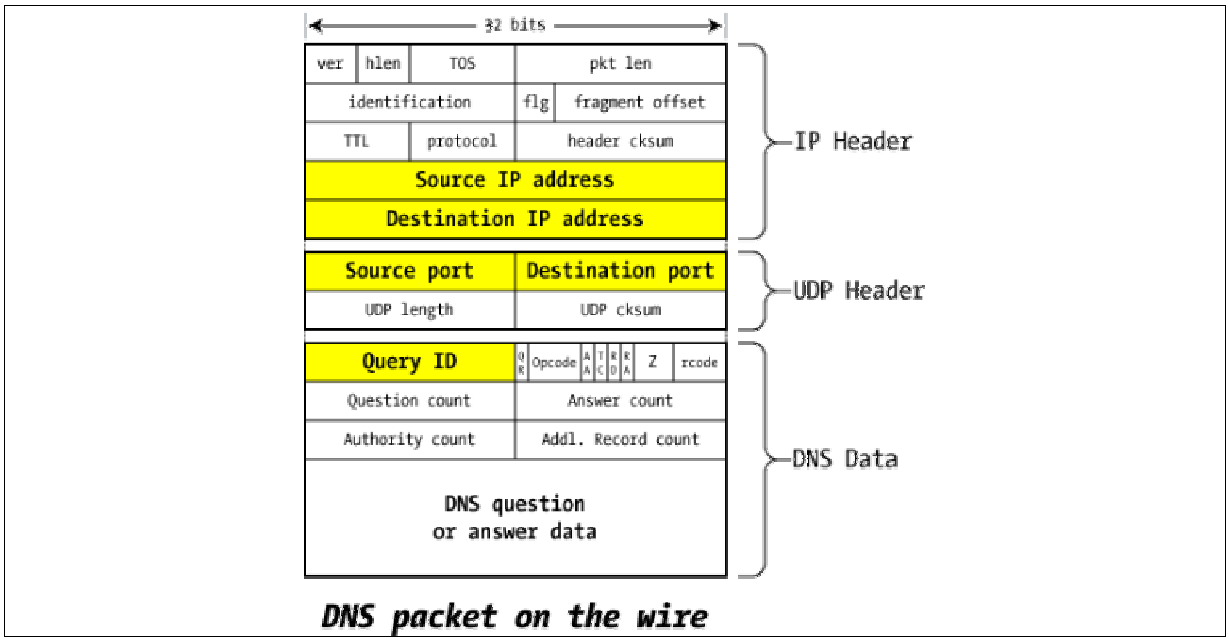


Fig. - 8: Prototipo de paquete IP que utiliza el protocolo DNS.

Todas las peticiones de resolución de nombres que se realizan a los diferentes servidores de nombres circulan por Internet encapsuladas en datagramas IP (figura 8 [www6]).

De todos los campos que contiene el datagrama IP, para nuestro estudio, la parte más relevante de estas peticiones hace referencia al *Query ID* que es el identificador de petición que crea el cliente cada vez que desea realizar una resolución de nombres. De esta forma el cliente puede solicitar simultáneamente varias peticiones para nombres distintos (cada una con un *Query ID* diferente) y no confundir las respuestas.

Este identificador es muy importante ya que no sólo nos permite disfrutar de un mecanismo que permite solicitar varias peticiones de resolución de nombres concurrentes (por ejemplo abrir un navegador con dos pestañas o tabs distintas, una a www.google.es y otra a www.rediris.es) si no que también nos protege de respuestas no solicitadas.

Al recibir una respuesta DNS lo primero que se realiza es una comprobación de los *Query ID* pendientes. Si existe una petición pendiente de resolverse con ese identificador, se acepta la respuesta. Si no existe ninguna petición pendiente o la petición no coincide exactamente con la que formuló el cliente, el paquete recibido es descartado inmediatamente.

A continuación analizaremos los mensajes que se van intercambiando en este proceso de resolución de nombres ya que es aquí dónde reside la clave del ataque Kaminsky. Podemos observar un ejemplo de respuesta a la petición de resolución del nombre www.rediris.es en la figura 9.

```
gaby@servidor:~$  dig www.rediris.es

; <<>> DiG 9.3.4-P1.1 <<>> www.rediris.es
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21940
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 7

;; QUESTION SECTION:
;www.rediris.es.                 IN     A

;; ANSWER SECTION:
www.rediris.es.       2785   IN     CNAME   babia.rediris.es.
babia.rediris.es.     2785   IN     A       130.206.1.22

;; AUTHORITY SECTION:
rediris.es.           814    IN     NS      sun.rediris.es.
rediris.es.           814    IN     NS      scsnms.switch.ch.
rediris.es.           814    IN     NS      ns02.fccn.pt.
rediris.es.           814    IN     NS      chico.rediris.es.

;; ADDITIONAL SECTION:
sun.rediris.es.          3321   IN     A       130.206.1.2
ns02.fccn.pt.            2677   IN     A       193.136.2.228
ns02.fccn.pt.            2408   IN     AAAA    2001:690:a80:4001::200
chico.rediris.es.        3320   IN     A       130.206.1.3
scsnms.switch.ch.     110003 IN     A       130.59.1.30
scsnms.switch.ch.     110003 IN     A       130.59.10.30
scsnms.switch.ch.     110003 IN     AAAA    2001:620::1

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Oct 10 23:16:04 2008
;; MSG SIZE  rcvd: 298
```

Fig. - 9: Petición del cliente al servidor DNS usando DIG.

Si analizamos el mensaje de la figura 9 podemos observar claramente una serie de apartados que nos aportan diferente información relevante [AL01]:

- **Question Section:** Contiene la petición original que se ha realizado. La especificación actual impide que exista más de una petición en este campo.

- **Answer Section:** En este apartado se incluyen los *Resource records*. Este apartado sí puede contener varios registros correspondientes a la consulta, por ejemplo en casos de *multihoming*[2].

- **Authority Section:** Explicita los registros de los servidores de nombre (*NS records*) referentes a la petición solicitada.

- **Additional Section:** Contiene la información que permite completar el resto de secciones. Por ejemplo en el caso de existir varios servidores de nombres se explicitan sus direcciones IP para que puedan ser contactados.

### 1.4 DNS poisoning

El envenenamiento de la caché DNS (*DNS poisoning*) es un mecanismo que busca falsear las correspondencias <*nombre*> / <*dirección IP*> que almacenan los *Nameserver*. Este tipo de ataques busca redireccionar todo el tráfico de los clientes hacia los servidores que controla el atacante.

Es importante diferenciar este tipo de ataques del *Phishing* [www17][www18] dónde se intenta engañar al incauto enmascarando la realidad tras unos engaños más o menos sofisticados de forma que este "pique el anzuelo".

Un ejemplo típico de *Phising* podría ser pedir dinero para una ONG por alguna catástrofe natural adjuntando un enlace a http://www.cruzroja.es que en realidad apunta a una copia fraudulenta del tipo http://www.cUrzroja.es que el atacante controla.

¡En el caso del *DNS poisoning* el usuario a todos los efectos se está conectando a http://www.cruzroja.es ya que su DNS ha resuelto la petición "correctamente"! Como podemos observar este tipo de ataques son potencialmente mucho más peligrosos ya que afectan a todos los usuarios que utilicen el servidor de nombres "infectado" (imaginemos el caso de un DNS de un gran ISP).

Afortunadamente no es tan sencillo "engañar" al sistema de DNS ya que para que una petición pueda ser falseada debe cumplir con estos cuatro requisitos:

1. El *Query ID* recibido en la respuesta ha de corresponderse con un identificador de consulta pendiente en el cliente. En la práctica esto significa que el atacante debe adivinar este identificador, de ahí la importancia que sea escogido de forma aleatoria.

2. El apartado *Question section* ha de corresponderse exactamente con la pregunta original formulada por el cliente. Si hemos preguntado por www.rediris.es no aceptaremos respuestas a www.google.es.

[2] Casos en los que varios nombres están asociados a una misma dirección IP.

3. Los apartados *Authority section* y *Additional section* de la respuesta han de pertenecer al mismo dominio que la petición original. Esta restricción exige que si estamos preguntando por www.rediris.es en la respuesta no se añada una refeerncia para otro dominio (por ejemplo www.google.es) que no sea el solicitado expresamente.

4. La respuesta, ver formato de la cabecera UDP de la figura 8, ha de llegar por el mismo puerto que fue solicitada. Al igual que ocurre con el *Query ID* que nos permite simultanear varias peticiones, el protocolo UDP también nos permite varias conexiones discriminando por el valor del puerto UDP de origen (*source port*).

Una respuesta a una petición de resolución de nombres que cumpla las restricciones anteriores es considerada válida y por tanto se utilizará en la resolución de nombres correspondientes a ese dominio y en la caché del servidor (al menos por el período establecido por el TTL).

Para adivinar el *Query ID* podemos utilizar varias técnicas dependiendo de la versión y del servidor de nombres. Actualmente existen decenas de programas que realizan las funciones de servidores de DNS [www19][www20] y cada una de ellas tiene incontables versiones. De esta forma como siempre sucede en estos casos, el sistema de prueba y paciencia es el que mejor funciona.

Por otro lado cabe recordar que como ya se ha indicado anteriormente las respuestas que no son "correctas" simplemente se ignoran, permitiendo al atacante usar la fuerza bruta para generar cientos o miles de peticiones con el objetivo de mejorar la probabilidad de acierto.

En el caso de versiones antiguas de servidores DNS, el *Query ID* se generaba simplemente incrementando en uno el número anterior. Este sistema poco sofisticado permite que la probabilidad de adivinar el identificador sea muy alta ya que "simplemente debemos espiar" la red o provocar que nos realice una petición de resolución de nombres para adivinar el número a partir del cual debemos probar.

La clave consiste en "obligar" al cliente a realizar una petición a un *Nameserver* que controlemos. Existen muchas maneras creativas de realizarlo, por ejemplo podemos enviar un email o crear una página web con una referencia a nuestro dominio de forma que casi de forma automática y sin que el cliente se de cuenta nos envíe la información que necesitamos.

Para nuestro ejemplo podemos suponer que tenemos el control del dominio ficticio de Internet "*dominiomalo.com*". Nuestro objetivo es que los distintos clientes se conecten para resolver la dirección IP asociada al nombre "*malisimo.dominiomalo.com*".

**NOTA**: Obviamente todo este sistema queda enormemente simplificado si tenemos acceso a "espiar" la red por la que circulan las peticiones del cliente.

Una vez que el cliente realiza la petición a un servidor que ya controlamos, podemos obtener el *Query ID* y el *puerto* UDP utilizados. Además en este momento tenemos una petición legítima de un cliente a un dominio que controlamos y que es *Authoritative* para "*dominiomalo.com*"

Para iniciar nuestro ataque realizamos una petición de resolución al *Nameserver* que utiliza

nuestra víctima con el nombre que deseamos suplantar (www.rediris.es en nuestro ejemplo).

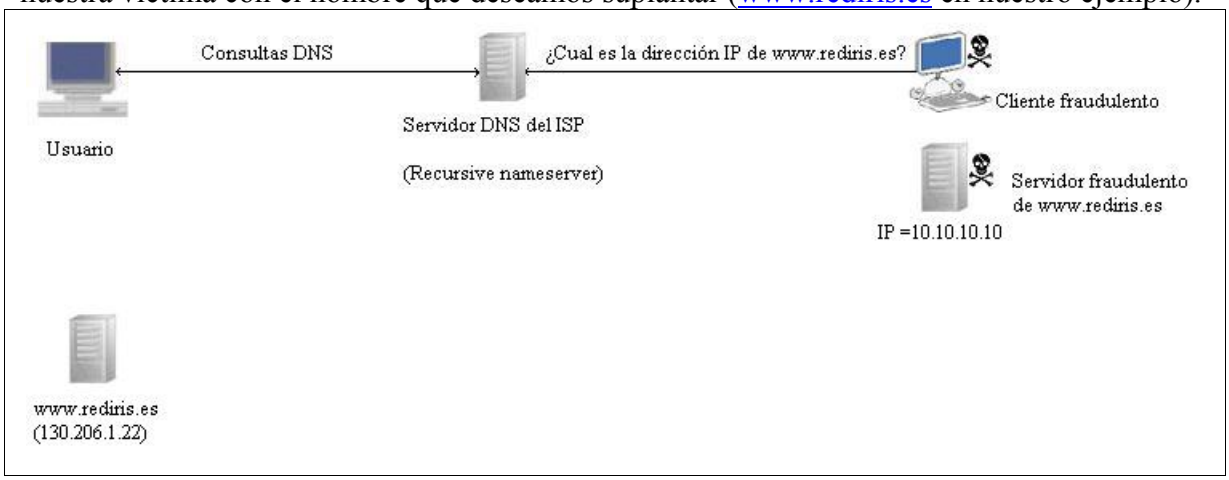


Fig. - 10: Solicitud de resolución de nombre por parte de un cliente fraudulento.

El atacante, una vez que ha elegido el nombre a suplantar, sabe que en un breve período de tiempo (ver procedimiento explicado en el punto 1.2 de este documento) el *Recursive nameserver* solicitará el nombre del servidor GTLD para obtener el servidor del dominio "*rediris.es*" (ver figura 10) que en nuestro caso es "*sun.rediris.es*".

**NOTA**: Obviamente debemos asumir que la respuesta a nuestra petición fraudulenta no se encuentra en la caché del servidor, ya que entonces todo el proceso de resolución de nombre no se realiza y nuestro ataque deja de ser efectivo.

Es en este mismo momento dónde el atacante empieza a bombardear al *Recursive nameserver* con miles de respuestas falsas creadas usando sucesivos *Query ID*[3] y completándolas con los datos del puerto UDP y las secciones que ya conoce (ver figura 11).

El objetivo del atacante es doble:

- Acertar el *Query ID* con alguna de las peticiones fraudulentas.
- Que la respuesta fraudulenta llegue antes que la auténtica ya que una vez "contestada" el *Query ID* deja de estar pendiente y por tanto se descartan segundas respuestas.

Cabe destacar que debido a que el atacante necesita asegurarse que sus respuestas fraudulentas lleguen antes que las legítimas y que cuantas más genere más posibilidades de éxito tendrá, el consumo de ancho de banda para este ataque puede ser grande. Existen diferentes argucias que se pueden emplear como ir atacando servidores próximos geográficamente (por ejemplo consiguiendo acceso a una o varias máquinas de la red del ISP y utilizarlas como lanzaderas) o realizar respuestas fraudulentas coordinadas desde más de una fuente.

Otra posibilidad es la de "silenciar" al servidor que envíe la respuesta legítima generando un ataque de denegación de servicio Distribuido (DDOS) [www22][www23][www24] de forma que el atacante pueda probar tranquilamente todas las combinaciones y casi asegurarse al 100% el éxito del ataque.

[3] No necesariamente consecutivos ya que el atacante podría ser capaz de predecir la secuencia "aleatoria".

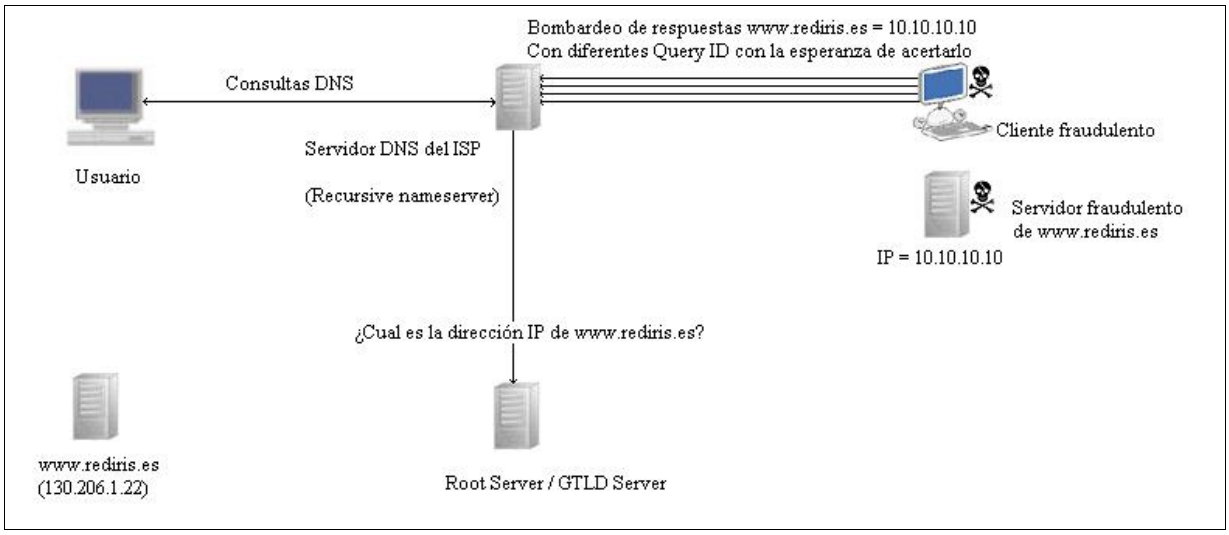


Fig. - 11: Bombardeo de respuestas fraudulentas para suplantar la identidad.

Una vez conseguido su objetivo la respuesta fraudulenta se almacena en la caché del servidor y es utilizada para todos los clientes que la soliciten. Es más, como la petición aceptada es la nuestra ¡podemos indicar en la respuesta cual es el *Authoritative nameserver* para todo el dominio[4]! (ver figura 12).

Pese a que muchos servidores eligen los campos de *UDP port* y *Query ID* de forma aleatoria, la realidad suele ser que las secuencias son predecibles y por tanto reproducibles. Esto es lo que hace tan importante la paciencia del atacante y el trabajo previo de investigación que permita averiguar que secuencia utiliza nuestra víctima.

La implementación de funciones realmente aleatorias es un problema clásico en informática que ya ha dado varias vulnerabilidades "gloriosas". Podemos destacar una reciente en la distribución de Linux Debian [www21].

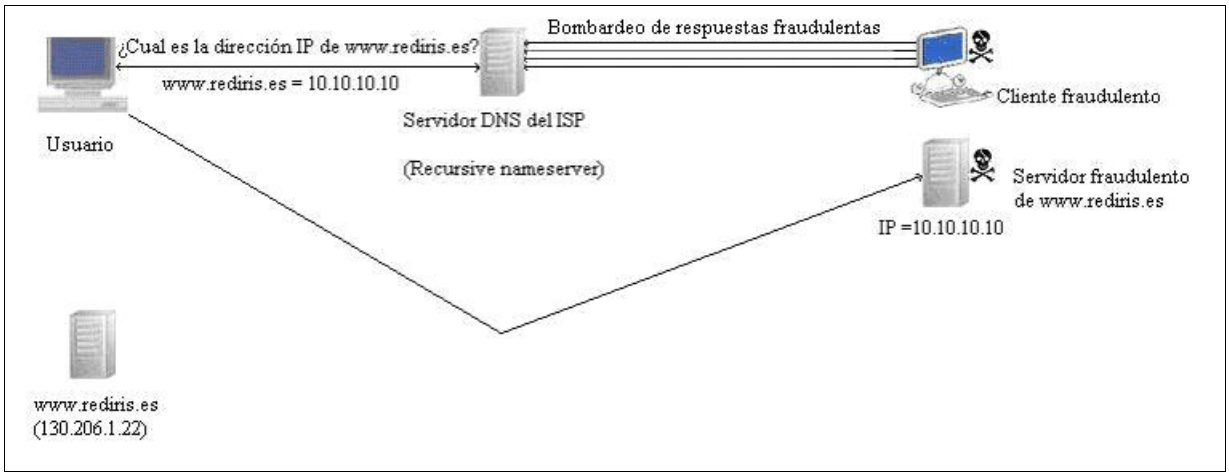


Fig. - 12: Suplantación de www.rediris.es para todos los usuarios del ISP.

[4] Este aspecto es importante y lo retomaremos en el ataque Kaminsky [www16].

## 1.5 La vulnerabilidad de Kaminsky

Como hemos analizado en el apartado anterior gran parte de la problemática se solventa con el uso de funciones realmente aleatorias. Sin embargo cabe destacar (ver figura 8) que el campo del *Query ID* es de 16 bits con lo que el espacio de posibles combinaciones es de 65535 posibilidades.

Si bien es un número razonable de posibilidades actualmente (el potencial atacante debería generar decenas de miles de respuestas antes de que llegara la respuesta legítima) la posibilidad de que las funciones no sean tan aleatorias como se pretende o el incremento constante del ancho de banda disponible hacen que no debamos perder de vista este aspecto. Por otro lado la longitud de este campo no puede ser variada, ¡es un estándar que usa toda Internet!, sin modificar todos los servidores DNS.

Hasta ahora hemos visto que "secuestrar" un nombre concreto en el sistema de DNS requiere un cierto volumen de trabajo. Si nuestros objetivos son más ambiciosos (redireccionar el correo, todos los servidores web del dominio...) podemos observar cómo la cantidad de trabajo a realizar lo hace prácticamente inviable.

La vulnerabilidad descubierta por Dan Kaminsky [www16][www25][www26][www27] da una vuelta de tuerca más al procedimiento anterior creando una respuesta fraudulenta que nos permitirá secuestrar el *Authoritative nameserver* y por tanto ¡**todo el dominio**!.

El primer paso consiste en configurar un servidor de DNS de realice las funciones de *Authoritative nameserver* del dominio a secuestrar (*rediris.es* en nuestro caso). Obviamente en este DNS fraudulento tendremos configurados todos los registros (*Resource records* [Liu02]) de forma que se resuelvan a nuestras direcciones IP que contendrán las copias de las webs y servicios a secuestrar.

Dependiendo de la sofisticación que deseemos alcanzar, únicamente necesitamos modificar las respuestas a un nombre concreto (por ejemplo www.rediris.es) y resolver el resto con las direcciones IP legítimas o replicar toda la estructura del dominio lo que nos permitiría entre otras cosas acceso a todo su correo.

**NOTA**: No hay ningún problema en que cualquier persona se configure su propio servicio de DNS que resuelva cualquier dominio de Internet. Sin embargo no es muy útil ya que nadie (excepto nosotros mismos) lo utilizará. Con Kaminsky esto cambia.

A continuación (figura 13) debemos forzar la resolución de un nombre perteneciente al dominio a secuestrar que no se encuentre en la caché del DNS atacado. Una manera de asegurarnos este punto puede ser creando nombres aleatorios que pertenezcan al dominio por ejemplo ***x12123424****.rediris.es* o ***x32321****.rediris.es.*

Una vez generadas las peticiones de resolución de nombres maliciosas, al igual que en el *DNS poisoning*, el atacante bombardea al DNS con una serie de respuestas fraudulentas con el objetivo de adivinar el *Query ID*. Sin embargo estas respuestas no contienen como antes la dirección IP del nombre solicitado, si no que indican que no se conoce la dirección IP y que ha de preguntar al *Nameserver* que controlamos (ver figura 14) para obtener la respuesta final (la explicación del proceso de resolución se encuentra en el punto 1.2 de este documento).

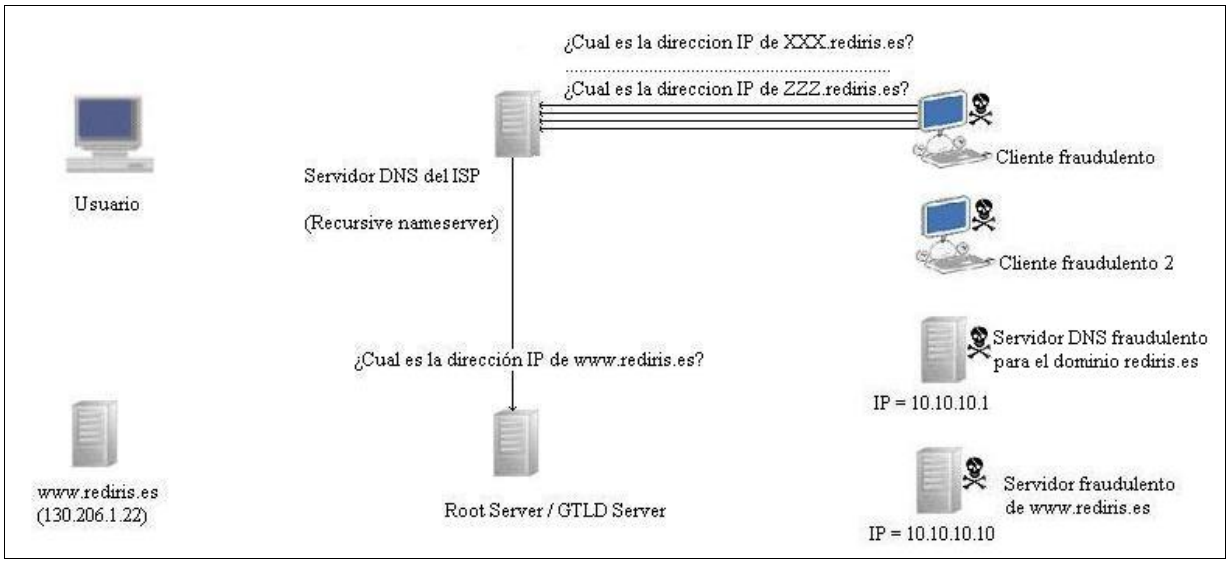


Fig. - 13: Forzando la resolución de nombres aleatorios del dominio *rediris.es*

Además como parte de esta respuesta fraudulenta se adjunta en el campo *Additional section* la dirección IP del servidor ilegítimo de DNS que controlamos (ver figuras 9 y 14). De esta forma podemos incluso dar el nombre real del servidor del dominio atacado ya que falseamos su dirección IP.

A partir de ese momento, ya que queda almacenada en la caché del servidor DNS del ISP, todas las peticiones de resolución de nombres asociadas al dominio *rediris.es* pasaran a resolverse por el servidor DNS del atacante. Razonablemente todas las respuestas que genere el servidor fraudulento de DNS proporcionaran un valor TTL lo más alto posible.

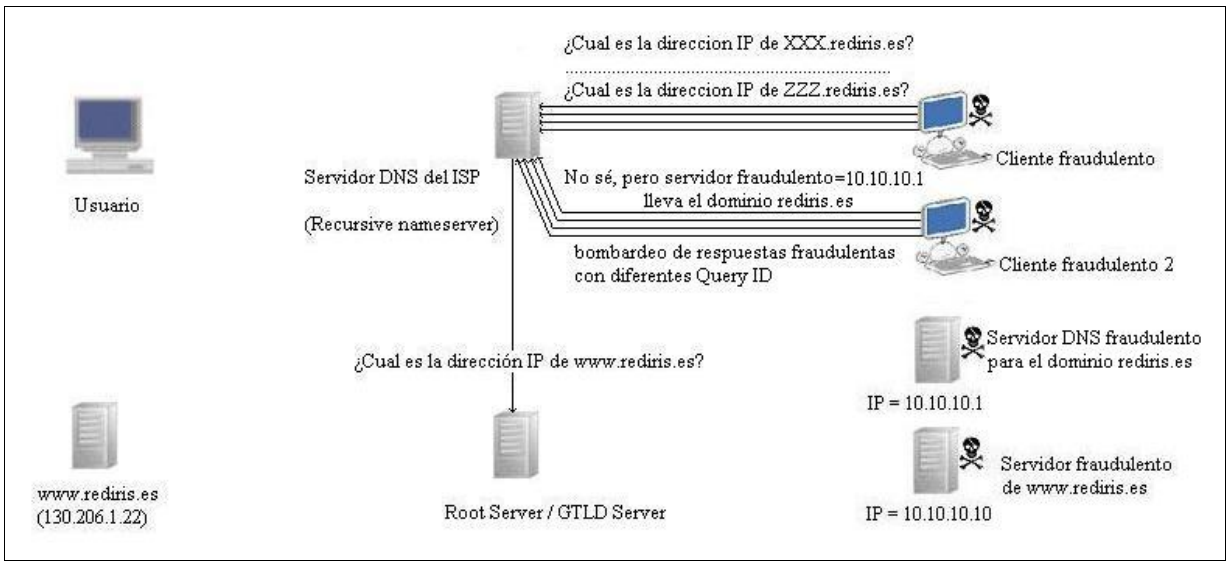


Fig. - 14: Bombardeo de respuestas maliciosas a las peticiones fraudulentas realizadas.

Este ataque puede optimizarse generando varias peticiones de resolución de nombres aleatorios que no existan en la caché. De esta forma si por ejemplo el atacante puede generar 50 respuestas fraudulentas antes de que el DNS legítimo responda, puede ir probando continuamente con nombres distintos hasta que finalmente su petición sea aceptada.

Un posible ataque DDOS [www23][www24] al servidor o servidores DNS que generan la respuesta legítima con el objetivo de consumir su ancho de banda o incluso colapsarlos podría incrementar el número de respuestas fraudulentas que el atacante podría realizar.

Cabe destacar que este ataque conceptualmente también podría generalizarse perfectamente para secuestrar todo un GTLD (*Global Top Level Domain*) con lo que por ejemplo todos los subdominios de *.es* o de *.com* estarían comprometidos, lo que equivale en la práctica a dominar Internet.

### 1.6 ¿Cómo mitigar la vulnerabilidad de Kaminsky?

Como ya hemos visto anteriormente, la atenuación del impacto que produce este ataque pasa por el uso de funciones realmente aleatorias de forma que se dificulte al máximo la posibilidad de acertar los campos claves (ver figura 8):

*Query ID*: El identificador de la petición de resolución de nombres es un campo de 16 bits que nos deja un espacio de 65536 posibilidades.

*Source port* (UDP): El puerto de origen de la comunicación UDP es también un número de 16 bits que nos deja un espacio de 65536 posibilidades.

Si utilizamos ambos campos para extender el espacio de claves tenemos 32 bits lo que nos da un total de 4.294.967.296 alternativas.

Desgraciadamente es complicado reservar todo el espacio de puertos UDP de un sistema ya que muchos otros protocolos y programas lo utilizan como sistema de comunicación (NFS, Skype, Windows...). De esta forma nuestro espacio real de posibilidades queda reducido considerablemente, aunque en la actualidad es considerado "seguro".

En el caso de Microsoft [www28] para sus sistemas Windows ha optado por reservar un máximo de 2500 puertos UDP aleatorios lo que nos deja un espacio de algo más de 160 millones de posibilidades.

2500 (*UDP*) * 65536 (*Query ID*) = 163.840.000

Otra posibilidad bastante ingeniosa que se ha planteado, y además es compatible con la anterior, es la de utilizar el *bit 0x20* [www49][www108][www109][www110]. En las letras del abecedario (a-z, A-Z), su representación ASCII permite "jugar" con el uso de este bit para crear combinaciones de mayúsculas y minúsculas.

www.rediris.es = WwW.rEdIrIs.Es = wWw.ReDiRiS.eS

De esta forma, como la petición de nombres es insensible a mayúsculas o minúsculas y como la respuesta debe incluir la petición original realizada, el posible atacante no sólo debe adivinar el QueryID y el Puerto UDP, si no que debe adivinar también la combinación ganadora de mayúsculas y minúsculas.

# 2. Impacto social

## 2.1 Cronología oficial

A continuación realizaremos una reconstrucción tan pormenorizada como sea posible del proceso que se ha seguido en la gestión de esta vulnerabilidad. Este seguimiento se inicia antes de empezar el año 2008 y abarca hasta finales de Agosto de ese mismo año. La selección de este período de tiempo nos permitirá conocer cómo y quien descubrió esta vulnerabilidad, que proceso de gestión se realizó hasta su anuncio público el mes de Julio y finalmente la evolución y respuesta de la comunidad.

La información utilizada en este apartado se ha obtenido principalmente de los propios comentarios de Dan Kaminsky accesibles desde su Blog personal [www16] así como de varias entrevistas aparecidas en distintos medios de comunicación [www50][www54] [www55][www56].

A finales del año 2007 Dan se encontraba trabajando en un proyecto que implicaba la elección de la ruta más rápida para servir datos desde distintos servidores web en el contexto de *triangular routing* [SJ08][www59].

Como parte de estas pruebas empezó a explorar distintas posibilidades relacionadas con el servicio de DNS y los *Resource records* [www60]. Dentro de los experimentos que realizó empezó a comprobar la posibilidad de actualizar los registros de dominio del DNS mediante el uso de nombres aleatorios para acelerar el proceso del intercambio de datos.

A inicios del 2008 [www38] y como fruto de estas pruebas descubre que es posible modificar la caché de los servidores DNS para inyectar nueva información mediante respuestas creadas específicamente saltándose el procedimiento ordinario de resolución de nombres. Una vez verificado el procedimiento y tras comprobar que con una implementación de su programa realizada en lenguaje C puede conseguirse en menos de 10 segundos, toma conciencia del problema de seguridad que implica esta vulnerabilidad.

El 20 de Febrero, una vez verificado el impacto potencial y tras comentarlo con gente de confianza perteneciente a su entorno más cercano, Dan contacta con Paul Vixie [www61] [www62] del Internet Systems Consortium, ISC [www63]. Es a partir de este punto dónde empieza el intercambio de emails entre las distintas personalidades del sistema de DNS y las empresas más importantes del sector (CISCO, Microsoft, Yahoo...).

El día 31 de Marzo un equipo de emergencia cuidadosamente seleccionado y formado por 16 personas de todos los rincones del mundo se reúne en los cuarteles generales de Microsoft. La empresa de Bill Gates generosamente accede a proporcionar toda la infraestructura necesaria para que este grupo pueda evaluar el impacto real de esta vulnerabilidad y realice el trabajo necesario para solventarlo.

Finalmente el martes 8 de Julio de 2008, que pasará a la historia de Internet como la primera gran acción coordinada por motivos de seguridad en la red, en una conferencia de prensa multitudinaria se presentan simultáneamente las actualizaciones para todos los servicios de DNS relevantes en Internet [www64][www65].

El día 9 de Julio Dan Kaminsky publica en su blog personal [www30] una entrada con el título "*An astonishing collaboration*" en la que señala la existencia de una terrible vulnerabilidad en el servicio de nombres (DNS). En su escrito detalla de forma general cuatro puntos principales que hacen referencia a esta vulnerabilidad así como al proceso que se ha seguido para minimizar su impacto en Internet.

1. Es una vulnerabilidad multi-plataforma, es decir, es independiente del hardware en el que funcione.

2. Es un fallo de diseño del protocolo DNS y por tanto es el mismo para casi todas las implementaciones de este servicio, independientemente del sistema operativo.

3. Se ha conseguido que, para la mayoría de los sistemas existentes, se publiquen las versiones corregidas del software simultáneamente.

4. Por primera vez en la historia de Internet se ha conseguido una acción coordinada ante una vulnerabilidad tan importante.

También agradece la colaboración de grandes empresas como Microsoft, CISCO o Yahoo en la gestión de esta vulnerabilidad. Detalla incluso el caso de Yahoo que se ha visto obligado a abandonar la versión 8 del programa BIND [www31] para utilizar la última actualización de la versión 9 que corrige este problema[5].

Además solicita a la comunidad de Internet una moratoria de 30 días para evitar las especulaciones públicas y dar tiempo para que los correspondientes administradores de sistemas puedan actualizar sus servidores DNS.

Finalmente emplaza a toda la comunidad a la conferencia que realizará en Las Vegas dentro del Defcon [www33][www34] el día 6 de Agosto de 2008 dónde desvelará los detalles de esta vulnerabilidad.

El lunes 21 de Julio Thomas Dullien, conocido por su alias Halvar Flake, publica una entrada en su blog [www53] en la que cuestiona la utilidad de esta moratoria de especulaciones públicas con el argumento de que hackers profesionales sí van a especular y ganar 30 días de calma puede crear una sensación de falsa seguridad.

De esta forma Dullien apunta la posibilidad de que este fallo esté relacionado directamente con la generación de respuestas espurias por parte de un atacante. Dada una consulta de un servidor DNS el atacante le enviará respuestas fraudulentas con el objetivo de suplantar al servidor de nombres legítimo que resuelva el dominio. A partir de este instante decenas de blogs de seguridad se hacen eco de esta posibilidad y empiezan a desarrollar esta teoría.

Este mismo día, cabe destacar que estamos hablando de diferentes zonas horarias, en el blog de Thomas Ptacek [www52] empleado de la empresa de seguridad Matasano [www68] (que ya estaba al corriente de la vulnerabilidad Kaminsky) aparece una entrada que describe con precisión el ataque. Si bien a las pocas horas se retira el comentario y se solicita públicamente disculpas, ya es tarde, miles de sitios en Internet se han hecho eco de este post.

[5] Cabe destacar que el software de DNS de Dan J. Bernstein [www32] no se ve afectado por esta vulnerabilidad, demostrando que un buen trabajo de ingeniería del software es posible en informática.

El día 22 de Julio y tras desvelarse la vulnerabilidad Kaminsky la práctica totalidad de revistas, boletines y blogs de seguridad de Internet reproducen los acontecimientos del día anterior en una vorágine de información, comentarios y opiniones [www51][www54] [www55][www56][www69][www70]. Incluso el mismo Dan Kaminsky intenta aclarar la situación concediendo una entrevista a la conocida revista *Wired* [www37][www38] pero ya es demasiado tarde, la gestión controlada de la vulnerabilidad ha fracasado.

El día 23 continúa la resaca de comentarios y especulaciones sobre el impacto real de la vulnerabilidad y los posibles *exploits*[6] existentes. Especialistas en seguridad como Bruce Schneier publican sus puntos de vista para evitar que se mate al mensajero [www71].

El 24 de Julio Dan publica un post en su blog [www41] con el título de "*Details*" dónde realiza una explicación detallada de su vulnerabilidad y proporciona algunos datos con el objetivo de contrarrestar las acusaciones de afán de protagonismo y alarmista. Entre los datos que proporciona se encuentra el de estimaciones personales sobre servidores vulnerables al ataque (85% el 9 de Julio, 50% el día 24) que confirman la necesidad de aplicar los parches y la importancia de disponer de una ventana de tiempo para que los administradores puedan actualizar sus sistemas (si bien en realidad han sido 13 días en vez de los 30 que solicitaba).

El día 26 de Julio Kaminsky vuelve a realizar una entrada en su blog dónde comenta muy escuetamente la situación antes de la publicación de su vulnerabilidad, el peligro una vez divulgada y la situación final una vez actualizado el servicio de DNS con los parches.

Los días siguientes siguen marcados por los comentarios y exposiciones que expertos en seguridad, supuestos en muchos casos, realizan en los medios de comunicación justificando o atacando a Dan. Podemos destacar a nivel de resumen esta reflexión de Schneier [www72].

El 6 de Agosto de 2008 sobre las 11.15 horas, Dan Kaminsky presenta en la conferencia Black Hat de Las Vegas [www75] su trabajo sobre la vulnerabilidad del sistema de nombres (DNS) [www74][www76].

El viernes 8 de Agosto Dan publica en su blog un post [www45] a modo de resumen que recoge los aspectos más importante sobre la charla realizada en la conferencia de Las Vegas y destaca lo insegura que es Internet en general ya que no existe un único punto crítico de fallo, sino que muchos aspectos son potenciales talones de Aquiles y por tanto la gestión de la seguridad ha ser tratado como algo global y no únicamente como un problema aislado de un protocolo.

El día 9 de Agosto Dan publica en su blog una interesante entrada de título "*On the flip side"* [www46] donde justifica la elección realizada para solventar esta vulnerabilidad ante las críticas recibidas por parte de la comunidad de "especialistas" en seguridad.

En realidad, como nunca se ocultó, la actualización proporcionada no elimina la posibilidad de este ataque si no que la minimiza incrementando el espacio de posibilidades de 65356 a varios cientos de millones.

[6] Programa o conjunto de técnicas que permite explotar una vulnerabilidad dada en un sistema o servicio para obtener un beneficio ilícito.

En muchas de estas discusiones se plantean otras alternativas a la escogida, como por ejemplo DNSSEC [www78][www79]. Incluso se discuten nuevos modelos de gestión de seguridad en el desarrollo de aplicaciones y los paralelismos de gestión existentes frente a una red hostil y una red segura. Daniel J. Bernstein [www77] es un ejemplo a seguir en este aspecto ya que ha demostrado con sus programas que si nos tomamos en serio la seguridad y forma parte básica del diseño en las aplicaciones y protocolos, estos tipos de ataques oportunistas quedan enormemente mitigados (aunque no eliminados)**.**

El 27 de Agosto Kaminsky reflexiona en su blog sobre la seguridad en Internet en una reseña titulada "*The emergence of a theme*" [www47] dónde señala como toda la red es vulnerable y no únicamente por el protocolo DNS. Protocolos clave como BGP [www80], SNMPv3 [www81] o SSL [www82] entre otros, que también han sido objeto en los últimos meses de grandes vulnerabilidades cuyo impacto es equiparable a la del servidor de nombres. El problema cada vez pasa a ser menos técnico y más político, ¿quien le pone el cascabel al gato en Internet?

Este mismo día Gabriel Somlo [www83] envía un correo electrónico a la lista de usuarios del programa BIND proponiendo una solución para este problema únicamente cambiando un carácter[7] (*One-character patch*) [www57]. La propuesta que realiza Somlo consiste, aproximadamente, en mantener las respuestas en la caché del servidor hasta que caduquen, es decir, que se cumpla en su totalidad su plazo de validez o TTL. El cruce de correos sobre esta nueva propuesta se sucede internamente en la lista durante un par de días.

Finalmente el 29 de Agosto una de las webs de noticias más consultadas en Internet, *Slashdot*, publica esta nueva aportación de Somlo [www84]. A partir de este momento y como suele pasar con este asunto, cientos de servicios de noticias de todo el mundo se hacen eco de esta posibilidad extendiendo el rumor por Internet [www85].

Este mismo día y obligado una vez más por las circunstancias, Dan Kaminsky que considera muy peligrosa esta propuesta, publica en su blog un análisis detallado de esta y otras aportaciones en dos artículos titulados "*Please do not destroy DNS in order to save it*" [www48] y "*Towards the next DNS fix*" [www49].

El argumento que esgrime Kaminsky contra esta posible solución es que puede pervertir una de las características importantes del DNS, su flexibilidad. Si realizamos un cambio de dirección IP en un nombre ya resuelto, por ejemplo porque se ha estropeado el servidor web, este cambio no será visible para los servidores que implementen esta "funcionalidad" mientras el anterior siga siendo válido (no se haya cumplido su TTL).

Los meses de Septiembre y Octubre de 2008 no aportan mucha más información sobre este tema por lo que no han sido detallados. En caso de que el lector desee una perspectiva más amplia puede consultar la bibliografía adjunta.

## 2.2 Controlar lo incontrolable

A primera vista podemos observar un ejemplo modélico de coordinación en la gestión de vulnerabilidades por parte de los especialistas de seguridad que la descubren, las empresas y

[7] Concretamente su propuesta es la de cambiar una comparación estricta del tipo A < B por una más laxa del tipo A <= B.

los desarrolladores. Se plantea incluso la posibilidad de establecer este mecanismo como modelo de comunicación de vulnerabilidades por parte de los CERT [www35][www36]. Sin embargo al realizar una exploración minuciosa podemos observar que tal vez no sea todo tan idílico como aparenta.

Para simplificar el análisis que realizaremos en este apartado, dividiremos en tres partes la gestión detallada en el punto anterior:

1. **Descubrimiento del problema y desarrollo de la solución**: Este apartado abarca desde finales de 2007, cuando Kaminsky es consciente de la gravedad de la vulnerabilidad, hasta el 7 de Julio de 2008.

2. **Anuncio público**: En esta parte incluiremos las 48 horas posteriores al anuncio mundial de la vulnerabilidad realizado el 8 de Julio.

3. **Libre albedrío**: Período de tiempo que va desde el anuncio público hasta la fecha de este escrito (Noviembre de 2008).

### 2.2.1 Descubrimiento del problema y desarrollo de la solución

Cuando Dan es consciente del problema y decide cual es su siguiente paso, fija para siempre el rumbo de los dos primeros apartados. En vez de publicar un *exploit* en cualquiera de los miles de webs de seguridad o en su mismo blog, intentar vender o aprovecharse de su conocimiento, decide buscar un peso pesado de la comunidad como Paul Vixie.

Esta decisión que es la correcta, al menos moralmente, no deja de ser un salto al vacío ya que debido al calado del problema había muchas posibilidades de que una vez reportado, quedase apartado del proceso por las diferentes empresas y organismos que tomasen el control.

Afortunadamente, según mi opinión no únicamente para Dan si no para toda Internet, Kaminsky pasó a ser colaborador de pleno derecho organizando las diferentes sesiones de trabajo y contribuyendo en la evaluación e implementación de las soluciones.

La importancia de que Kaminsky no quedara relegado del proceso es clave, ya que su conocimiento no sólo de la vulnerabilidad en sí si no de cómo llegó a descubrirla es básica para proporcionar una solución razonable. A fin de cuentas hay que ser muy ingenuo para no pensar que antes o después podría aparecer otro Dan Kaminsky.

Contar con su visión permite de primera mano evaluar las soluciones desde el punto de vista adecuado, la del usuario "real" y experimentado, evitando la tentación de una única visión académica o teórica. No hay que olvidar que mayoritariamente los *hackers* viven de estas visiones únicas.

Una vez seleccionado discretamente un grupo de trabajo (no he conseguido detalles sobre el criterio seguido) que generosamente acoge Microsoft, se establece un plan que consiste en buscar una solución, comunicarla a los distintos fabricantes para que la implementen de forma discreta con el objetivo de proporcionar simultáneamente las actualizaciones a toda la comunidad de Internet, y hacer un aviso público conjunto de esta vulnerabilidad sin incluir detalles.

La discreción con la que se ha llevado esta parte vuelve a ser acertada. No ha existido ni un solo indicio de filtración del proceso realizado en estas fechas. Es más, cuando se encuentran con empresas como Yahoo que utilizan la versión 8 del programa BIND este grupo tiene el suficiente peso como para hacerles cambiar todos sus sistemas a la versión 9.

Cabe destacar cómo una y otra vez Kaminsky destaca muy favorablemente el apoyo y soporte incondicional de grandes empresas y multinacionales en la gestión de esta crisis. Algo que a priori (cada empresa tiende a cuidar lo suyo, por eso son empresas y no ONGs) podría haber resultado muy difícil de coordinar.

Finalmente, nada es gratis, Microsoft impone que el día de la semana en que se publicaran las versiones corregidas de los programas, un martes (curiosamente el mismo día en el que esta compañía publica sus actualizaciones).

### 2.2.2 Anuncio público

De forma coordinada y tras tener accesibles las actualizaciones de los distintos programas, el martes 8 de Julio se produce el anuncio oficial de la existencia de una vulnerabilidad crítica en el sistema de DNS. Se envían las notas a todos los sistemas de alerta (CERT) de Internet y todos los boletines y webs de seguridad hacen referencia a este tema indicando la necesidad de actualizar los sistemas inmediatamente [www100][www101].

Dan Kaminsky publica en su blog el día 9 de Julio una entrada en la que describe cómo los distintos organismos y empresas de Internet han colaborado estrechamente ante este riesgo y remarca la importancia de actualizar los sistemas. Señala que no va a proporcionar detalles concretos de la vulnerabilidad debido a su gravedad y solicita a la comunidad de expertos que durante 30 días se abstengan de especular al respecto con el objetivo de dar tiempo a todos los administradores para actualizar sus sistemas[8].

Además emplaza a todos lo interesados en la vulnerabilidad a la conferencia que realizará el 6 de Agosto dentro del Black Hat en Las Vegas.

Si bien el modelo escogido en este caso es el adecuado al centralizar toda la gestión y emitir un único aviso en vez de cientos de avisos por parte de cada fabricante, no deja de ser muy ingenuo realizar una petición a toda Internet. Como en cualquier grupo numeroso, no olvidemos que Internet tiene cientos de millones de usuarios, siempre tendremos gente dispuesta a sacar ventaja a cualquier precio. Como se suele decir popularmente "el mal nunca descansa".

Otra cuestión a tener en cuenta es el tema de la divulgación (*disclosure*) de vulnerabilidades en sistemas o servicios [www86][www87]. Pese a que nosotros como usuarios percibamos Internet como un "todo" no hay que perder de vista que en última instancia cada servidor, y por tanto cada servicio, reside físicamente en un país u otro y como tal está obligado por su legislación vigente.

[8] No hay que olvidar que existen decenas de millones de servidores de nombres en Internet.

De forma muy resumida podemos clasificar en dos grupos las distintas posiciones existentes. Divulgación total (*full disclosure*), dónde estas vulnerabilidades son publicadas[9] en su totalidad incluyendo los detalles técnicos para favorecer el conocimiento y el desarrollo de programas seguros, y contra la divulgación total (*non full disclosure*), dónde se prohíbe la difusión pública de detalles en favor del orden y control del riesgo asociado[10].

En una serie de artículos aparecidos este año en SecurityFocus [www58][www88] se repasa la legislación actual en doce países europeos, España no aparece, formulando a un abogado de cada uno de estos países la misma pregunta sobre la legislación vigente referente a la divulgación de vulnerabilidades (*What does the current law of your country say about disclosure of security vulnerabilities in software?*).

De las respuestas recibidas se puede deducir asombrosamente que prácticamente no hay legislación específica sobre el tema, derivando la cuestión a leyes sobre intrusión en sistemas o posesión de herramientas o información que permite realizar un acto ilegal.

En el caso de Europa existe como normativa de referencia el *Council of Europe Convention on Cybercrime* [www93][www94][www95] que intenta regular la persecución de los actos ilegales en Internet dotando de un marco común a los distintos países. Desgraciadamente todo y recoger aspectos como el fraude, acceso no autorizado a datos y ordenadores o los actos que infrinjan el Copyright, no dejan de ser una serie de normas o recomendaciones con poca cobertura real que únicamente suelen utilizarse en casos concretos que afectan a varios países. Esta normativa también ha sido ratificada entre otros muchos países por Estados Unidos.

Mayoritariamente la ley no sanciona la publicación de vulnerabilidades en sistemas o servicios siempre y cuando el descubridor no obtenga beneficio de ello. Es indicativo un caso en Dinamarca dónde un adolescente publicó la dirección de una página web que dejaba sin servicio a un sistema de pago por Internet denominado Valus [www89][www90]. Si bien este joven fue absuelto, puesto que no se demostró intención de dañar o causar perjuicio directo al sistema, la gente que la utilizó sí fue condenada con una multa.

También quedan recogidas más o menos universalmente en las distintas legislaciones el concepto de perjuicio ocasionado, no el posible si no el real, por la divulgación de una vulnerabilidad. Cabe destacar como posible ejemplo los casos de divulgaciones de problemas de seguridad falsos o exagerados (bulos) con el objetivo de perjudicar a la competencia. En este caso es necesario denuncia expresa del perjudicado.

Finalmente desde el punto de vista legal mayoritario, el tema es peliagudo y debe ser tratado con cautela ya que como no hay normas claras/jurisprudencia cada tribunal tiende a aplicar su criterio, también sería factible perseguir la divulgación de vulnerabilidades por la vía de posesión de información privilegiada o el uso de herramientas de *hacking*/ingeniería inversa.

En muchos países la posesión de herramientas que permitan cometer un delito (por ejemplo una lanza térmica [www91][www92]) si se demuestra intencionalidad es punible.

---

[9] Es importante destacar que esto **no** significa publicitar irresponsablemente o sin control. Existen organismos como los CERT encargados de recoger esta información y divulgarla adecuadamente.

[10] Las empresas y organismos afectados controlan en este caso el flujo de información. Si bien puede parecer que este modelo evitaría potenciales problemas del sistema anterior, cabe el peligro de la ocultación o perversión en la gestión de la información.

Como ejemplo mediático de esta parte queremos destacar el caso de Guillaume T. [www93] un joven investigador francés que publicó en su blog varias vulnerabilidades [www96] de un sistema antivirus denominado Viguard perteneciente a la compañía Tegam Internacional que en 2004 estaba realizando en Francia una campaña publicitaria muy agresiva asegurando que su producto era 100% seguro contra todo tipo de virus conocidos y desconocidos.

Guillaume (conocido popularmente como Guillermito) fue denunciado por esta compañía solicitando 900.000 Euros en compensación por los cargos de falsedad, divulgación de información maliciosa y uso de herramientas/procedimientos ilegales (denominación del uso de técnicas de ingeniería inversa). Tras varios meses de juicio Guillermito fue condenado y su posterior apelación desestimada (puede consultarse la cronología y la sentencia original en [www97]).

Al margen del gran circo mediático que en su momento creó este juicio y su sentencia, la parte a destacar y que generalmente se omite por ambas partes es la que motivó realmente la condena: ¡El uso de una copia del antivirus para la que Guillaume no tenía licencia! [www98].

De esta forma lo que en realidad empezó a ser un juicio por la "libertad de expresión y difusión del conocimiento", acabó convertido en un simple uso ilegítimo de un programa para el cual no había pagado la licencia.

### 2.2.3 Libre albedrío

Una vez realizado el anuncio público de la vulnerabilidad del sistema de DNS y tras 13 días de comentarios y especulaciones discretas, Thomas Dullien (Halvar Flake) el 21 de Julio abre la caja de Pandora apuntando más o menos acertadamente en qué consiste la vulnerabilidad que Kaminsky intenta esconder hasta su conferencia en Las Vegas.

Es a partir de este momento dónde los comentarios empiezan a subir de tono cuestionando el impacto real de este problema, el tipo de gestión que se ha aplicado y el papel protagonista de Dan Kaminsky.

Mucha gente desea llevarse la gloria del descubrimiento y ser el primero en publicar un documento o *exploit* en Internet, a fin de cuentas como dijo Senna el segundo es el primero de los perdedores. Tampoco hay que obviar el negocio que supondría vender esta información a grupos de extorsionadores y piratas informáticos.

Ese mismo día (¿por afán de protagonismo?) un empleado de la empresa Matasano que estaba informado de la vulnerabilidad, publica los detalles concretos del ataque. En pocas horas se borra esta información del blog, pero ya es tarde, cientos y en pocas horas miles de sitios web se han hecho eco de esta explicación.

Justo a partir de este momento todo el trabajo cuidadosamente planificado y desarrollado durante seis meses cae como un castillo de naipes. Empieza una frenética competición de artículos y comentarios de "expertos en seguridad" destinados a cuestionar la importancia de esta vulnerabilidad e incluso la necesidad de actualizar los servidores de DNS [www54] [www55][www56][www99].

El mismo día 23 aparecen los primeros *exploits* en Internet para que cualquiera pueda utilizarlos a su conveniencia [www66][www67]. Abrumado por los acontecimientos Kaminsky intenta reconducir la situación publicando diferentes artículos en su blog insistiendo en la importancia de que se actualicen los sistemas y concediendo múltiples entrevistas.

Sin embargo y pese a los esfuerzos desplegados tanto por Dan como por el resto de instituciones y empresas que participaron en la gestión de esta vulnerabilidad, cada vez más voces discrepantes se alzan en Internet cuestionando la importancia que se le ha dado e incluso acusándoles de exagerar una vulnerabilidad conocida desde hace años [www73].

En muchos foros y webs de noticias de seguridad aparecen opiniones contradictorias que incluso recomiendan no actualizar los sistemas hasta no conocer todos los detalles. Desde el punto de vista de costes económicos actualizar todos los DNS (imaginemos una gran empresa o un ISP) son "únicamente" muchas horas extras en el mejor de los casos. Como ejemplo de esta situación caótica podemos señalar que a finales de Julio ¡Apple aún no había distribuido la versión corregida de su servidor DNS! [www103].

Durante el mes de Agosto empiezan a aparecer varias acusaciones de censura y exceso de control [www104][www105][www106] hacia las personas externas a los grupos de trabajo. Diferentes propuestas como la de utilizar conexiones TCP en vez de UDP [www107] o la de no actualizar los registros previamente resueltos mientras sean válidos (ver apartados 1 y 2 de este documento referentes a tiempo de vida o TTL) añaden más fuego al debate oficial al ser rechazadas tajantemente.

### 2.3 Conclusiones

Como hemos podido observar analizando el proceso tan en detalle como ha sido posible, la gestión discreta en la primera fase es clave para poder conseguir unos buenos resultados. La elección crítica se produce siempre en el primer paso ya que cuando se filtra un rumor y se pierde el control de una situación, es imposible reconducirla.

Por este motivo es muy importante potenciar la existencia de organismos como los CERT que permitan iniciar una gestión adecuada de las situaciones de riesgo y poner al cargo de estos centros verdaderos profesionales de la seguridad informática capaces de iniciar y coordinar las acciones necesarias en casos de peligro. También se ha de formar y concienciar a todo el sector de las tecnologías de la información (TI) sobre la importancia real de la seguridad. En un mundo (Internet) sin fronteras ni distancias, lo que afecta a mi vecino me acabará afectando a mí.

Los profesionales de las TI deben saber dónde han de acudir en busca de información o asesoramiento en las situaciones de riesgo. Han de tener las nociones de que protocolos de actuación se han de seguir y sobre todo que no se ha de hacer.

La sociedad también debe ser sensibilizada sobre el tema de la seguridad de forma que exija profesiones capaces de dar respuesta a estas necesidades y confíe en las gestiones realizadas.

Finalmente la auditoria pública de la gestión realizada por estos centros cuando el riesgo haya disminuido es clave para conseguir no sólo una red segura, si no una red en la que confiemos.

## 3. Agradecimientos

Para ser honestos debo reconocer que este documento es bastante más largo de lo que en un principio planifiqué. También reconozco que he tenido que dedicar bastante más tiempo del que disponía inicialmente a documentarme, leer y sobre todo entender las infinitas referencias del tema.

De esta forma, si usted ha soportado las veintitrés páginas anteriores y ha llegado hasta este punto, mi agradecimiento sincero es para usted. También debo agradecer a Arnau, David y Rubén/outime sus acertados comentarios.

## 4. Copyright

# Bibliografía


[Stevens94] "TCP/IP Illustrated, Volume I", W. Richard Steven. Addison-Wesley 994.
[AL01] "DNS and Bind", Paul Albitz, Cricket Liu. Oreilly 2001.
[Liu02] "DNS & Bind Cookbook", Cricket Liu. Oreilly 2002.
[SJ08] "Effect of triangular routing in mixed Ipv4/IPv6 Networks", Teerapat Sanguankotchakorn, Preeda Jaiton, IEEE 2008. (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04498189)

[www1] http://www.kb.cert.org/vuls/id/800113
[www2] http://tools.ietf.org/html/rfc920
[www3] http://tools.ietf.org/html/rfc1032
[www4] http://www.unix.com.ua/orelly/networking_2ndEd/dns/ch12_09.htm
[www5] http://www.dnsdoctor.org/Tests_description#Root_servers
[www6] http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html
[www7] http://tools.ietf.org/html/rfc2308
[www8] http://en.wikipedia.org/wiki/Domain_Information_Groper
[www9] http://www.madboa.com/geek/dig/
[www10] http://www.kloth.net/services/dig.php
[www11] http://www.geektools.com/digtool.php
[www12] http://en.wikipedia.org/wiki/DNS_Backbone_DDoS_Attacks
[www13] http://www.cs.cornell.edu/People/egs/beehive/rootattack.html
[www14] http://tools.ietf.org/html/rfc791
[www15] http://gabriel.verdejo.alvarez.googlepages.com/
[www16] http://www.doxpara.com/
[www17] http://www.onguardonline.gov/topics/phishing.aspx
[www18] http://es.wikipedia.org/wiki/Phishing
[www19] http://mydns.bboy.net/survey/
[www20] http://cr.yp.to/surveys/dns1.html
[www21] http://www.debian.org/security/2008/dsa-1571
[www22] http://www.rediris.es/rediris/boletin/57/enfoque2.html
[www23] http://gabriel.verdejo.alvarez.googlepages.com/DEA-es-2DOS-DDOS.pdf
[www24] http://www.cert.org/homeusers/ddos.html
[www25] http://www.securityfocus.com/columnists/477
[www26] http://www.securityfocus.com/news/11526
[www27] http://www.securityfocus.com/news/11529
[www28] http://www.microsoft.com
[www29] http://www.rediris.es/rediris/
[www30] http://www.doxpara.com/?p=1162
[www31] http://www.isc.org/products/BIND/
[www32] http://cr.yp.to/djbdns/install.html
[www33] http://defcon.org/html/defcon-16/dc-16-schedule.html
[www34] http://media.defcon.org/dc-16/video/dc16_kaminsky/dc16_kaminsky_cache_full.mov
[www35] http://www.cert.org/
[www36] http://www.rediris.es/cert/
[www37] http://blog.wired.com/27bstroke6/2008/07/details-of-dns.html
[www38] http://blog.wired.com/27bstroke6/2008/07/kaminsky-on-how.html
[www39] http://isc.sans.org/diary.html?storyid=4687
[www40] http://www.doxpara.com/?p=1162

[www41] http://www.doxpara.com/?p=1185
[www42] http://www.doxpara.com/?p=1189
[www43] http://www.doxpara.com/?p=1202
[www44] http://www.doxpara.com/?p=1204
[www45] http://www.doxpara.com/?p=1213
[www46] http://www.doxpara.com/?p=1215
[www47] http://www.doxpara.com/?p=1231
[www48] http://www.doxpara.com/?p=1234
[www49] http://www.doxpara.com/?p=1237
[www50] http://www.doxpara.com/?p=1250
[www51] http://www.securityfocus.com/brief/779
[www52] http://amd.co.at/dns.htm
[www53] http://addxorrol.blogspot.com/2008/07/on-dans-request-for-no-speculation.html
[www54] http://beezari.livejournal.com/141796.html
[www55] http://www.linuxjournal.com/content/understanding-kaminskys-dns-bug
[www56] http://www.linuxjournal.com/content/dns-bug-why-you-should-care
[www57] http://marc.info/?t=121981071400003
[www58] http://www.securityfocus.com/print/columnists/466
[www59] http://www.eecs.wsu.edu/~smedidi/Mipv6.ppt
[www60] http://www.dns.net/dnsrd/rr.html
[www61] http://www.isc.org/index.pl?/about/mgmt/vixie.php
[www62] http://en.wikipedia.org/wiki/Paul_Vixie
[www63] http://www.isc.org/index.pl
[www64] http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1320461,00.html
[www65] http://news.cnet.com/8301-10789_3-9985618-57.html
[www66] http://www.caughq.org/exploits/CAU-EX-2008-0002.txt
[www67] http://www.caughq.org/exploits/CAU-EX-2008-0003.txt
[www68] http://www.matasano.com/
[www69] http://www.hispasec.com/unaaldia/3546
[www70] http://www.hispasec.com/unaaldia/3559
[www71] http://www.schneier.com/essay-230.html
[www72] http://www.schneier.com/blog/archives/2008/07/the_dns_vulnera.html
[www73] http://isc.sans.org/diary.html?storyid=4693
[www74] http://www.doxpara.com/DMK_BO2K8.ppt
[www75] http://www.blackhat.com/html/bh-usa-08/bh-usa-08-schedule.html
[www76] http://venturebeat.com/2008/08/06/black-hat-dan-kaminsky-explains-the-bug-that-threatened-the-internet/
[www77] http://cr.yp.to/djb.html
[www78] http://www.dnssec.net/
[www79] http://www.dnssec-deployment.org/
[www80] http://www.bgp4.as/
[www81] http://tools.ietf.org/html/rfc3411
[www82] http://tools.ietf.org/html/rfc5246
[www83] http://www.cs.colostate.edu/~somlo/
[www84] http://www.securityfocus.com/brief/808
[www85] http://it.slashdot.org/article.pl?sid=08/08/29/127210
[www86] http://www.csoonline.com/article/440110/The_Vulnerability_Disclosure_Game_Are_We_More_Secure_?CID=28072
[www87] http://www.schneier.com/blog/archives/2007/01/debating_full_d.html
[www88] http://www.securityfocus.com/columnists/448/
[www89] http://www.valus.dk
[www90] http://www.fauxhemian.dk/archives/000189.html

[www91] http://www.multipick-service.cc/htdocs/es/werkzeug/thermolanzen/
[www92] http://camarasacorazadas.blogspot.com/
[www93] http://www.usdoj.gov/criminal/cybercrime/COEFAQs.htm
[www94] http://www.coe.int/t/DG1/LEGALCOOPERATION/ECONOMICCRIME/cybercrime/default_en.asp
[www95] http://conventions.coe.int/Treaty/Commun/QueVoulezVous.asp?NT=185&CL=ENG
[www96] http://www.hispasec.com/unaaldia/2686
[www97] http://www.guillermito2.net/archives/2004_12_28.html
[www98] http://www.schneier.com/blog/archives/2005/03/france_makes_fi.html
[www99] http://blog.invisibledenizen.org/2008/07/kaminskys-dns-issue-accidentally-leaked.html
[www100] http://news.bbc.co.uk/2/hi/technology/7496735.stm
[www101] http://www.kriptopolis.org/fallo-critico-dns-obliga-parchear-internet
[www102] http://www.wired.com/politics/security/commentary/securitymatters/2008/07/securitymatters_0723
[www103] http://db.tidbits.com/article/9706
[www104] http://www.ops.ietf.org/lists/namedroppers/namedroppers.2008/msg01433.html
[www105] http://www.av8.net/IETF-watch/DNSEXT/Management.html
[www106] http://cr.yp.to/djbdns/namedroppers.html
[www107] http://readlist.com/lists/isc.org/bind-users/2/10854.html
[www108] http://www.ietf.org/proceedings/08jul/slides/dnsext-2.pdf
[www109] http://www.caida.org/workshops/wide/0808/slides/source_port_randomness.pdf
[www110] http://tools.ietf.org/html/draft-vixie-dnsext-dns0x20-00