

## DNS Snooping, saber lo que sabes

### Introducción

Por estos días en que el mundo habló de la famosa **vulnerabilidad en DNS** [1] y de **Dan Kaminsky** [2]; los servidores de nombres tomaron protagonismo nuevamente en el escenario de las vulnerabilidades informáticas. Casualmente me encuentro por motivos laborales con otra "vulnerabilidad" en servidores DNS; que curiosamente también llegó en su momento a las noticias, de la mano de Dan Kaminsky. La misma, se llama **DNS Snooping** y en el presente artículo describiré sus aspectos principales: en qué consiste, cuál es el impacto que se puede sufrir, cómo conocer si se está infectado y algunas vías de solución disponibles.

### La vulnerabilidad

**DNS cache snooping** es una técnica que permite conocer los nombres de dominio que ha resuelto un servidor DNS. Permite al atacante averiguar qué dominios están resueltos por el servidor y, consecuentemente, cuáles no.

El hecho de que un servidor DNS ofrezca esta vulnerabilidad, como se puede observar en la introducción, fue colocado entre comillas ya que representa lo que es considerado como el nivel más bajo de vulnerabilidades que puede sufrir un sistema informático. Un servidor DNS vulnerable está otorgando información sobre la red al atacante y dicha información puede ser utilizada para explotar eficientemente otras vulnerabilidades. La información obtenida por el atacante, explotando esta vulnerabilidad, se encuentra enmarcada en la **fase de descubrimiento**, si se consideran las etapas de un ataque informático.

**Traducción**  
*snoop = husmear*

La vulnerabilidad, explota una debilidad en el **DNS cache** (la forma en que nuestro servidor almacena la información y la entrega a través de las consultas que recibe). Se basa en la realización de consultas no recursivas (o iterativas) [3] a la caché del servidor para obtener información asociada al servicio.

No existe una incidencia directa entre la vulnerabilidad y una intrusión o ataque a la información; aunque la información obtenida por el DNS snooping permitirá optimizar futuras intrusiones, explotando otras vulnerabilidades.

En resumen, no es una vulnerabilidad de alto riesgo ya que representa simplemente la posibilidad de un atacante de obtener información de la organización; no de modificarla o alterarla.

### Impactos

El primer impacto consiste en la posibilidad para un atacante de conocer información de nuestra infraestructura. Como se mencionaba anteriormente, aún sin existir un efecto directo entre la exposición de dicha información y un impacto de

negocio; como principio de seguridad es importante resguardar la confidencialidad de nuestra información y que ésta sea accedida solo por las personas correctas.

La información obtenida es interesante para el atacante, ya que permite crear un perfil de los patrones de uso de la comunidad de usuarios de dicho servidor DNS.

Por ejemplo, si un atacante está interesado en conocer si una empresa utiliza ciertos servicios en línea de una determinada institución financiera, es posible utilizar este ataque para conocer el acceso desde la empresa a la mencionada institución financiera. Por supuesto, también es posible utilizar el ataque para encontrar socios B2B, o establecer patrones de navegación, acceso a servidores de correo externos, y mucho más. [4]

Luego del impacto inmediato, existe la posibilidad de que un atacante haga uso de la información obtenida, para explotar otras vulnerabilidades o realizar otros ataques y/o intrusiones.

Por ejemplo, un delincuente con interés de utilizar la técnica de **typosquatting** [5] para perpetrar infecciones o ataques, puede utilizar la técnica de DNS snooping para conocer qué dominios son más consultados por un grupo determinado de usuarios. De manera más explícita: el typosquatting se basa en el aprovechamiento de errores de tipeo de los usuarios. En ese caso, alguien podría montar en el sitio **www.google.com**, una web con publicidades o malware y sacar algún tipo de provecho de los errores de los usuarios. Por lo tanto, a través del uso de DNS snooping, un atacante puede conocer qué dominios son mayormente consultados por los usuarios entre **www.google.com** o **www.google.com**.

Otros usos de la técnica pueden ser localizar usuarios en Internet, hacer seguimiento de e-mails o conocer IDs de sesión; aunque estos casos son menos probables por la complejidad necesaria para obtener la información. Puede obtenerse más información de este tipo de escenarios en el documento "**Snooping the Cache for Fun and Profit**" (ver al pie **Fuentes**).

### **Ethical DNS snooping**

*"El 31 de Octubre (de 2005), Mark Russinovich irrumpió con el tema en su blog: **Sony BMG Music Entertainment** distribuía un esquema de protección contra copia en sus Cds musicales que **instalaba en secreto un rootkit en los ordenadores.**"* [6]

Durante el escándalo que causó el sistema de detección DRM de Sony [7], Dan Kaminsky utilizó la técnica de DNS cache snooping [8] para conocer cuántos servidores DNS habían contactado los servidores de Sony y, por lo tanto, trazar estadísticas aproximadas sobre la tasa de infección del rootkit. El concepto que aplicó Kaminsky es el siguiente: *"Si es posible aplicar la técnica de "Cache Snooping" a todos los servidores DNS que existen en Internet sería posible detectar todos aquellos DNS que existen en el mundo los cuales han sido utilizados por este malware."* [9]

La técnica puede ser extendida para detectar cualquier tipo de malware que contacte servidores, ya que para ello, cualquier equipo deberá resolver primariamente el nombre de dominio de los servidores que el código malicioso utilice. Kaminsky concluyó, en su momento, que al menos 568,200 redes estaban infectadas con el rootkit de Sony [10]. Lo interesante de este experimento es que revela qué enorme cantidad de servidores DNS permiten este tipo de consultas y no han implementado ningún tipo de contramedidas para denegar este tipo de accesos.

### **¿Soy vulnerable?**

Existen diferentes procedimientos para conocer si un servidor DNS es o no vulnerable.

El primero, automatizado, consiste en utilizar la herramienta **DNS Report** [11]. Si esta devuelve una salida similar a la que se presenta a continuación, significa que el servidor DNS permite consultas recursivas por cualquiera [12]:

```
«ERROR: One or more of your nameservers reports that it is an open DNS server. This usually means that anyone in the world can query it for domains it is not authoritative for (it is possible that the DNS server advertises that it does recursive lookups when it does not, but that shouldn't happen). This can cause an excessive load on your DNS server. Alos, it is strongly discouraged to have a DNS server be both authoritative for your domain and be recursive (even if it is not open), due to the potential for cache poisoning (with no recursion, there is no cache, and it is impossible to poison it). Alos, the bad guys could use your DNS server as part of an attack, by forging their IP address»
```

Otro procedimiento, permite detectar manualmente si somos vulnerables, consiste en utilizar la herramienta dig [13] para hacer consultas DNS iterativas y concluir al respecto. Si el servidor DNS regresa una respuesta válida, se puede concluir que el servidor DNS ha resuelto el nombre de dominio. Incluso se puede conocer, con el TTL del registro, cuanto tiempos atrás el servidor DNS "conoció" dicho dominio.

Para verificar si un servidor es vulnerable, ejecutar el comando dig con el siguiente formato:

```
$ dig @[IP_DNS] [sitio_web] A +norecurse
```

Por ejemplo:

```
user@workstation:~> dig @200.xxx.2.66 www.google.com A +norecurse
; <<>> DiG 9.2.4 <<>> @200.118.2.66 www.google.com A +norecurse
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50160
;; flags: qr; QUERY: 1, ANSWER: 5, AUTHORITY: 7, ADDITIONAL: 7
```

```
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                550594  IN      CNAME   www.l.google.com.
www.l.google.com.              135     IN      A       74.125.47.104
www.l.google.com.              135     IN      A       74.125.47.147
www.l.google.com.              135     IN      A       74.125.47.99
www.l.google.com.              135     IN      A       74.125.47.103

;; AUTHORITY SECTION:
l.google.com.                  32194   IN      NS      c.l.google.com.
l.google.com.                  32194   IN      NS      d.l.google.com.
l.google.com.                  32194   IN      NS      e.l.google.com.
l.google.com.                  32194   IN      NS      f.l.google.com.
l.google.com.                  32194   IN      NS      g.l.google.com.
l.google.com.                  32194   IN      NS      a.l.google.com.
l.google.com.                  32194   IN      NS      b.l.google.com.

;; ADDITIONAL SECTION:
a.l.google.com.                38252   IN      A       209.85.139.9
b.l.google.com.                34682   IN      A       64.233.179.9
c.l.google.com.                34682   IN      A       64.233.161.9
d.l.google.com.                34682   IN      A       66.249.93.9
e.l.google.com.                34682   IN      A       209.85.137.9
f.l.google.com.                41290   IN      A       72.14.235.9
g.l.google.com.                36622   IN      A       64.233.167.9

;; Query time: 264 msec
;; SERVER: 200.118.2.66#53(200.118.2.66)
;; WHEN: Mon Aug 11 17:04:27 2008
;; MSG SIZE rcvd: 340
```

Se puede observar que el campo "ANSWER:" es igual a 5 y se devuelve una respuesta a la consulta. Es decir, que el servidor **conoce el dominio** `www.google.com` en su cache.

Si hacemos sobre el mismo servidor, una consulta de un dominio desconocido, obtenemos el siguiente resultado:

```
user@workstation:# dig @200.xxx.2.66 www.dominioquenoexiste.com
A +norecurse

; <<>> DiG 9.2.4 <<>> @200.118.2.66 www.dominioquenoexiste.com A
+norecurse
;; global options: printcmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 61151
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 13

;; QUESTION SECTION:
```

```
;www.dominioquenoexiste.com.      IN      A

;; AUTHORITY SECTION:
com.      121111  IN      NS      a.gtld-servers.net.
com.      121111  IN      NS      b.gtld-servers.net.
com.      121111  IN      NS      c.gtld-servers.net.
com.      121111  IN      NS      d.gtld-servers.net.
com.      121111  IN      NS      e.gtld-servers.net.
com.      121111  IN      NS      f.gtld-servers.net.
com.      121111  IN      NS      g.gtld-servers.net.
com.      121111  IN      NS      h.gtld-servers.net.
com.      121111  IN      NS      i.gtld-servers.net.
com.      121111  IN      NS      j.gtld-servers.net.
com.      121111  IN      NS      k.gtld-servers.net.
com.      121111  IN      NS      l.gtld-servers.net.
com.      121111  IN      NS      m.gtld-servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.  122689  IN      A        192.5.6.30
a.gtld-servers.net.  122689  IN      AAAA     2001:503:a83e::2:30
b.gtld-servers.net.  124878  IN      A        192.33.14.30
b.gtld-servers.net.  129738  IN      AAAA     2001:503:231d::2:30
c.gtld-servers.net.  124878  IN      A        192.26.92.30
d.gtld-servers.net.  122006  IN      A        192.31.80.30
e.gtld-servers.net.  121398  IN      A        192.12.94.30
f.gtld-servers.net.  121398  IN      A        192.35.51.30
g.gtld-servers.net.  122371  IN      A        192.42.93.30
h.gtld-servers.net.  121398  IN      A        192.54.112.30
i.gtld-servers.net.  121332  IN      A        192.43.172.30
j.gtld-servers.net.  122006  IN      A        192.48.79.30
k.gtld-servers.net.  122884  IN      A        192.52.178.30

;; Query time: 232 msec
;; SERVER: 200.118.2.66#53(200.118.2.66)
;; WHEN: Fri Aug 8 16:23:43 2008
;; MSG SIZE rcvd: 500
```

En este caso, el dominio **no es conocido** por el servidor DNS que está siendo consultado. Es decir, el servidor responde en la salida el campo **ANSWER**, "0" cuando no conoce el dominio y un valor "1" o superior, cuando conoce el dominio.

Un servidor que no posea esta vulnerabilidad **debe responder siempre "0"** cuando se genera este tipo de consultas no recursivas. En realidad, se están denegando las consultas no recursivas.

Puede comprobarse un servidor seguro, haciendo las consultas dig antes descriptas, a los servidores de **OpenDNS** [14], cuyas direcciones IP son **208.67.220.220** y **208.67.222.222**.

Por ejemplo:

```
user@workstation:# dig @208.67.222.222
www.dominioquenoexiste.com A +norecurse

; <<>> DiG 9.2.4 <<>> @208.67.222.222 www.dominioquenoexiste.com A
+norecurse
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 9598
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.dominioquenoexiste.com.      IN      A

;; Query time: 173 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Fri Aug  8 16:28:05 2008
;; MSG SIZE  rcvd: 44

user@workstation:# dig @208.67.222.222 www.google.com A
+norecurse

; <<>> DiG 9.2.4 <<>> @208.67.222.222 www.google.com A +norecurse
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 30686
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                  IN      A

;; Query time: 170 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Fri Aug  8 16:28:51 2008
;; MSG SIZE  rcvd: 32
```

Puede obtenerse información más detallada de los chequeos, en el documento "**Snooping the Cache for Fun and Profit**" (ver al pie **Fuentes**).

### **Posibles soluciones**

Existen varias alternativas para mitigar el riesgo de exposición de nuestros servidores y denegar la posibilidad a un atacante de "conocer lo que conocemos".

En la realidad, solo debería permitirse hacer consultas recursivas a los usuarios de la red local. Por lo tanto, la primer recomendación es no exponer el servidor DNS, salvo que sea necesario. En caso que la organización necesite exponer el servicio de DNS, hacerlo con un servidor dedicado para tal fin, diferente al que resuelva las peticiones de la red interna de la organización.

Otras alternativas, que implican un mayor grado de riesgo y son más complejas, incluyen configurar los DNS para que no respondan queries no recursivos. Existen procedimientos para hacerlo en Windows [15] y en Linux y se puede consultar el funcionamiento de cada servidor DNS y configurarlo para tal fin.

La última alternativa es utilizar listas de control y definir cuáles direcciones IP pueden realizar consultas de este tipo y cuáles no [12].

### **Conclusión**

Evidentemente esta vulnerabilidad no es un riesgo crítico ni responderá un "alerta roja" en cualquier potencial análisis de riesgos que se realice.

Sin embargo, la confidencialidad es uno de los tres pilares de la seguridad de la información en cualquier organización. A pesar que el impacto directo no pueda ser importante, no es posible conocer para qué es utilizada la información extraída de nuestros servidores DNS y, además, las medidas de control para mitigar esta vulnerabilidad, son simples y de rápida implementación.

Incluso un servicio como DNS, que para muchos administradores implica un servicio simple de configurar y de baja administración, requiere la atención necesaria en lo que respecta a seguridad.

## **Referencias**

- [1] <http://seguinfo.blogspot.com/2008/07/resumen-del-tema-dns-al-da-de-hoy-por.html>
- [2] <http://seguinfo.blogspot.com/2008/08/habl-kaminsky-en-black-hat.html>
- [3] <http://es.wikipedia.org/wiki/DNS>
- [4] <http://www.securityspace.com/smysecure/catid.html?id=12217>
- [5] <http://virusattack.blogspot.com/2008/04/qu-es-el-typosquatting-leccin-15.html>
- [6] <http://www.kriptopolis.org/el-rootkit-del-drm-de-sony-la-verdadera-historia>
- [7] [http://en.wikipedia.org/wiki/Dan\\_Kaminsky](http://en.wikipedia.org/wiki/Dan_Kaminsky)
- [8] <http://www.enterprisenetworkingplanet.com/netsecur/article.php/3564731>
- [9] <http://www.seguridad0.com/index.php?ea75a6a5ac2b0e231b9d4e4ead9a1f73&tim=20-1-2006&ID=2359>
- [10] <http://www.wired.com/politics/security/news/2005/11/69573>
- [11] <http://www.dnsreport.com/>
- [12] <http://www.alcancelibre.org/article.php/como-prevenir-contaminacion-cache-dns>
- [13] [http://en.wikipedia.org/wiki/Domain\\_Information\\_Groper](http://en.wikipedia.org/wiki/Domain_Information_Groper)
- [14] <http://www.opendns.com/>
- [15] <http://technet2.microsoft.com/windowsserver/es/library/e1fe9dff-e87b-44ae-ac82-8e76d19d9c373082.mspx?mfr=true>

## **Fuentes**

Más allá de las referencias presentadas anteriormente, el presente artículo está basado principalmente, en el documento "**Snooping the Cache for Fun and Profit**", en donde es posible extender (en inglés) la información aquí presentada.

El documento puede descargarse en el siguiente enlace: [www.rootsecure.net/content/downloads/pdf/dns\\_cache\\_snooping.pdf](http://www.rootsecure.net/content/downloads/pdf/dns_cache_snooping.pdf)

## **Créditos**

Varios créditos de la información aquí presentada son para mi compañero de trabajo, **Ulises Cuñé**, y a la empresa **Openware**, por permitirnos investigar para brindar un mejor servicio a nuestros clientes y a la comunidad.

## **Licencia**

El presente artículo es liberado bajo [licencia de Creative Commons](http://creativecommons.org/licenses/by/3.0/).

Usted es libre de:

- copiar, distribuir, exhibir, y ejecutar la obra.
- hacer obras derivadas de ella.

Bajo las siguientes condiciones:

- Usted debe atribuir la obra en la forma especificada por el autor o el licenciante.

- Usted no puede usar esta obra con fines comerciales.
- Si usted altera, transforma, o crea sobre esta obra, sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.

<http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>

### **Descarga**

El presente documento puede ser descargado de los siguientes sitios web:

- <http://www.openware.biz>
- <http://blog.openware.biz>