Guía de Seguridad en Servicios

DNS

- Fundamentos de DNS
- Seguridad en DNS
- Vulnerabilidades y amenazas en DNS
- Bastionado DNS
- DNSSEC

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

INTECO
Instituto Nacional de Tecnologías de la Comunicación

Autor

Antonio López Padilla

Coordinación

Daniel Fírvida Pereira

INDICE

1 OBJETIVO DE LA GUÍA

El objeto de esta guía es ofrecer una visión general del servicio DNS, describir los principales ataques de los que es objeto este protocolo o que hacen uso del mismo y proporcionar una orientación sobre las buenas prácticas a aplicar para asegurar su funcionamiento.

Dirigida a operadores y administradores sistemas y redes, pretende servir de ayuda en la implementación o bastionado del servicio.

Aunque el enfoque del documento es el protocolo DNS en general, los ejemplos e implementaciones aquí propuestas se particularizan para el software de código abierto BIND al ser el más extendido en este tipo de servicios [1].

Para ello, este documento se compone de cinco secciones principales:

I. Fundamentos de DNS: conceptos, objetivos y funcionamiento de un sistema DNS.

II. Seguridad en DNS En este apartado se identifican en un escenario típico DNS los posibles vectores de ataque y los activos afectados.

III. Vulnerabilidades y amenazas en DNS: se explican debilidades intrínsecas al diseño del protocolo DNS y los principales ataques que sacan partido de la las mismas.

IV. Bastionado DNS: en esta sección se analizan las medidas de seguridad a implementar en los tres grandes superficies de ataque del servicio DNS: Infraestructura del servicio DNS, Comunicaciones y transacciones y Datos.

V. DNSSEC: Finalmente, se ofrece una introducción a DNSSEC, la evolución en la seguridad de DNS donde, con la introducción de criptografía se pretende dotar al servicio DNS de un mecanismo eficaz para eliminar vulnerabilidades históricas del diseño.

[1] BIND actualmente está distribuido y soportado por Internet System Consortium (ISC: https://www.isc.org/)

2 FUNDAMENTOS DE DNS

En esta sección, se describen los elementos integrantes de una infraestructura DNS, su nomenclatura, organización jerárquica y el protocolo en sí. Se detalla el formato de los mensajes, operaciones y transacciones fundamentales con objeto de tener una visión clara de los conceptos necesarios para la comprensión de las vulnerabilidades que afectan al protocolo.

¿QUÉ ES DNS?

Domain Name System (DNS) es un sistema globalmente distribuido, escalable y jerárquico. Ofrece una base de datos dinámica asociando direcciones IP de computadoras, servicios o cualquier recurso conectado a internet o red privada con información de diverso tipo. Soporta tanto IPv4 como IPv6, y la información se almacena en forma de registros Resource Records (RR) de distintos tipos los cuales pueden almacenar direcciones IP u otro tipo de información. Esta información se agrupa en zonas, que corresponden a un espacio de nombres o dominio y que son mantenidas por el servidor DNS autoritativo de la misma.

Fundamentalmente, DNS se encarga de traducir direcciones IP de recursos de red a nombres fácilmente legibles y memorizables por las personas, y viceversa. A esta acción se la conoce como "resolución DNS". De esta forma, se establece un mecanismo amigable para la localización e identificación de recursos. Comúnmente se usa la analogía de una guía de teléfonos donde se puede localizar a partir de un nombre su número asociado, o a la inversa. En este símil, los números representarían direcciones IP y los nombres, registros del espacio de dominios.

ELEMENTOS INTEGRANTES DE DNS

DNS se estructura en tres componentes principales:

- **Espacio de dominios de nombres:** Consiste en un estructura jerárquica de árbol donde cada nodo contiene cero o más registros (*Resource Records*, o RR) con información del dominio. Del nodo raíz, situado en el nivel más alto, parten las ramas que conforman las mencionadas zonas. Estas, a su vez, pueden contener uno o más nodos o dominios que a su vez pueden dividirse en subdominios según se baja en la jerarquía. Véase *Ilustración 1. Jerarquía del espacio de nombres***Jerarquía del espacio de nombres**
- **Servidores de Nombres:** Son servidores encargados de mantener y proporcionar información del espacio de nombres o dominios. Por una parte, existen servidores que almacenan información completa para uno o varios conjuntos del espacio de nombres (dominios) y de las cuales es responsable. Se dice que son servidores autoritativos de esas zonas/dominios en cuestión. Por otro lado, hay otro tipo de servidor que almacena conjuntos de registros de distintas zonas/dominios que obtiene consultando a los correspondientes servidores autoritativos de las mismas (búsquedas recursivas). Esta información la almacenan localmente de forma temporal (caché) y la renuevan periódicamente. Son los llamados servidores caché. Con los servidores de nombres y su intercomunicación se consigue la distribución y redundancia del espacio de dominios. Con esta organización de servidores de nombres, y su intercomunicación, se consigue la distribución y redundancia del espacio de dominios.

- **Resolvers:** Son servidores caché o programas cliente los cuales se encargan de generar las consultas necesarias y obtener la información solicitada para ofrecerla al usuario que la solicita.

ESPACIO DE DOMINIOS DE NOMBRES. JERARQUÍA Y SINTAXIS

- **Estructura jerárquica**

DNS está compuesto por un espacio de nombres de dominio organizados en jerarquía de árbol donde se enlazan nodos, cada uno representando un nivel del espacio de dominios. El nivel más alto de toda la jerarquía es el dominio raíz o *root*, representado por "." (punto). Justo un nivel por debajo se encuentran los *Top Level Domains* o TLDs. Éstos, a su vez, son nodos padre de otros niveles inferiores que se conocen como TLDs de segundo nivel. Sucesivamente, la jerarquía continúa hasta llegar a un nodo final que representa un recurso. El nombre formado por toda la cadena se conoce como *Fully Qualified Domain Name* (FQDN).

Una zona es una porción del espacio de dominio de nombres cuya administración es delegada a un servidor DNS que ejerce como "autoridad" de esa porción o dominio. A este servidor se le conoce como **servidor autoritativo de la zona**.

La jerarquía comienza en la zona raíz "." siendo el nivel más alto. Aunque normalmente no es mostrado, todo dominio completo termina en un punto final "." que indica el final del espacio en la zona raíz. Por ejemplo *"www.ejemplo.com"* realmente es *"www.ejemplo.com."*, donde el punto final más a la derecha representa la zona raíz. Este dominio completo es lo que se denomina *Fully Qualified Domain Name* (FQDN).

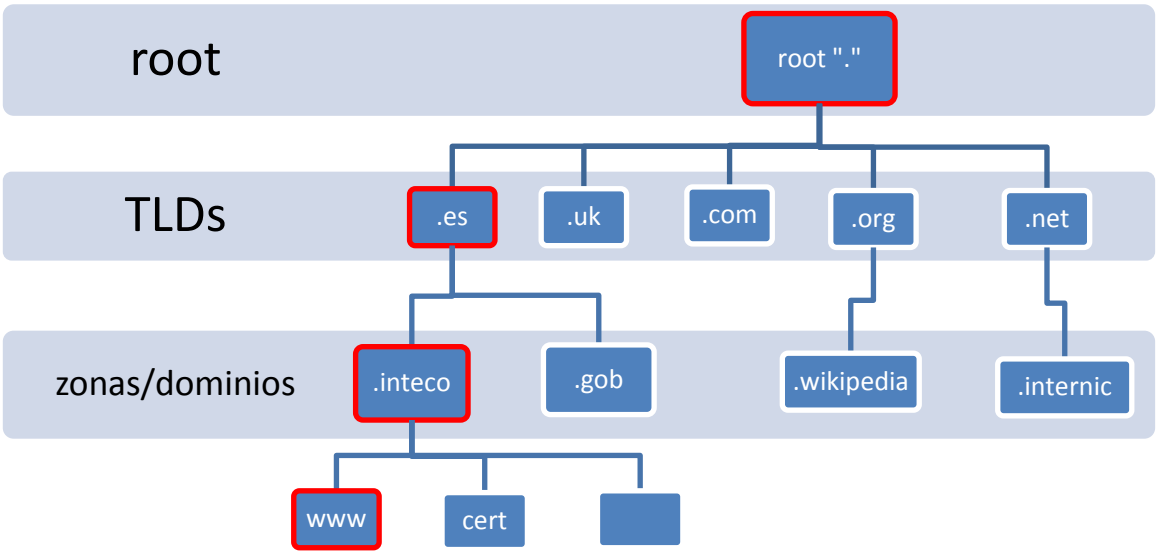

Ilustración 1. Jerarquía del espacio de nombres

- **Nomenclatura DNS**

Dependiendo de su posición en la jerarquía, cada nombre de dominio del espacio de nombres está compuesto por una o más etiquetas separadas por un punto "." cada una con un máximo de 63

caracteres. Un nombre final FQDN puede contener hasta un máximo de 255 caracteres, incluyendo los “.”.

Las etiquetas se construyen de derecha a izquierda, donde la etiqueta más a la derecha representa el *Top Level Domain* (TLD) del dominio. Por ejemplo, ***.es*** es el TLD de la zona ***inteco.es***. Las etiquetas se separan por un punto “.”.

A continuación del TLD, cada etiqueta a la izquierda representa una subdivisión o subdominio. Como se ha indicado, cada etiqueta puede ser de hasta 63 caracteres y puede a su vez subdividirse en otros subdominios, siempre que el FQDN final no exceda el máximo de 255 caracteres. Esta normativa proporciona cierta flexibilidad a la hora de diseñar la jerarquía de subdominios dependientes de un dominio particular. Por último, la parte más a la izquierda del dominio FQDN suele expresar un nombre de máquina o recurso final, genéricamente conocido como host.

En los nombres de dominios no se diferencia entre mayúsculas y minúsculas: por ejemplo, los nombres de dominio *www.misitio.com* y *www.MiSitio.com* serán considerados idénticos

- **Espacio de dominio de direcciones IN-ADDR.ARPA**

En DNS se utiliza el dominio *in-addr.arpa* para definir el espacio de direcciones IP. Mediante este domino se garantiza la resolución inversa de una dirección IP a su correspondiente nombre y así facilitar su búsqueda en Internet.

Los subdominios en *in-addr.arpa* tienen una estructura de hasta 4 etiquetas (IP versión 4), cada una de las cuales representaría un octeto de una dirección IP. Así por ejemplo la información de la dirección IP *213.4.108.69* se localizaría en el dominio *69.108.4.213.in-addr.arpa*. Obsérvese como se sigue el criterio jerárquico en la *Ilustración 2. Dominio 69.108.4.213.in-addr.arpa.*

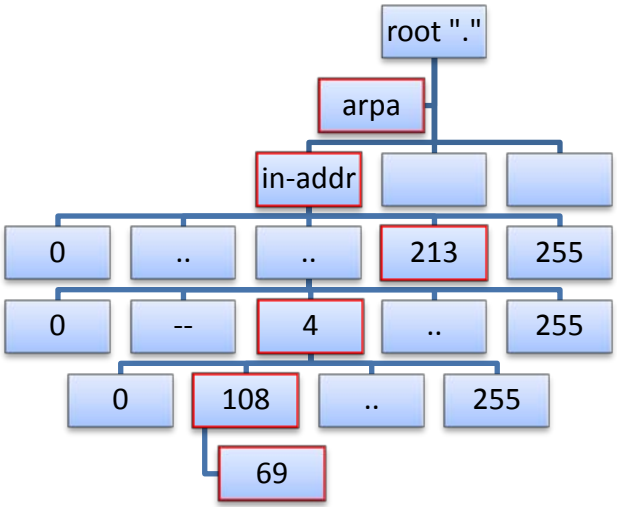

Ilustración 2. Dominio 69.108.4.213.in-addr.arpa

En la siguiente ilustración se muestra un ejemplo de resolución inversa utilizando la utilidad dig

```
kuko@DNS ~ dig -x 213.4.108.69

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> -x 213.4.108.69
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63960
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;69.108.4.213.in-addr.arpa.     IN      PTR

;; ANSWER SECTION:
69.108.4.213.in-addr.arpa. 21599 IN     PTR     www.interdomain.org.

;; Query time: 144 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Mar  3 10:21:32 2014
;; MSG SIZE  rcvd: 76
```

Ilustración 3. Resolución inversa de IP 213.4.108.69

SERVIDORES DE NOMBRES

Un servidor de nombres es una computadora que se encarga de almacenar y proporcionar información sobre el espacio de nombres y espacio de direcciones. Proporciona de este modo la traducción (resolución) de un nombre a dirección IP y viceversa. Esta información se denomina Registro DNS y será detallado adelante.

- **Servidores Autoritativos**

Un servidor de nombres autoritativo es aquel que mantiene zonas almacenadas localmente y proporciona respuestas a solicitudes de las mismas. Recuérdese que la zona es un conjunto de dominios los cuales contienen a su vez información de registros. Los servidores autoritativos sólo proporcionan respuestas de los dominios para los que han sido configurados por el administrador. Los servidores autoritativos *pueden ser* ***maestros o esclavos***. En los ***maestros*** ,o primarios, se guardan y administran las versiones definitivas de los registros que son transferidas a servidores autoritativos ***esclavos,*** que guardan una copia que es actualizada cada vez que se produce un cambio. Esta actualización se conoce como **transferencia de zona**.

Cuando un nombre de domino se registra a través de un servicio registrador, se solicitará la asignación de un servidor primario y al menos un servidor secundario para proporcionar redundancia en caso de inoperatividad de alguno de los servidores y mantener accesible la información del dominio. Generalmente, *los servidores primarios son servidores autoritativos maestros y los secundarios, servidores autoritativos esclavos*. Cuando es un servidor autoritativo el que proporciona la respuesta al cliente, ésta es marcada con un *flag* que indica que es una respuesta autoritativa AA (*Authoritative Answer*). Cuando el cliente recibe la respuesta de otro servidor caché intermedio, la respuesta se recibe como no autoritativa. En la siguiente ilustración se observa la diferencia entre una respuesta obtenida del servidor autoritativo y una obtenida desde un servidor caché:

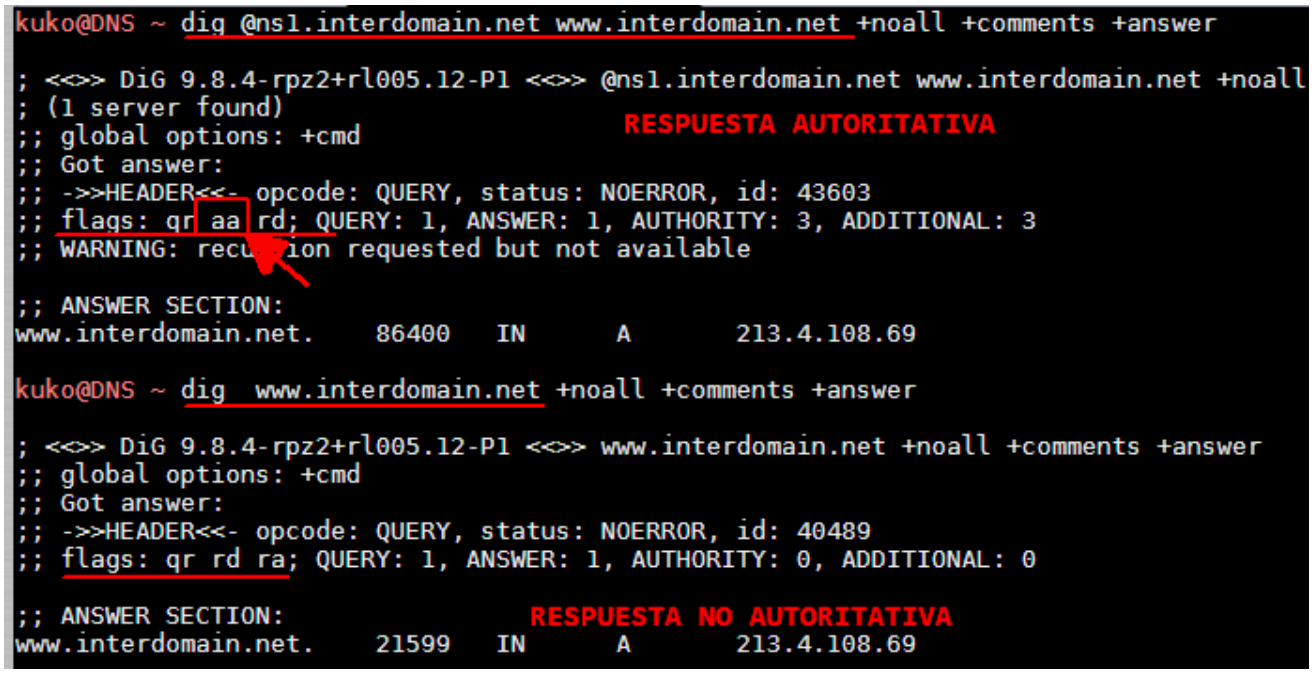

Ilustración 4. Respuesta autoritativa y respuesta cacheada. Obsérvese el flag aa (autoritative answer)

- **Servidores caché**

Los servidores DNS cachés almacenan información de consultas (*queries*) DNS por un determinado tiempo denominado TTL (*Time To Live*) de cada registro DNS. Los servidores caché optimizan el uso de red reduciendo el tráfico DNS en Internet puesto que almacenan los registros consultados, pudiendo ofrecerlos directamente sin tener que repetir la consulta recursiva. Igualmente, reducen la carga sobre los servidores autoritativos, especialmente los de la zona raíz o *root servers*.

RESOLVERS

Los *resolvers* son programas o servicios con los que el usuario interacciona en su máquina para generar una consulta DNS. Son los encargados de formatear la consulta según las especificaciones necesarias del mensaje DNS y manejar la comunicación con el servidor para enviar y recibir la información de los registros requeridos.

REGISTROS DNS. FORMATO Y TIPOS

Un nombre de dominio se identifica con un nodo en la jerarquía DNS. Cada nodo contiene un conjunto de información conocido como registros (*Resource Registers,* RR) de los cuales es responsable o autoridad.

Registros DNS: Resource Records (RR)

Existen varios tipos de registros, cada uno identificando un tipo de información. Esta información es formateada en un registro que se compone de 6 campos, que se utiliza al transmitir dicha información en los mensajes DNS. En la siguiente tabla se describen los 6 posibles campos en un mensaje DNS

Campo	Descripción	Longitud (bytes)
NAME	Nombre del dominio al que pertenece el registro	Cadena variable
TYPE	Código del tipo de registro	2 bytes
CLASS	Código de clase del registro	2 bytes
TTL	Tiempo en segundos durante el cual el registro es cacheado	4 bytes
RDLENGTH	Indica la longitud en bytes del campo RDATA	4 bytes
RDATA	Cadena de longitud variable que describe el registro de acuerdo al tipo y clase del mismo	Cadena variable

Tabla 1. Formato de registro. Resource Record (RR)

El campo **TYPE** contiene un código que identifica de qué tipo de registro se trata. Existen multitud de tipos de registros definidos en distintos RFCs[2] para cubrir otras tantas funcionalidades. Algunos de los tipos más comunes se muestran en la siguiente tabla:

TIPO (valor campo TYPE)	Función
A = *Address* – (Dirección)	Traduce (resuelve) nombres de recursos a direcciones IPv4
AAAA = *Address* – (Dirección)	Traduce (resuelve) nombres de recursos a direcciones IPv6
CNAME = *Canonical Name* – (Nombre Canónico)	Crear nombres adicionales, o alias, para el recurso
NS = *Name Server* – (Servidor de Nombres)	Indica qué servidor(es) almacenan la información del dominio consultado
MX = *Mail Exchange* (Registro de Intercambio de Correo)	Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio. Tiene un balanceo de carga y prioridad para el uso de uno o más servicios de correo.
PTR = *Pointer* – (Puntero)	Inverso del registro A, traduciendo IPs en nombres de dominio.
SOA = *Start of authority* – (Autoridad de la zona)	Indica el servidor DNS primario de la zona, responsable del mantenimiento de la información de la misma.
HINFO = *Host INFOrmation* – (Información del recurso)	Descripción de la CPU y sistema operativo que almacena la información de un dominio. Suele ocultarse.
TXT = TeXT - (Información textual)	Permite a los dominios proporcionar datos adicionales.
LOC = LOCalización	Permite indicar las coordenadas geográficas del dominio.
SRV = SeRVicios -	Información sobre los servicios que ofrecidos
SPF = *Sender Policy Framework* -	Ayuda a combatir el Spam. En este registro se especifica cual o cuales hosts están autorizados a enviar correo desde el dominio dado. El servidor que recibe, consulta el SPF para comparar la IP desde la cual le llega. Su uso se pretende abandonar a favor de registro TXT[3].
ANY = Todos	Para solicitar todos los registros disponibles

Tabla 2. Valores más habituales campo TYPE

[2] RFC: Las Request for Comments ("Petición De Comentarios" en español) son una serie de notas sobre Internet, y sobre sistemas que se conectan a internet, que comenzaron a publicarse en 1969. Se abrevian como RFC.
[3] IETF. RFC 6686 Resolution of the Sender Policy Framework (SPF) and Sender ID Experiments https://tools.ietf.org/html/rfc6686

El campo **CLASS** es comúnmente fijado al valor IN (Internet) para registros DNS relacionados con *hostnames*, servidores o, en resolución inversa, direcciones IP. Existen además las clases CH (Chaos) y HeSiod (HS) para otros sistemas menos comunes.

En el campo **TTL**, un valor numérico que indica el tiempo en segundos que se cacheará el registro. Un valor 0 indica validez sólo para la transacción en curso y el registro asociado no será almacenado en caché. Los registros SOA siempre tienen TTL igual a 0.

En el campo **RDATA** se describe el contenido del registro según el tipo indicado en el campo TYPE: SOA, A, NS, MX, etc. El tamaño de esta información se indica en el campo **RDLENGTH**.

COMUNICACIONES Y TRANSACCIONES DNS

PROTOCOLO DNS

DNS usa para las comunicaciones el puerto 53, tanto para datagramas UDP como paquetes TCP. Generalmente, en la actividad DNS se usan datagramas UDP ya que requieren menos recursos de proceso y de red. Cuando el tamaño de una respuesta supera el máximo especificado en el estándar DNS para un paquete UDP (512 bytes, sin contar cabeceras IP o UDP) y no se usa ENDS0[4] (que permite extender la consulta DNS hasta 4Kb) se pasa a utilizar TCP por la necesidad de tener un control sobre en la capa de transporte, para asegurar una correcta transmisión. En este caso, el servidor responde con el flag *truncated* (TC) y el cliente reintenta la respuesta sobre TCP. Otras operaciones como las de transferencia de zona se usan directamente TCP.

La implementación de DNS tomando UDP como base principal para sus comunicaciones supone el origen de multitud de amenazas relacionadas con la falta de fiabilidad intrínseca a las transmisiones de este protocolo. Al no haber un control sobre los datos transmitidos por UDP, se da por sentado que la fuente es fiable y que la respuesta siempre es recibida por el solicitante. Esto tiene gran impacto en la seguridad de las comunicaciones y constituye un vector de ataque fácilmente explotable. Sobre esto se discutirá más adelante en relación a la seguridad del protocolo.

MENSAJES DNS

- **Formato genérico de mensaje DNS**

Todas las comunicaciones en el protocolo DNS siguen un formato estándar llamado ***mensaje***. El mensaje se divide en una cabecera HEADER y 4 secciones: QUESTION, ANSWER, AUTHORITY y ADITTIONAL. Dependiendo del tipo de mensaje alguna sección puede ser nula. La cabecera HEADER siempre está presente pues contiene importante información sobre el contenido mensaje.

[4] Véase la sección 2.4.2.4. Formato Mecanismo de Extension para DNS EDNS0

SECCION	Descripción
HEADER	Contiene información sobre el tipo de mensaje. Incluye campos que informan sobre el número de entradas en otras secciones del mensaje.
QUESTION	Contiene una o más solicitudes de información (queries) que se envían al servidor DNS
ANSWER	Contiene uno o más registros que responden a la(s) solicitud(es)
AUTHORITY	Contiene uno o más registros que apuntan al servidor autoritativo del dominio en cuestión
ADDITIONAL	Registros con información adicional no necesaria para responder a la query

Tabla 3. Formato genérico de Mensaje DNS

- **HEADER. Cabecera de un mensaje DNS.**

La sección *HEADER* de un mensaje DNS consta de 16 bytes que se desglosan en los siguientes campos:

ID:(16 bits octetos). Los dos primeros bytes se destinan al ID del mensaje. Este campo es especialmente importante ya que identifica el paquete y será el objeto a atacar cuando se intenta falsificar un mensaje.

QR (1 bit): Utilizado para indicar si se trata de una consulta (0) o una respuesta (1).

Opcode (4 bits): Indica el tipo de query, consulta estándar, consulta inversa, notificación, actualización dinámica o estado servidor.

Flags (4 flags de 1 bit). **AA**: Respuesta autoritativa. **TC:** Truncation. Indica que el mensaje está truncado al haberse superado la longitud máxima permitida en la transmisión. **RD:**Recursion Desired, especifica que se solicita una consulta recursiva. **RA:** Recursion Available. Denota en una respuesta que se ofrece la posibilidad de recursión.

Z (3 bits): Reservado para futuros usos.

RCODE (4 bits): Campo fijado en las respuestas informado del estado de la misma: No Error, Error de formato, error del servidor, rechazada.

QDCOUNT,ANCOUNT,NSCOUNT,ARCOUNT (16 bits) Campos destinados a especificar el número de entradas o registros en las secciones query, answer, authority y additional.

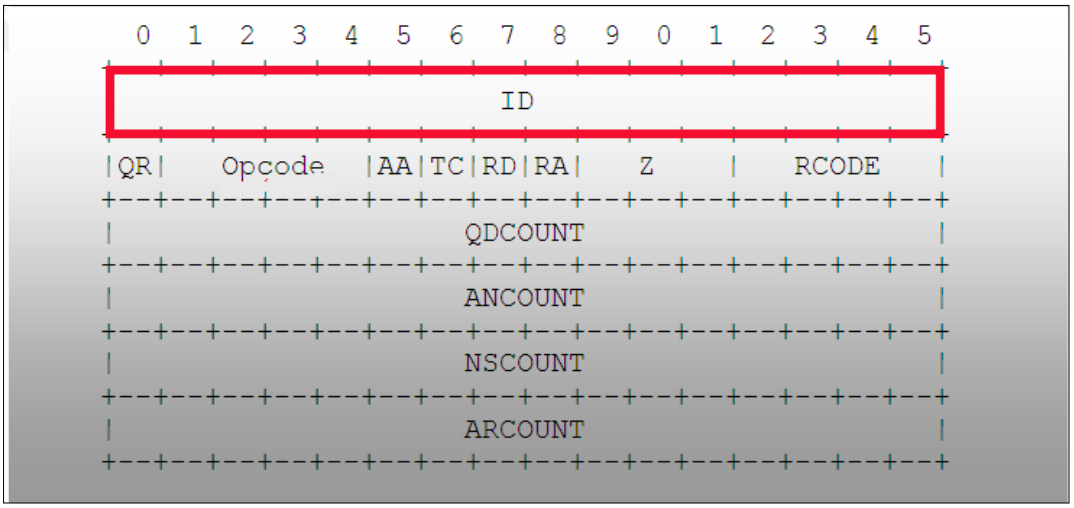

Ilustración 5. Sección Header en mensaje DNS.
Campo ID, identificador del mensaje. Fuente: RFC1035[5]

- **Formato de mensajes DNS de consulta y respuesta**

Dependiendo de si el mensaje DNS es una pregunta o una respuesta, alguno de los campos puede no aparecer.

Mensajes DNS de consulta (QUESTION)

En *mensajes DNS de pregunta* aparece el campo **QUESTION**, que contiene la pregunta que se solicita al servidor DNS y que el cliente (*resolver*) formatea siguiendo la siguiente estructura de 3 campos que se refleja en la siguiente tabla:

Mensaje DNS consulta. Campo QUESTION.	
QNAME	**Indica el dominio por el que se pregunta**
QTYPE	**Tipo de información (registro) que se requiere en la consulta.**
QCLASS	**Clase de registro**

Tabla 4. Campo QUESTION en una mensaje DNS de pregunta.

En el campo QYTPE donde se indica el tipo de información, son valores válidos todos los tipos definidos como TYPE (Tabla 2. **Valores más habituales campo TYPE.**) y otros valores adicionales para especificar operaciones, como por ejemplo AXFR en transferencias de zonas. En el campo QCLASS son válidos todos valores CLASS (IN, CH, HS) definidos en el formato de registro. Generalmente toma el valor IN. (INternet). Un ejemplo se muestra en la Ilustración 6. Ejemplo de consulta (query) DNS tipo A

[5] RFC1035. http://www.ietf.org/rfc/rfc1035.txt

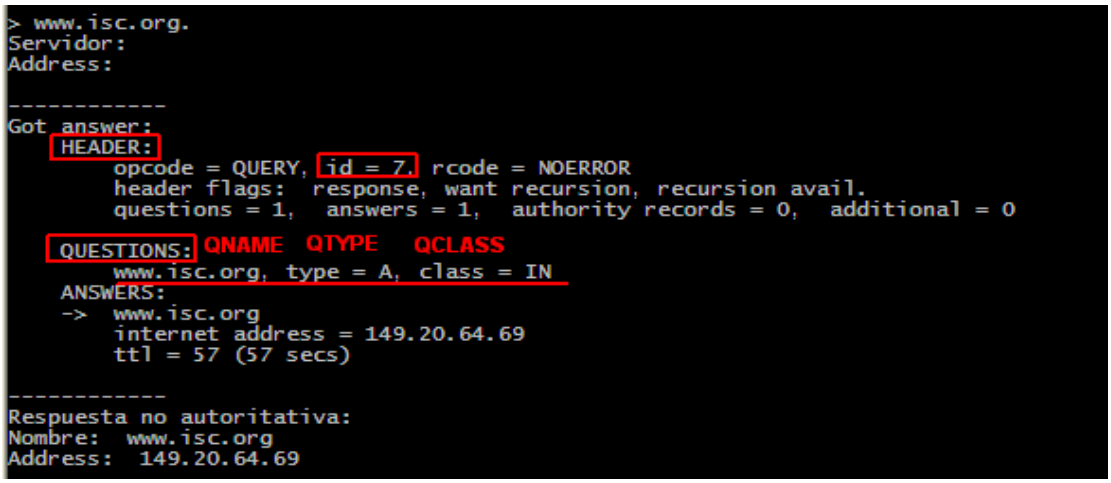

Ilustración 6. Ejemplo de consulta (query) DNS tipo A

Mensajes DNS de respuesta

En mensajes *DNS de respuesta* aparecen las secciones **ANSWER, AUTHORITY y ADDITIONAL** todas siguiendo el formato de registro DNS (RR) descrito en la Tabla 1. **Formato de registro. Resource Record (RR).** Se compone pues, de los campos NAME, TYPE, CLASS, TTL, RDLENGTH, y RDATA.

- **Formato Extension Mechanism for DNS EDNS0**

Los mensajes DNS pueden hacer uso del mecanismo de extensión definido en el RFC2671 y posibilitar la comunicación de mensajes con una extensión mayor que la prefijada de 512 bytes sobre UDP. Mediante esta funcionalidad se posibilita el uso de un buffer mayor para datagramas UDP. Es generalmente usado con operaciones que requieren un tamaño superior a 512 bytes , o en operaciones de transferencia de zona. En versiones de Bind a partir de la 9, el formato EDNS0 es el empleado por defecto. Un servidor advierte de su capacidad para usar EDNS0 especificando un pseudoregistro OPT en la sección ADDITIONAL del mensaje DNS. Esto puede identificarse en una consulta con la utilidad *dig* donde aparece como “OPT PSEUDOSECTION”, como puede observarse en la siguiente ilustración:

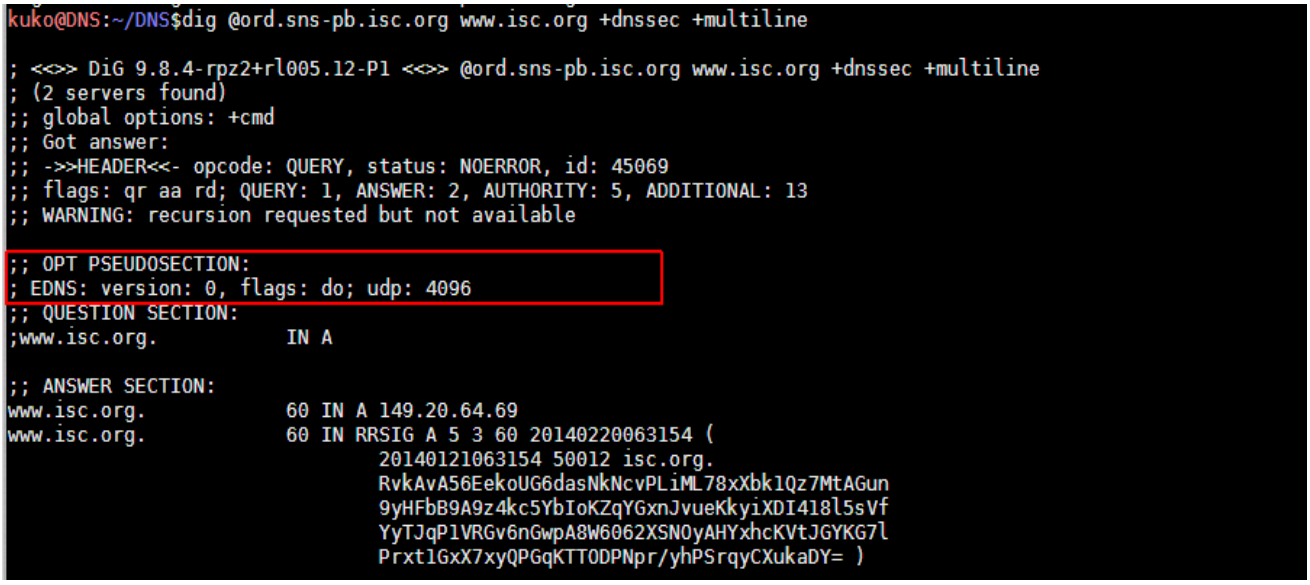

Ilustración 7. Formato extendido EDNS0. 4096 bytes UDP

Véase también el ejemplo en el Anexo B, ejemplos avanzados.

TRANSACCIONES DNS

Las transacciones más comunes en DNS son:

Consultas/respuestas DNS (Queries).

Transferencias de Zona: Mecanismo de replicación de ficheros de zona entre servidores.

Actualizaciones Dinámicas: Mecanismo usado para actualizar los ficheros de zona de un servidor DNS.

Notificaciones: Transacción que usa un servidor autoritativo para notificar cambios en su base de datos de zonas.

- **Consultas DNS (Queries)**

Es el tipo más común de transacción DNS. Las queries pueden ser de consulta o de respuesta. Una consulta DNS tiene origen en un *resolver* con destino un servidor DNS autoritativo o caché.

Los consultas DNS realizadas por un *resolver* pueden ser *iterativas* o *recursivas*:

a) La ***consulta iterativa*** es aquella en la que el *resolver* (cliente) requiere al servidor DNS devolver la mejor respuesta basada en sus ficheros de zona o caché. Si el recurso solicitado no se encuentra en el propio servidor, éste en su respuesta devolverá un *referral,* es decir, un puntero al servidor autoritativo del nivel más bajo del dominio solicitado, al que debe dirigirse a continuación para seguir la iteración. Por ejemplo, si se pregunta al servidor A, por el dominio *www.midominio.org*, y el servidor A no dispone de esa información, le contestará con el *referral* (servidor autoritativo) del dominio root "." para que le solicite el nombre. A continuación, el *resolver* continuará la consulta iterativamente, donde preguntará por el dominio al servidor raíz, el cual le devolverá el *referral* (servidor autoritativo) del dominio .org. El resolver repite (itera) el proceso hasta que, recorriendo los *referrals*, llega al servidor autoritativo del dominio deseado donde obtendrá la respuesta o un error (si no existe el registro). Normalmente el resolver final solicita *consulta recursiva* al servidor DNS que actúa como *resolver* intermediario (caché recursivo) evitando al cliente realizar la iteración.

b) La ***consulta recursiva*** es aquella en que el *resolver* solicita del servidor DNS una respuesta final o un error (si el recurso no existe). En este caso el servidor DNS actúa de intermediario realizando las consultas iterativas necesarias para obtener la respuesta o el error.

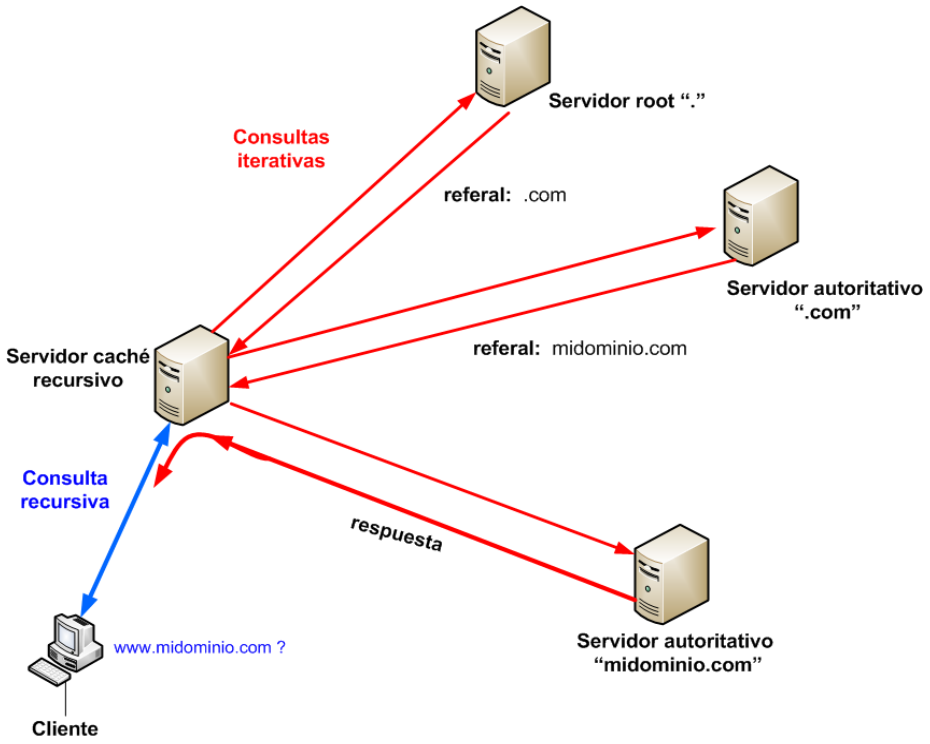

Ilustración 8. Consultas iterativas y recursivas

- **Mecanismo de resolución en una consulta DNS**

El proceso que se sigue en una resolución DNS es el siguiente. El cliente (*resolver*) hace llegar la consulta al servidor DNS:

a) Si el servidor DNS está configurado como autoritativo y recibe una consulta DNS sobre un dominio sobre el que él es autoritativo, devolverá la respuesta consultando los registros almacenados en su configuración y devolviendo la respuesta marcada como *Authoritative Answer* en la sección "ANSWER" del mensaje de respuesta. Si no tiene la información, responde con el mensaje NXDOMAIN (*Non-Existent-Domain*).
b) Si el servidor DNS es autoritativo y no configurado como recursivo y recibe una consulta sobre un dominio sobre el que no es autoritativo, responderá con un mensaje conteniendo registros en la sección "AUTHORITY" y en la sección ADDITIONAL informando al resolver que no proporciona recursión y donde debe dirigir su consulta para obtener información autoritativa del dominio solicitado. Se conoce como *Referral Response*.
c) Si el servidor DNS no es autoritativo, pero está configurado como recursivo y recibe una consulta, éste inicia consultas iterativas (recursión) para encontrar el servidor autoritativo del dominio. Una vez obtiene respuesta devuelve el registro al cliente (*resolver*) indicando que se trata de una respuesta no autoritativa. La información la guarda en caché, de modo que si vuelve a ser preguntado por el mismo recurso y el tiempo con que el registro está marcado para "caducar" (TTL, o *Time To Live*) no ha pasado, contestará consultado esta caché.

Un ejemplo del flujo en una consulta recursiva para el dominio www.ejemplo.com sería:

1. El *resolver* lanza la consulta, preguntando al servidor DNS la resolución del nombre www.ejemplo.com.

2. El servidor, que no tiene la respuesta inicia la consulta iterativa para obtener el registro. Para ello pregunta a los servidores raíz por el dominio com.
3. La pregunta llega a los servidores DNS raíz que contestan con un registro AUTHORITY y ADDITIONAL para referir (*referral*) los servidores autoritativos del dominio com.
4. Se pregunta a los servidores referidos por el root server por el dominio www.ejemplo.com.
5. El servidor DNS autoritativo de la zona com , igualmente le devuelve una respuesta *referral* con el puntero al servidor DNS autoritativo del dominio ejemplo.com.
6. Cuando el servidor recursivo, a través de las Respuestas de Referido (*Referral Responses*) conoce las direcciones de servidores DNS autoritativos del dominio ejemplo.com. les redirige la pregunta www.ejemplo.com
7. El servidor DNS autoritativo de ejemplo.com busca en la información almacenada el registro solicitado y devuelve la respuesta
8. Finalmente, el servidor recursivo cachea la respuesta con el TTL configurado y resuelve la solicitud del resolver

En las *ilustraciones 9 y 10* podemos ver gráficamente el proceso arriba descrito:

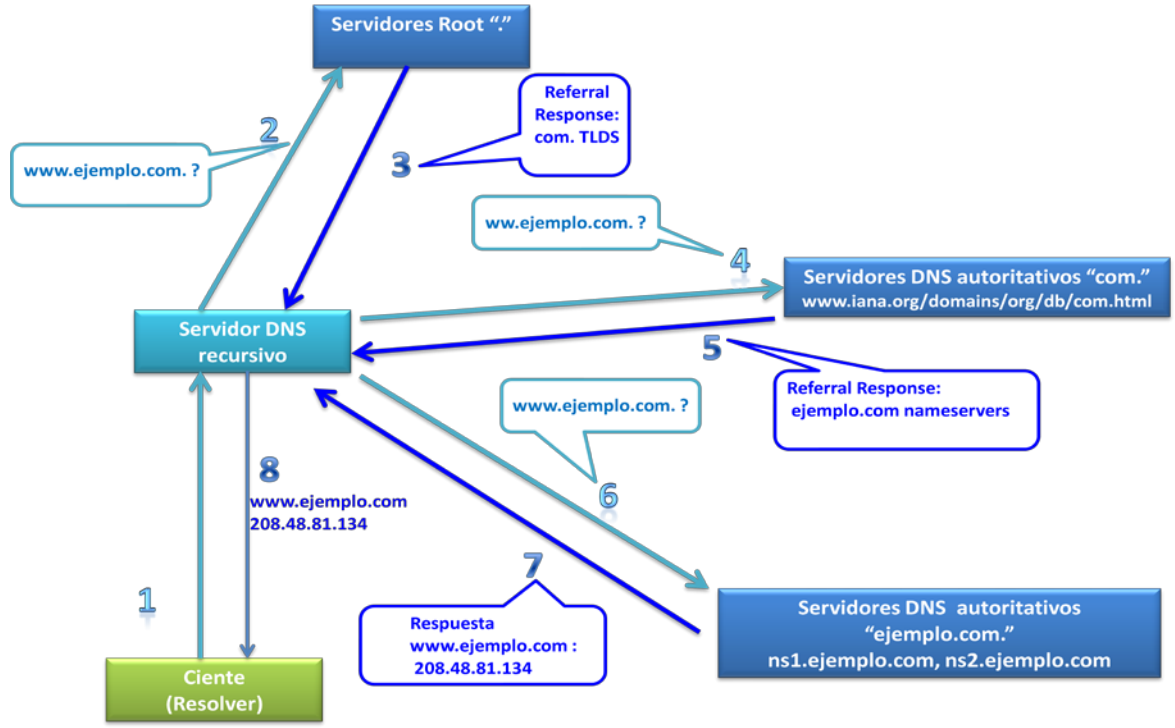

Ilustración 9 Sucesión de consultas en una resolución recursiva

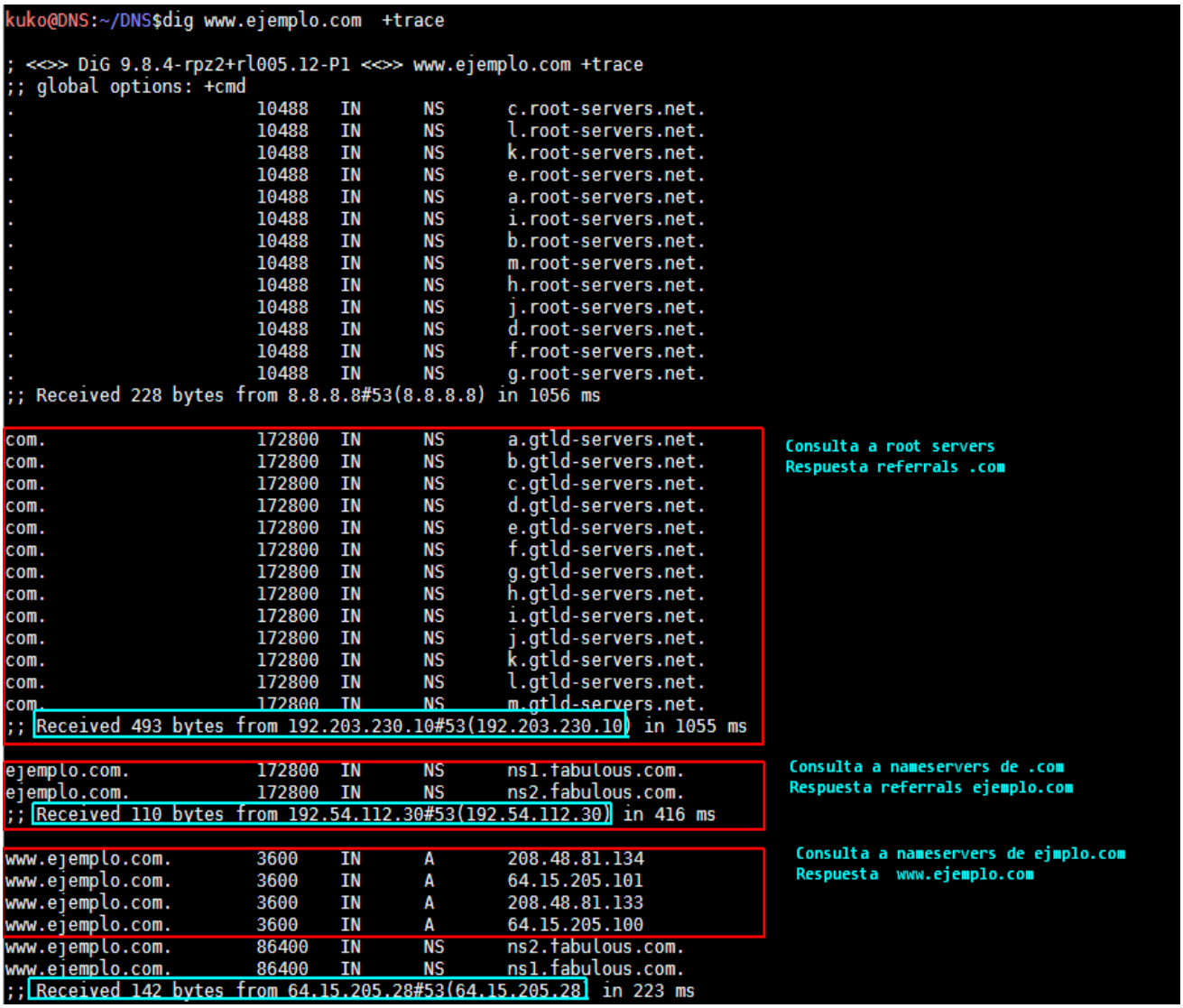

Ilustración 10. Sucesión de consultas iterativas en una resolución DNS

- **Transferencias de zona**

La transferencia de zona es una transacción por la cual un servidor DNS **secundario** (esclavo) actualiza los contenidos de zona desde un servidor **primario** *(master)* y de este modo, mantener una copia sincronizada con las base de datos maestra. La transacción se inicia con un mensaje de transferencia de zona (*zone transfer query)* donde se solicitan todos los registros (*resource records* o RRs) de un dominio. La solicitud de transferencia de zona se genera en el servidor secundario de forma automática bajo dos posibles circunstancias:

1 Se recibe un mensaje de notificación <<NOTIFY>> por parte del primario para dar a conocer que sean producido cambios o modificaciones en los contenidos de la zona.
2 Ha transcurrido el tiempo especificado en el valor <<Refresh >> del campo RDATA del registro SOA de la zona. (*Ilustración 11)*

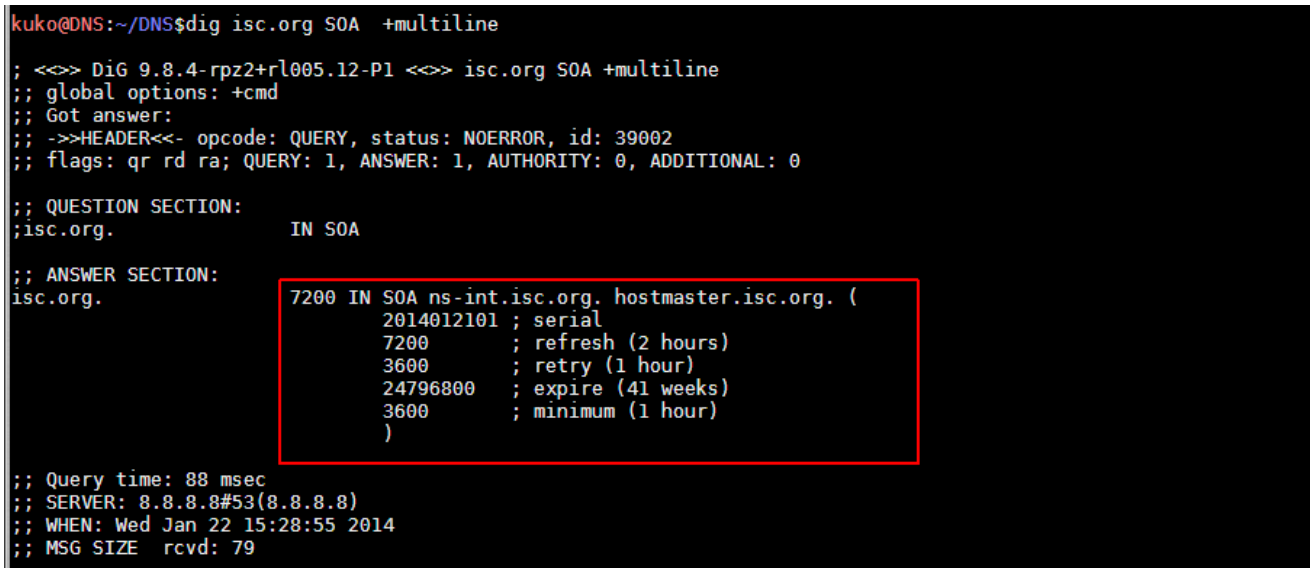

Ilustración 11. Registro tipo SOA (Start of Authority). Obsérverse el valor refresh

- **Actualizaciones Dinámicas**

En algunos entornos donde el número de registros y zonas crecen y varían frecuentemente, es inviable una gestión manual. Por ejemplo, los registros de nombres tipo "A" en un servicio de DHCP y sus inversos "PTR" dentro de una gran compañía proveedora de servicios. De esta necesidad surge el concepto de actualización dinámica. El mecanismo de actualización dinámica proporciona dos operaciones: *añadir* o *borrar* registros en un fichero de zona. El caso de una actualización queda cubierto por un borrado y la posterior recreación del mismo. La especificación detallada se puede consultar en el RFC 2136 Dynamic Updates in the Domain Name System (DNS UPDATE)[6]

Teniendo en cuenta las dos posibles operaciones añadir/borrar definidas en el proceso de actualización dinámica las posibles acciones pueden ser:

- ✓ Añadir o borrar registros individuales.
- ✓ Borrar conjuntos de registros que cumplan un criterio específico de un dominio determinado.
- ✓ Borrar un dominio existente (por ejemplo, todos los registros del dominio midominio.com).
- ✓ Añadir un nuevo dominio con uno o más registros.

- **Notificaciones y actualizaciones por transferencia de zona.**

Cada vez que se produce un cambio en los ficheros de zona del servidor autoritativo primario, el servidor secundario debe ser informado de dicha modificación y así proceder a actualizar su copia de las zonas, solicitando una transferencia de zona al primario. Por este mecanismo, tras una modificación en los registros de zona, el servidor maestro manda un mensaje (NOTIFY) a los secundarios para advertir este cambio. Adicionalmente, aunque menos preciso que el NOTIFY del primario, es el proceso de transferencia de zona desencadenado en el servidor secundario cuando ha caducado el valor "*refresh*" especificado en el registro SOA almacenado. Por defecto, BIND utiliza el método de notificación NOTIFY.

[6] RFC1236. http://www.ietf.org/rfc/rfc2136.txt

Cuando el servidor *master* necesita hacer una notificación, seleccionará los registros NS "*name servers*" especificados en el fichero de la zona y será a quienes mandará la misma. Cuando el servidor esclavo recibe la notificación, reinicia el valor "`refresh`" a cero y comprueba si el "*serial"* (número que identifica la versión de la zona) se ha incrementado, en cuyo caso, solicita la transferencia. Este comportamiento de notificación por defecto puede poder notificar a otros servidores que no aparezcan como "*nameservers*" en los ficheros de zona.En los servidores secundarios, la directiva `allow-notify` identificará los servidores desde los cuales se permite recibir notificaciones.

CONCEPTOS CLAVE

- ***Resolver*:** Un cliente DNS que se encarga de componer y mandar los mensajes DNS a los servidores para obtener la información requerida sobre el dominio deseado.
- ***Open resolver*:** Servidor que ofrece servicio DNS recursivo accesible públicamente a cualquier cliente (*resolver*) que lo solicite.
- **Recursión:** Las acciones que un servidor DNS toma para entregar la información solicitada a un resolver preguntando a otros servidores.
- **Servidor Autoritativo:** El servidor DNS que mantiene, distribuye y responde a solicitudes DNS consultando la información almacenada en sus registros, en inglés, *Resource Records* (RRs). Puede ser primario o secundario.
- **Servidor Autoritativo *Master* (Primario):** Es el servidor DNS autoritativo que contiene almacena las versiones definitivas de los registros que administra.
- **Servidor Autoritativo *Stealth* (Oculto):** Servidor autoritativo primario para algunas zonas pero que no aparece en los registros NS de las mismas. El objeto es mantenerlo oculto a consultas tipo NS, que puede ser útil por ejemplo para servidores internos.
- **Servidor Autoritativo Esclavo (Secundario):** Es el servidor DNS autoritativo que almacena una copia de los registros administrados por el servidor *Master*. Cuando algún cambio se ha producido en los registros del servidor *master* o primario, es notificado a los esclavos que solicitan e inician una transferencia de zona.
- **Servidor DNS caché (*resolver* recursivo):** Es un servidor DNS intermediario que obtiene la respuesta a solicitudes DNS, consultando servidores autoritativos, y la almacena en caché para tenerlas disponibles y servirlas a clientes (*resolvers*). Su función es mejorar el rendimiento de las respuestas y contribuir a reducir la carga de tráfico DNS en internet.
- **Zona:** Base de datos que un servidor autoritativo contiene sobre un conjunto de dominios.
- **Transferencia de Zona:** Comunicación (transacción) entre servidores DNS para la replicación de los contenidos de zona entre ellos. Es una comunicación cliente-servidor TCP con en dos tipos: completa (AXFR) o incremental (IXFR, para actualizar de cambios).
- **FQDN:** *Fully Qualified Domain Name*. Es el nombre absoluto y completo que identifica un recurso en la base de datos distribuida del espacio DNS.
- **Registro DNS ó RR:** *Resource Record*. Contiene la información de un registro DNS que se envía en los mensajes DNS. *Tabla 1. Formato de registro. Resource Record (RR).* Compuesto por seis campos: NAME, TYPE, CLASS, TTL, RDLENGTH y RDATA
- **Mensaje DNS:** Estructura diseñada para la comunicación IP entre los integrantes del espacio DNS y transmitir información. Se compone de 5 campos: HEADER, QUESTION, ANSWER, AUTHORITY y ADDITIONAL. SEGURIDAD EN DNS.

3 SEGURIDAD EN DNS

AMENAZAS Y VULNERABILIDADES EN DNS

En un entorno DNS se identifican varios puntos donde posibles ataques pueden desarrollarse. Estos puntos o "vectores de ataque" se sitúan tanto localmente en el propio servidor DNS y red local, como en las comunicaciones entre servidores y clientes.

VECTORES DE ATAQUE Y AMENAZAS EN UN ESCENARIO DNS

Sobre el escenario típico DNS, representado en la *Ilustración 12* se numeran las 5 áreas principales que presentan una superficie susceptible a amenazas. Estas amenazas y sus posibles contramedidas se pueden resumir en:

1 *Amenazas locales :* En la prevención de las amenazas locales, la solución más sencilla es la implementación de medidas y políticas de seguridad en la red interna. Mecanismos *anti-spoofing*, IDS/IPS[7], así como la protección de los canales de acceso a los servidores y sus archivos sentarán la línea base de protección en esta área.

2 *Amenazas Servidor-Servidor: Actualizaciones dinámicas.* Presentes cuando el tamaño de la organización o el número de servidores a administrar obliga a centralizar la administración de los datos a través de DDNS (*Dynamic DNS*). Una opción válida para asegurar la comunicación sería dedicar un canal de comunicación restringido y/o implementar TSIG.

3 *Amenazas Servidor Master - Servidor Esclavo: Transferencias de zona.* Cuando una organización cuenta con servidores esclavos, tiene la necesidad de ejecutar transferencias de zona maestro/esclavo. En estos casos la solución a considerar es la implementación de TSIG (*Transaction SIGnature),* de modo que las operaciones de transferencia de zona se firmen con una clave conocida por ambas partes. Adicionalmente la seguridad en las comunicaciones puede reforzarse usando SSL/TLS. Otras opciones pasarían por un canal de red privado para la transacción, o en caso extremo deshabilitarla y realizarla manualmente, lo cual no es una alternativa funcional.

4 *Amenazas Servidor Master - Servidor Cliente Caché/Resolver.* Como se verá en el apartado *Aleatoriedad del ID de transacción y puerto origen*, las mejoras implementadas en las versiones recientes de Bind con la introducción de aleatoriedad en los puertos origen de la consulta, así como en los identificadores de mensaje, dificultan la posibilidad de envenenamiento de caché en los servidores DNS, pero aún así, el ataque sigue siendo posible. La única solución considerada efectiva es adoptar DNSSEC.

5 *Servidor - Cliente (5):* En el flujo de información entre un cliente/*resolver* y un servidor *master* o caché, cabe la posibilidad de ataques locales para interceptar datos y de spoofing

[7] IDS/IPS: Intrusion Detection System/Intrusion Prevention Systems. Sistemas para la prevención y/o detección de amenazas. http://es.wikipedia.org/wiki/Sistema_de_prevenci%C3%B3n_de_intrusos

con objeto de suplantar al servidor DNS. Nuevamente, DNSSEC es la contramedida eficaz contra esta amenaza.

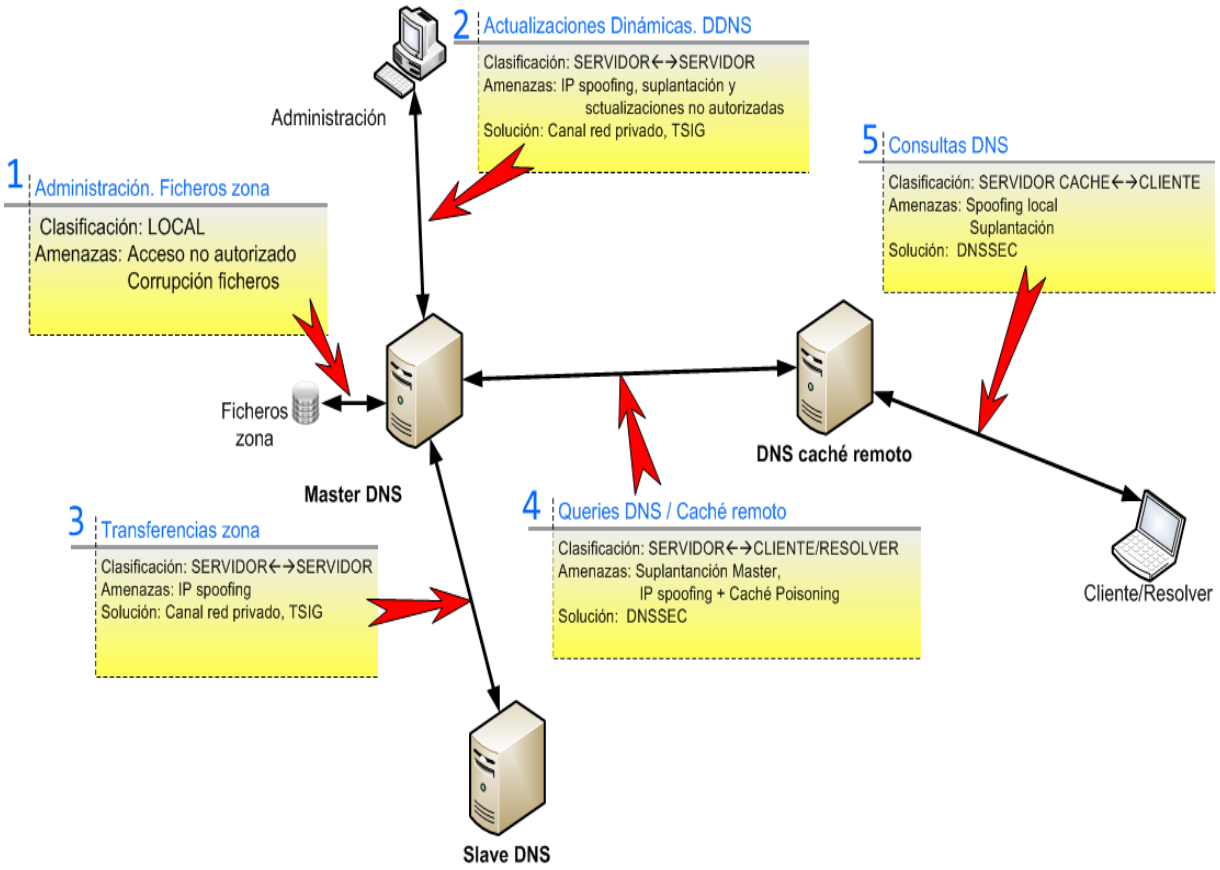

Ilustración 12 Escenario DNS. Vías de ataque y clasificación de amenazas

VULNERABILIDADES Y PUNTOS DÉBILES EN LA ESPECIFICACIÓN DNS.

- **Capa de transporte UDP e IP spoofing.**

La principal debilidad de la que adolece DNS tiene su origen directo en el uso principal del protocolo UDP para transmitir mensajes. UDP es un protocolo de transporte de red en el que prima la velocidad de la transmisión y sobre el cual se envía y recibe la información sin que se haya establecido previamente una conexión y sin confirmación ni control de entrega/recepción de la misma. Esto posibilita el falseo de direcciones IP (*IP spoofing*) y la suplantación de mensajes de consulta/respuesta. Aunque en DNS se contempla el uso de TCP para la transmisión de mensajes, en las especificaciones de implementación se recomienda, por motivos de rendimiento, usar UDP en las consultas. Se sugiere limitar el uso de TCP para transacciones de transferencias de zona o para aquellas consultas que superan el tamaño máximo, establecido en 512 bytes en mensajes sobre UDP. Dada la carencia de control/confirmación en las transmisiones UDP, la responsabilidad final de validar un mensaje recaerá directamente sobre el protocolo DNS.

- **Debilidad en la identificación y validación de mensajes DNS.**

Paralelamente al problema del uso del protocolo UDP en el transporte de mensajes DNS se añaden debilidades de diseño en el aspecto de la identificación y validación de los paquetes que favorecen la falsificación de los mismos.

Como se describe en *Formato genérico de mensaje DNS,* en la sección HEADER de un mensaje DNS se destina el campo ID para identificar el mismo. Este identificador, representado por un número de sólo 16 bits, juega un papel importante en el mecanismo de validación de mensajes de respuesta. Como se explica a continuación, su limitada longitud unido a un débil proceso de validación sobre UDP posibilitan ataques de suplantación IP con relativa facilidad.

Validación de respuestas

No obstante, el campo ID no es el único elemento que se comprueba al validar una respuesta y según se infiere del RFC1034[8], los mínimos requisitos para aceptar una respuesta son:

- El puerto destino en el datagrama de respuesta debe ser el mismo que el puerto origen de la pregunta.
- El ID del mensaje de respuesta debe ser el mismo que el ID del mensaje de pregunta.
- El campo ANSWER debe referir el mismo recurso que el campo QUESTION.
- La sección AUTHORITATIVE contiene los servidores autoritativos de la sección ANSWER.

Todas estas condiciones, a excepción del identificador de transacción ID, son fácilmente identificables y es sencillo construir una respuesta falsa si se conoce el recurso solicitado. De este modo, un atacante que consiga encontrar el ID con el que se emitió la consulta, dispondrá de la información necesaria para falsear una respuesta. Esto, unido a una transmisión sobre UDP, la cual carece de un control/validación de la comunicación, da como resultado que la respuesta falsa será aceptada por el servidor como válida para la consulta realizada previamente.

Identificador de mensaje ID

Debido a la escasa longitud destinada al campo ID del mensaje (16 bits) y a debilidades en la generación de la secuencia de los mismos, computacionalmente es relativamente sencillo construir un número suficiente de ID's en un tiempo limitado para conseguir "acertar" con el ID original. Sin embargo, se han mejorado muchos aspectos en la fortificación del ID y otros valores en el mensaje DNS desde que, en 2008 Dan Kaminsky [9]un investigador en Seguridad IT, presentó su trabajo sobre "*DNS Caché Poisoning", donde* demostró lo sencillo que era conseguir falsificar una respuesta a una consulta DNS y de este modo, lograr que el servidor solicitante almacenase la misma en su caché.

Estas debilidades en la transmisión y validación del mensaje convierten al protocolo DNS en un objetivo fácilmente explotable para dos grandes tipos de ataques basados en *DNS IP spoofing*: *DNS Caché Poisoning y Denegación de servicio por amplificación DNS*.

[8] RCF1034. Domains Names Concepts and facilities. http://tools.ietf.org/html/rfc1034#section-5.3.3
[9] Dan Kaminsky. investigador en seguridad conocido por descubrir el error de envenenamiento de Cache DNS en el 2008 y el Rootkit de Sony. http://es.wikipedia.org/wiki/Dan_Kaminsky

DNS CACHE POISONING Y DNS SPOOFING.

Como ya se ha visto, en una query DNS se usa el campo ID de la sección HEADER del mensaje para identificar la transacción y su correspondiente respuesta. Bajo UDP y sin usar ningún otro mecanismo de control, un atacante puede enviar multitud de respuestas (*flooding*) con distintos ID hasta lograr acertar con el ID generado en la consulta. Si es así, y se consigue hacer llegar la respuesta falsa antes de que llegue la legítima (condición de carrera), el servidor que ha iniciado la consulta la aceptará y la almacenará en su caché. De este modo, es posible "envenenar" la caché de un servidor DNS recursivo con un registro manipulado. A partir de ese momento, durante el tiempo que el registro queda almacenado en la caché (TTL), el servidor víctima redirigirá a una IP ilegítima todas las solicitudes de un *resolver* que le consulte por el recurso manipulado.

DESCRIPCIÓN DEL ATAQUE

La secuencia que se produce en un envenenamiento caché y que se muestra en la *Ilustración 13* es la siguiente:

El atacante, con un servidor DNS bajo su control, solicita un nombre que pertenezca al dominio al cual quiere suplantar (1). Se asegura que este nombre no esté cacheado. El servidor víctima que no encuentra en su cache el recurso pedido, inicia la secuencia de peticiones iterativas empezando por los servidores raíz (2) y recorriendo los TLD que se le indican en los *referrals* (3) hasta que sepa a qué servidor, autoritativo del recurso, dirigir la pregunta (4). En ese instante, el servidor malicioso inicia un bombardeo de respuestas (6) con distintos ID con el objetivo de acertar en una de ellas con el ID coincidente con la query original del servidor víctima. En esas respuestas se indica que el servidor autoritativo (AUTHORITY) para el dominio a suplantar se encuentra en la IP del servidor malicioso. Si se consigue hacer llegar la respuesta falseada (6) antes de que llegue la original (5), el servidor víctima almacenará en caché el registro falseado con la IP del servidor malicioso como servidor autoritativo suplantando al servidor legítimo. La respuesta legítima que llegará después será descartada.

En este momento, el envenenamiento de caché o ***caché poisoning*** del servidor víctima se ha completado con éxito, y todas las solicitudes de resolvers que le lleguen de subdominios pertenecientes al dominio suplantado se dirigirán al servidor malicioso que se encargará de ofrecer las IP bajo su control según le convenga.

Si no se aplica ninguna otra defensa, el atacante sólo tiene que generar con la velocidad necesaria un número de respuestas suficiente para acertar con el ID original.

Gráficamente se puede observar el proceso en la siguiente ilustración

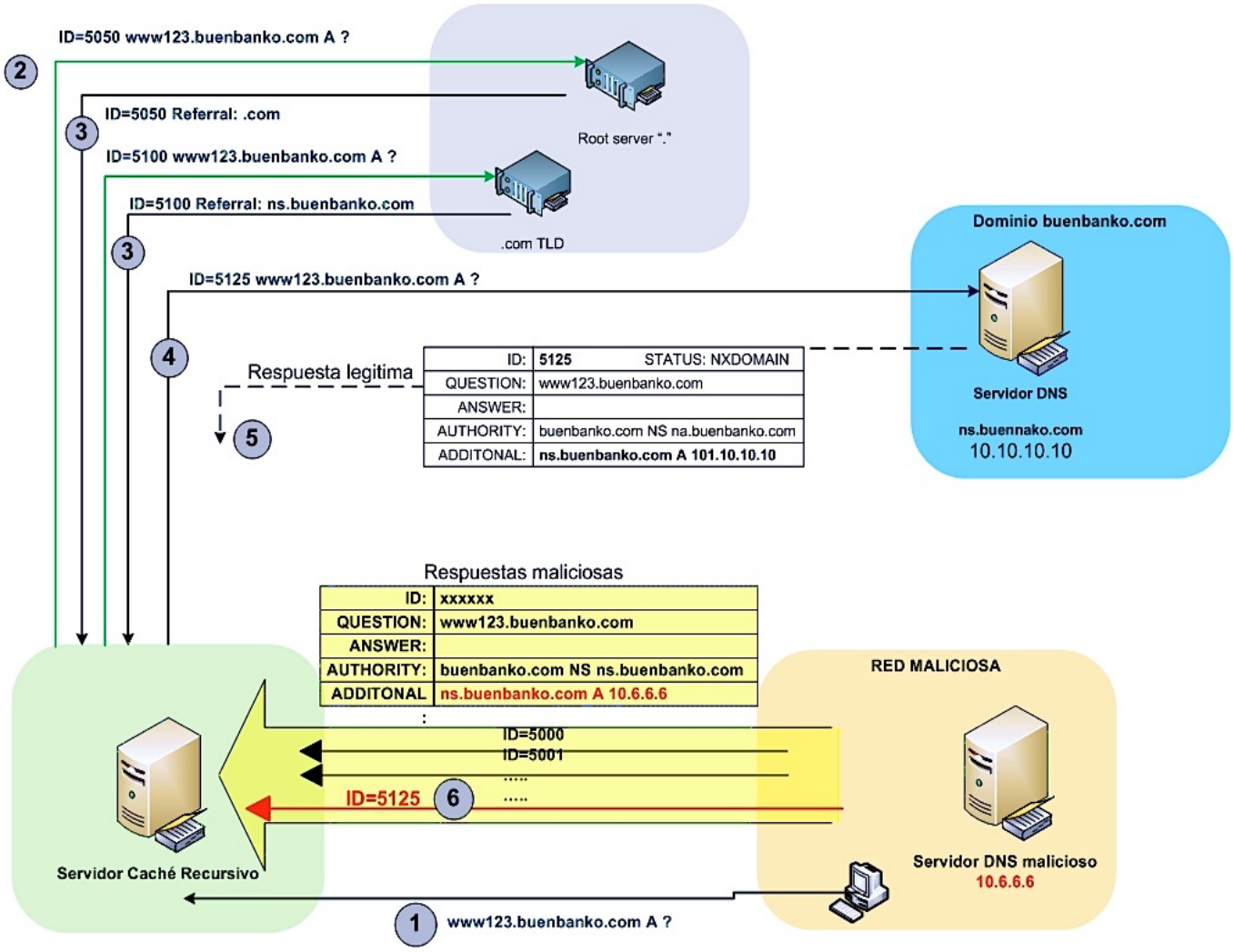

Ilustración 13. Ataque de Caché Poisoning. Averiguado el ID original, se falsea la respuesta

MEDIDAS CONTRA EL ATAQUE DE CACHÉ POISONING

En el documento de la IETF *RFC5452 Measures for making DNS more resilient against Forged Answers*[10] se describe la problemática del DNS spoofing y se trazan líneas base en las implementaciones del software DNS con objeto de detectar y evitar esta amenaza:

> *"Un resolver, con objeto de validar una respuesta DNS, debe realizar las siguientes comprobaciones:*
>
> *a) La sección question del paquete de respuesta es equivalente a la consulta realizada en la query.*
> *b) El campo ID del paquete de respuesta coincide con el ID de la consulta.*
> *c) La respuesta procede de la misma dirección de red a la cual la pregunta fue enviada.*
> *d) La respuesta llega a la misma dirección y puerto desde la cual se emitió la pregunta.*
> *e) Es la primera respuesta recibida en cumplir las cuatro condiciones anteriores."*

Partiendo de las premisas reseñadas, el software DNS debe implementar las medidas descritas a continuación.

[10] RFC5452 . http://tools.ietf.org/html/rfc5452

- **Aleatoriedad del ID de transacción y puerto origen**

Puesto que el campo ID es clave en la identificación de los mensajes DNS, y desde el ataque de caché poisoning demostrado por Dan Kaminsky, se ha tratado de dificultar la predictibilidad del ID de la transacción, añadiendo aleatoriedad en su generación en las consultas y así hacerlo menos predecible. Esta medida, a pesar de todo, fue insuficiente debido a la limitación del campo ID, 16 bits que resulta en 2^{16} = 65535 valores posibles. Posteriormente se introdujo aleatoriedad en el puerto origen de la consulta, que tradicionalmente estaba fijo en el puerto 53. Los puertos disponibles para asignar aleatoriamente descontando los privilegiados 1-1023 son pues del 1024-65535 es decir 2^{11} = 2048 puertos. El resultado total de estas dos medidas da 2^{11} x 2^{16} = 134.215.680 posibles valores. Con este número de posibilidades se dificulta enormemente obtener el ID de la transacción en el tiempo limitado disponible hasta la llegada de la respuesta legítima (sin considerar una denegación de servicio para retrasarla).

- **0x20 bit encoding**

Como complemento a la aleatoriedad en el ID de transacción y el puerto origen, existen otros factores complementarios que algunos fabricantes como Nominum[11] implementan. La técnica *0x20 bit encoding*[12] consiste en realizar las consultas DNS alternado aleatoriamente mayúsculas y minúsculas. Puesto que el protocolo DNS no distingue entre ambas, se resolverá de igual forma el dominio *WwW.EjEmPlo.Com* que *www.ejemplo.com*, sin embargo la implementación del software solo validará aquella respuesta que coincida en la capitalización de los caracteres con la consulta. De este modo se dificulta la posibilidad de aceptar una respuesta falsa si no es coincidente con el formato de mayúsculas y minúsculas de la consulta.

- **Validación de respuestas y detección de spoofing. Retransmisión sobre TCP**

Introduciendo mecanismos aleatorios para seleccionar ID y puerto origen en la generación de consultas se consigue dificultar el ataque de *spoofing*, pero teóricamente aún es posible. Por ello, se hacen necesarias comprobaciones adicionales sobre la respuesta en sí.

Un buen *resolver* debe detectar intentos de *spoofing* aplicando criterios como los reseñados en el RFC5452, de modo que, si aplicando esos criterios, se están descartando muchos paquetes en respuestas a una consulta determinada, se abandone la petición sobre UDP y se reintente a través de TCP.

- **Limitar recursión**

Una mejora complementaria a las anteriores es, en la medida de lo posible, limitar la recursión y de este modo reducir la superficie de exposición a atacantes. De hecho, la gran cantidad de servidores recursivos que ofrecen su servicio públicamente (conocidos como *open resolvers*) constituye la principal fuente usada para establecer ataques de gran potencia, como los de denegación de servicio por amplificación DNS.

---

[11] Nominum Vantio CacheServer. http://nominum.com/infrastructure/engines/caching-dns/

[12] IETF. Use of Bit 0x20 in DNS Labels http://tools.ietf.org/html/draft-vixie-dnsext-dns0x20-00

- **Solución al *poisoning*: DNSSEC**

Finalmente, se considera que la solución más eficaz para eliminar esta amenaza pasa por la implementación de DNSSEC[13]. Conceptos sobre esta mejora de DNS se describen en el apartado *6 DNSSEC.*

[13] DNSSEC, del inglés Domain Name System Security Extensions, es un conjuto de especificaciones destinadas a autenticar el origen de los datos en los mensaje DNS

4 ATAQUES DE DENEGACIÓN DE SERVICIO

El protocolo DNS debido a su vulnerabilidad intrínseca a spoofing IP, se convierte un poderoso aliado a la hora de implementar ataques de denegación de servicio. Esto, unido a su amplia distribución y acceso a nivel mundial hacen de este tipo de ataque uno de los más eficaces y utilizados.

ATAQUE DE AMPLIFICACIÓN DNS

DESCRIPCIÓN DEL ATAQUE

Una vez más, el uso de UDP en el transporte de mensajes DNS, así como la enorme cantidad de servidores recursivos accesibles en internet (*open resolvers*) posibilita el uso del servicio para establecer ataques distribuidos de denegación de servicio hacia otros servidores. Uno de los rincipales, basado en DNS, es el ataque *Amplificación DNS.*

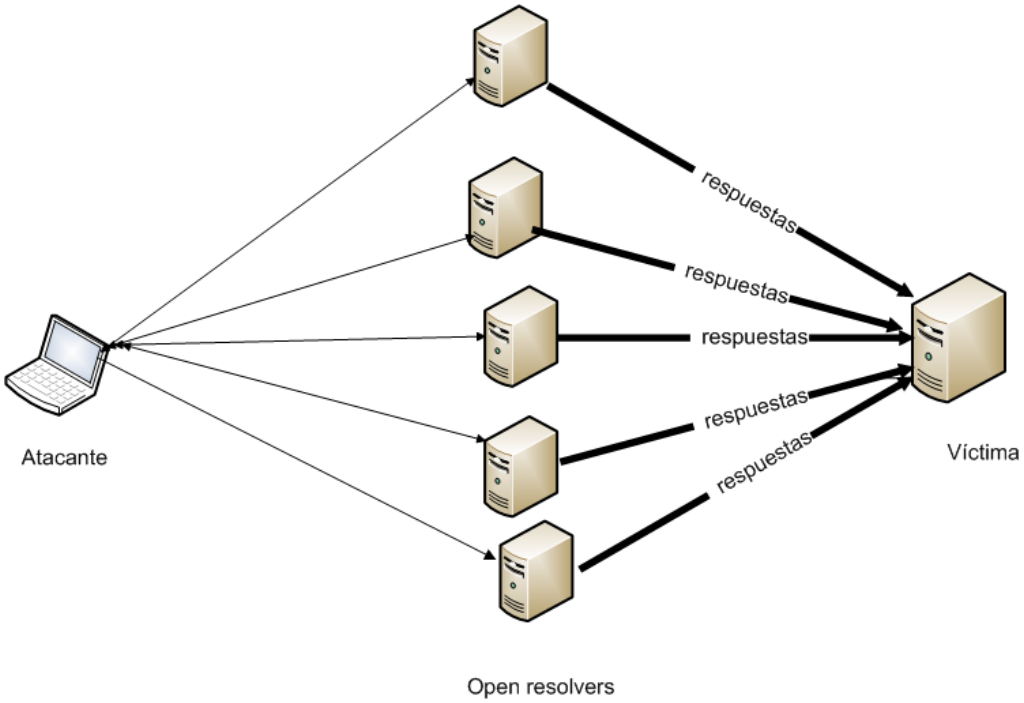

Ilustración 14. Ataque de amplificación DNS

En un ataque de amplificación DNS (*Ilustración 14*) se pretende desbordar la capacidad de respuesta de un servidor haciéndole llegar una gran cantidad de datos DNS. El procedimiento consiste en lanzar consultas DNS a un *open resolver* falseando la IP de origen con la IP del servidor/host a atacar. Esta técnica es conocida como *IP spoofing* y es ampliamente utilizada en protocolos con base UDP donde la falta de control sobre conexión posibilita el falseo de direcciones IP. En las consultas manipuladas, se cambia la dirección IP de origen por la IP objeto del ataque, y se envía de forma masiva a tantos servidores (*open resolvers)* como sea posible. Dichos resolvers, al recibir la consulta responderán enviando la respuesta a la dirección IP indicada. Con un volumen suficiente de consultas, y solicitando recursos cuya respuesta sea mucho mayor que la consulta emitida, por ejemplo registros TXT o ANY, se consigue generar, partiendo un volumen discreto de

datos un gran volumen de tráfico hacia el objetivo. Esto provocará la congestión de los recursos de la víctima y provocar pérdida o degradación del servicio. Cuando el ataque se lanza simultáneamente desde distintos orígenes, el volumen de tráfico es aún más potente. En este caso se denomina ataque de Denegación de Servicio Distribuido o DDoS.

En la siguiente ilustración (*Ilustración 15*) se puede ver claramente el factor de amplificación, donde se obtienen 2066 bytes. Una consulta suele rondar los 66 bytes

```
kuko@DNS ~ dig dhs.gov ANY  +bufsize=4096

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> dhs.gov ANY +bufsize=4096
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56170
;; flags: qr rd ra; QUERY: 1, ANSWER: 21, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;dhs.gov.                       IN      ANY

;; ANSWER SECTION:
dhs.gov.                475     IN      A       173.252.133.166
dhs.gov.                475     IN      RRSIG   A 8 2 900 20140313183320 20140303180959
cnIF2s859F zpa72ufdM3wATFIxTVtDWoYYVwgfGcZw9BBe+vaSZc475HA637bjfMlD NYM=
dhs.gov.                21175   IN      NS      asia2.akam.net.
dhs.gov.                21175   IN      NS      use3.akam.net.
dhs.gov.                21175   IN      NS      use1.akam.net.
dhs.gov.                21175   IN      NS      usc2.akam.net.
dhs.gov.                21175   IN      NS      usw3.akam.net.
dhs.gov.                21175   IN      NS      eur2.akam.net.
dhs.gov.                21175   IN      NS      asia3.akam.net.
dhs.gov.                21175   IN      NS      usw4.akam.net.
dhs.gov.                21175   IN      RRSIG   NS 8 2 28800 20140313230021 201403032249
MH1stBao89Ctp KFkoBF7mu18S5Frrxf2uLQwEyt6HzV2GXtuu4aW/MmOOsNsEL0uh2paL r0g=
dhs.gov.                21175   IN      SOA     ns5.dhs.gov. dnssec1net.cbp.dhs.gov. 206
dhs.gov.                21175   IN      RRSIG   SOA 8 2 28800 20140314084458 20140304074
EAQ9LDprmeRDxc LQ4+6ae9lyCDFGYtJmKOXKZqTLYHMuhzsUoEPlDlAkrUdD68YCKUPwb5 G0o=
dhs.gov.                21175   IN      MX      10 mail.us.messaging.microsoft.com.
dhs.gov.                21175   IN      RRSIG   MX 8 2 28800 20140313082229 201403030816
xN0eiqQZOWdUl GYoJaw0NdXuPtlXLdokkfISJyUeSSagI53TSF+Bk7q2yg4iyjcEApkRx iMA=
dhs.gov.                21175   IN      DNSKEY  256 3 8 AwEAAc9nhbytBN/boahvvLyf2Nk04wnl
2RD86gR/jLWleOz7VevO yDAiqKYH
dhs.gov.                21175   IN      DNSKEY  257 3 8 AwEAAdCPBJACeS7+jhZV2p76YkyjtnL/
kC70snNdWmWLkIbo74P4 SbxCXIzBYU+D/hfhC3pyFl63U8JFlWTBOi1hNmh0dXycUlRTkkxFv4V2 3okFI1+kpF
dhs.gov.                21175   IN      RRSIG   DNSKEY 8 2 432000 20140313052936 201403
2wxtpthR/EI13NmSo 2cpXEc9WnceHp0v0suZVeMNMTvhcjleZcwrAiQA8vsrAMDr3biAZLuwT pPz4LOCO3UqA2
j2NdrBgSEQv5Po9VzaysMq8I9ZHIOtLillDW37y 0IK5bQ==
dhs.gov.                21175   IN      RRSIG   DNSKEY 8 2 432000 20140313052936 201403
z8m4PNHOPKrD2mvE4R bVb6T/fA81WImFaCEKZCEW0FOUn3CTPNXzijVq10/R+Emx6XIcs7f9sr tp0=
dhs.gov.                475     IN      NSEC3PARAM 1 0 10 CE73FD6B7A5EF6FC74F72799
dhs.gov.                475     IN      RRSIG   NSEC3PARAM 8 2 900 20140313010739 201403
LrrW6+Q6l3DKhRAlGhJ +DUMEyng2db2lMfAJDYWotUOZqv73vetBbncTx0Cjq6GZ4+nTK/FBet5 mCA=

;; Query time: 82 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Mar  4 11:31:20 2014
;; MSG SIZE  rcvd: 2066
```

Ilustración 15.Factor de amplificación. Consulta tipo ANY de tamaño 66 bytes, respuesta 2066 bytes

PROTECCIÓN DE UN SERVIDOR EN ATAQUES DE AMPLIFICACIÓN DNS

Globalmente, el problema de denegación de servicio basado en amplificación DNS tiene su origen en la enorme cantidad de servidores DNS distribuidos mundialmente y configurados como *open resolvers*, esto es, que ofrecen su servicio sin ningún tipo de restricción a cualquier solicitante de

Internet. Hay proyectos como *Open Resolver Project* dirigidos a motivar el control de *open resolvers* dirigida a propietarios de servidores DNS de modo que controlen o limiten las consultas recursivas procedentes de localizaciones ajenas a su red[14].

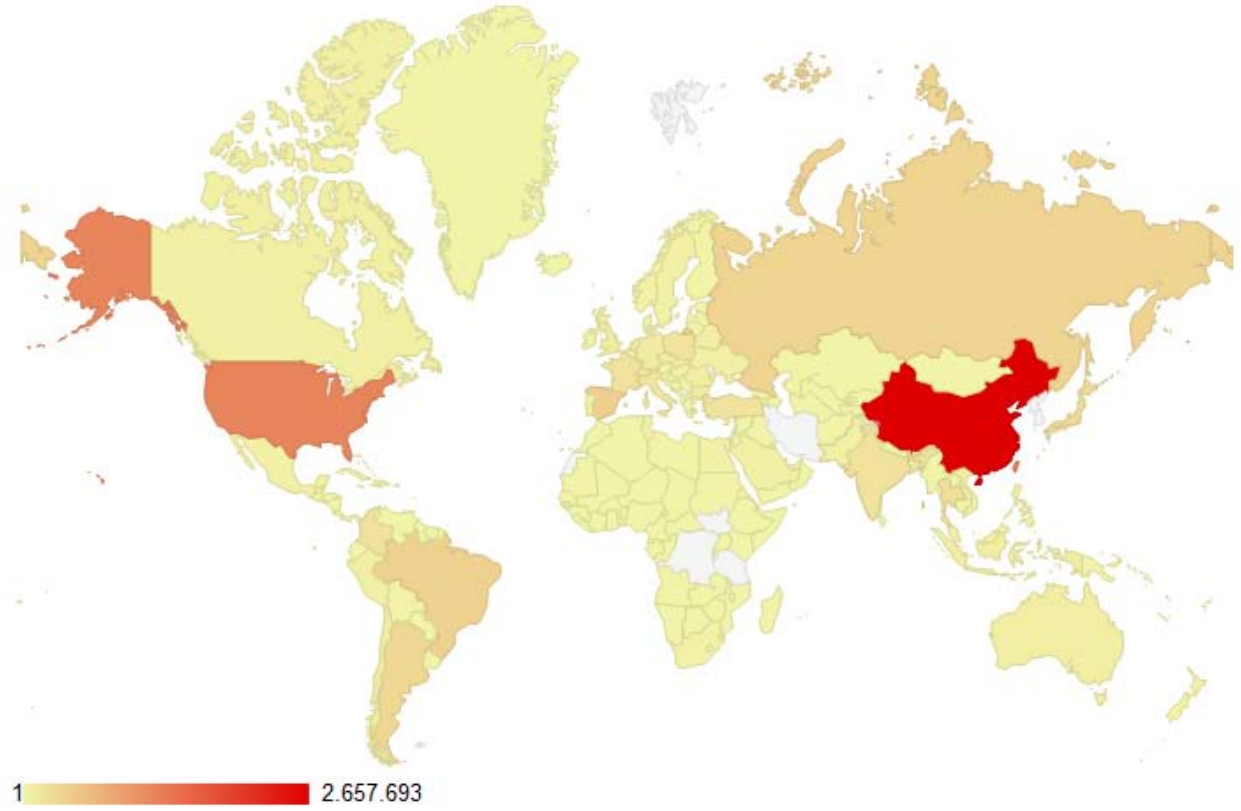

Ilustración 16. Distribución mundial de open resolvers. Fuente: DNS Amplification Attacks Observer

Específicamente, los administradores de *resolvers* DNS pueden llevar a cabo una serie de tareas para prevenir que sus sistemas sean utilizados para establecer ataques de denegación de servicio. Entre las tareas que deben contemplarse, están las medidas *anti-spoofing*, filtrado de tráfico y, técnicas como *Rate Response Limit* y configuración correcta de recursividad que son descritas en el apartado *Seguridad en Consultas y respuestas DNS.*

DENEGACIÓN DE SERVICIO DOS

Los ataques de denegación de servicio son realmente difíciles de evitar, y cualquier recurso públicamente accesible se convierte en un objeto potencial de ataque. En el caso de DNS, sus características y la debilidad intrínseca del transporte UDP en que se basa, hacen que el propio servicio sea una víctima en sí misma y no un mero elemento colaborador, como ocurre en el caso de ataques de amplificación. Dada la dificultad de localizar y bloquear un ataque sobre UDP con direcciones IP falsificadas, es importante contar con mecanismos reactivos para defenderse de un ataque de denegación de servicio cuando se es víctima final del mismo.

Genéricamente y referente a la arquitectura de red, algunos errores comunes a evitar son:

[14] Open Resolver Project : http://openresolverproject.org

- Colocar los servidores en una misma subred
- Detrás del mismo router
- Usar una única línea o ruta de provisión a internet
- Dentro de un único sistema autónomo (AS)[15]

ATAQUES SOBRE EL REGISTRO DE DOMINIOS. DNS HIJACKING

DESCRIPCIÓN

Muchos servicios de registro de dominios que operan con multitud de empresas de gran valor comercial poseen procedimientos automatizados para ofrecer una vía ágil para el control de los registros. Muchos ataques sobre los registradores parten del conocimiento y el análisis de estos procedimientos. Así por ejemplo, conociendo que el correo electrónico pueda ser el método preferido para notificar cambios de configuración, contactos, renovación del registro, etc., un atacante puede usar esta información para intentar a través de *phising* y/o ingeniería para lograr un secuestro (*hijacking*) de su dominio, redirección a una IP bajo control del atacante o cambios en la configuración del mismo.

Frecuentemente, una mala política de seguridad sobre el control o acceso a la cuenta de administración del registro de dominio, tanto por parte del registrador como por el cliente, desemboca también en el compromiso del dominio.

MEDIDAS CONTRA DNS HIJACKING O SECUESTRO DE DOMINIO

Por parte de los servicios de registro de dominios, las medidas que deben tener en cuenta para garantizar la protección de los activos de sus clientes se focalizarán en la protección contra accesos no autorizados y verificación de la autenticidad de las identidades de los registros y/o cambios solicitados. Los clientes a su vez, deben establecer con el registrador el compromiso de mantener una política de seguridad referente al control de acceso, contactos y verificación de identidades para prevenir suplantaciones derivadas de *phising* o ataques de ingeniería social.

- ***Verificación de Registro***.
  Las cuentas de registro de dominios deben contar con un método de verificación para garantizar la autenticidad del solicitante y su información de contacto, con el objetivo de reducir suplantaciones de identidad y abuso de dominio. Estudios realizados sobre *phising*[16], experiencias con botnets, y ataques de *fast-flux*[17] dejan patente la importancia sobre el control cuentas de registros de dominios en actividades cibercriminales.
  Es pues de gran importancia que el registrador implemente un mecanismo de verificación del correo de contacto y confirmación del registro a través de un hipervínculo enviada a la cuenta proporcionada.

- ***Fortificación del sistema de autenticación***.
  Implementar políticas y mecanismos de fiabilidad contrastada para la gestión, mantenimiento y transmisión de contraseñas de acceso a la cuenta de registro.

[15] AS: Autonomous System. Agrupación de redes con su propia política de rutas. http://es.wikipedia.org/wiki/Sistema_aut%C3%B3nomo
[16] Global Phishing Survey: Trends and Domain Name Use in 1H2012 http://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2012.pdf
[17] Fast-Flux. http://en.wikipedia.org/wiki/Fast_flux

Autenticación multi-factor y seguridad en la conexión (SSL, VPN) pueden considerarse medidas adicionales altamente recomendables.

- ***Multiplicidad de puntos de contacto***.
  Permitir la especificación de varios contactos para proporcionar mayor seguridad y granularidad en cuanto a la confirmación sobre las acciones a realizar.

- ***Política de renovación***. Igualmente que una verificación de los contactos e identidades de las peticiones de registro, es importante mantener una vía de comunicación en cuanto a las renovaciones y cambios que se produzcan en el dominio registrado. En ocasiones, por ejemplo una falta de comunicación y no renovar un registro que ha caducado deriva en el "robo" por parte de un tercero ajeno que retoma el registro con objeto de poder obtener beneficio. Esta técnica es conocida como ciberocupación o *cybersquatting*[18]

[18] Cibersquatting. http://en.wikipedia.org/wiki/Cybersquatting

5 FORTIFICACION DE UN SERVICIO DNS

En esta sección se describen las medidas recomendadas para el bastionado y protección de un servicio DNS de forma genérica y con aplicación específica al software DNS Bind, el más usado mundialmente y actualmente en versión 9. Para ello se agrupan en tres capas los elementos que integran el conjunto del servicio para dotar de una mayor granularidad la identificación de vectores de ataques y las medidas aplicables. Esta agrupación es la siguiente:

- **Entorno base**. Elementos base del servicio a nivel de sistemas y comunicaciones.
  - Sistema operativo
  - Software de Bind
  - Topología de Red
- **Datos**. Medidas en relación a la seguridad de los datos
  - Parametrizaciones
  - Información de registros de zona
- **Transacciones**. Protección de los mensajes en operaciones DNS
  - *Queries*. Consultas/Respuestas
  - Transferencias de Zona
  - Notificaciones
  - Actualizaciones Dinámicas

Ilustración 17. Fortificación DNS. Capas

SEGURIDAD DEL ENTORNO BASE DEL SISTEMA Y EL SOFTWARE

SISTEMA OPERATIVO

El sistema operativo del servidor debe ser estar actualizado y parcheado. Existe un amplio número de sistemas soportados por el software de BIND 9. Por tanto, cualquiera que sea la opción escogida en función de las necesidades del servicio, es importante una política de mantenimiento de parches y seguimiento de posibles vulnerabilidades que pudiesen comprometer el sistema.

Desactivar servicios innecesarios. Destinar el servidor exclusivamente al servicio DNS, desactivando todos los servicios innecesarios adicionales al software de DNS y a la administración del sistema. Aplicar las reglas de firewall estrictamente necesarias para permitir el funcionamiento de DNS.

CONFIGURACIÓN DEL SOFTWARE

- **Control y seguimiento del software**

Establecer una política de revisión del software para estar correctamente actualizado y al corriente de posibles vulnerabilidades o parches de seguridad. Se puede consultar el estado de las últimas versiones del software de BIND en el sitio Web del fabricante.

- **Ocultar la versión**

Deshabilitar directivas que puedan mostrar información sobre versión del software en ejecución. Esta información puede solicitarse con una consulta tipo TXT y clase CHAOS, como se muestra en el ejemplo siguiente:

```
 @<servidor_dns> TXT CHAOS

o bien con nslookup:

# nslookup -q=txt -class=CHAOS version.bind <servidor_dns>
```

Ejemplo 1. Averiguando versión BIND

En el software DNS BInd, la directiva que informa de la versión se especifica en el fichero de configuración named.conf. Esta información puede ser modificada, como se muestra en la siguiente configuración:

```
// Fichero: named.conf

 options { version “version no disponible”; }
```

Configuración 1. Ocultar información software BIND

- **Ejecutar el software DNS con un usuario no privilegiado .**

El servicio de DNS no debe ejecutarse nunca como root o usuario privilegiado del sistema. Esta medida, unida al "enjaulado" del servicio en un entorno chroot, evitará posicionar al atacante en una situación de control del sistema en caso de ser comprometido.

Crear un usuario específico. Generalmente se suele usar "named", bloqueando la cuenta para evitar logins como usuario named:

```
# groupadd named

# useradd -g named -d /chroot/named -s /bin/false named

# passwd -l named
```

Configuración 2. Usuario no privilegiado para correr BIND

- **Crear el entorno restringido chroot**

Crear la estructura de directorios donde se confinará el servicio por ejemplo */chroot/named*:

```
/chroot
 +-- named
    +-- dev
    +-- etc
    |  +-- namedb
    |     +-- slave  <
    |     +-- master  < Directorios donde se albergarán ficheros
       |     +--     < de zonas
    +-- var
    |   +-- run
    |
    +-- log
```

```
mkdir -p /chroot/named
cd /chroot/named
mkdir -p dev etc/namedb/slave var/run
```

Configuración 3. Chroot. Estructura de directorios chroot

A continuación copiar los ficheros necesarios para la ejecución del software de BIND:

Suponiendo que se parte de una instalación previa de bind en la ruta /var/named y /etc/named.conf, se copiarán estos en el entorno de chroot y se asignarán los permisos necesarios para las rutas donde el usuario named necesite escribir:

```
cp -a /var/named/* /chroot/named/etc/namedb/

## El fichero general de configuración es usualmente enlazado al fichero
## /etc/named.conf fuera de la jaula con objeto de darle visibilidad dentro del
## sistema y facilitar la administración
```

```
 ls –s /chroot/named/etc/named.conf /etc/named.conf

Fichero de zona horaria

 cp /etc/localtime /chroot/named/etc

# El usuario named necesitará escribir en ficheros de zona donde sea esclavo
(transferencia zona) o el PID del proceso al arracar el servicio:

 chown -R named:named /chroot/named/etc/namedb/slave
 chown named:named /chroot/named/var/run

# # Creando los ficheros necesarios de dispositivo, confirmar los major/minor
## numbers con "ls –lL /dev/random /dev/null"

 mknod /chroot/named/dev/null c 1 3
 mknod /chroot/named/dev/random c 1 8
 chmod 666 /chroot/named/dev/{null,random}
```

Configuración 4. Creando el entorno de jaula para bind

- **Asignación de permisos**

Asimismo, verificar la correcta asignación de permisos sobre los sistemas de ficheros y su contenido, evitando accesos no autorizados a configuraciones o ficheros de datos:

```
 cd /chroot/named
 chown -R named:named . # Estableciendo propietario named

 find . -type f -print | xargs chmod u=rw,og=r   # estableciendo permiso ficheros
 find . -type d -print | xargs chmod u=rwx,og=rx  # permisos directorios

 chmod o= etc/*.conf # restringiendo acceso a ficheros de configuración


## El directorio etc/namedb es donde se almacenarán los ficheros zonas. EL
## usuario named debe tener permisos para actualizar/crear nuevos ficheros


 find etc/namedb/ -type f -print | xargs chown named:named
 find etc/namedb/ -type f -print | xargs chmod ug=r,o=

 chown named:named etc/namedb/
 chmod ug=rwx,o= etc/namedb/
 chmod ug=rwx,o=rx var/run/

##logfiles
 chown named:named logs/
 chmod ug=rwx,o=rx logs/

## Protegiendo el entorno de la jaula:
 chown root /chroot
 chmod 700 /chroot
 chown named:named /chroot/named
 chmod 700 /chroot/named
```

```
cd /chroot/named
chattr +i etc etc/localtime var
```

Configuración 5. Protección y permisos de los ficheros de bind

- **Configuración de ficheros de log**

Configurar la recolección de logs a través de las directivas de logging en el fichero de configuración named.conf. Además, activar el envío a servidores remotos en la configuración de logs del sistema (por ejemplo rsyslog.conf)

Una configuración apropiada puede ser la siguiente, donde se definen dos canales: uno de ámbito general y otro específico para recoger los mensajes de seguridad y separarlos en un fichero independiente.

NOTA: Al estar el software limitado a un entorno de chroot, debe configurarse rsyslog para poder comunicarse con el entorno de la jaula. Para ello es necesario añadir un "socket" para que rsyslogd reciba mensajes desde por ejemplo "/chroot/named/dev/log". Detalles específicos de cada sistema concreto se pueden encontrar en el manual del demonio syslog del mismo sistema..

```
// Fichero: named.conf

logging {

channel default_syslog {
// Enviar la mayor parte de los mensajes a syslog ( /var/log/messages )
syslog local2;
severity debug;
};


channel audit_log {
// Enviar los mensaje de seguridad a un fichero independiente.
file "/var/named/log/named.log";
severity debug;
print-time yes;
};

category default { default_syslog; };
category general { default_syslog; };
category security { audit_log; default_syslog; };
category config { default_syslog; };
category resolver { audit_log; };
category xfer-in { audit_log; };
category xfer-out { audit_log; };
category notify { audit_log; };
category client { audit_log; };
category network { audit_log; };
category update { audit_log; };
category queries { audit_log; };
category lame-servers { audit_log; };

};
```

Configuración 6. Configuración de logging

- **Arranque del servicio en el entorno restringido**

Una vez configurado el entorno de chroot para Bind en /chroot/named, el servicio se debe arrancar tomando esa ruta como raíz:

```
named –t /chroot/named –u named –c /etc/named.conf
```

Configuración 7. Arranque de bind en un entorno de chroot

TOPOLOGÍA DE RED.

Una buena implementación de DNS debe separar siempre los servidores según su rol. Servidores autoritativos y cache recursivos serán dos componentes funcionales claramente diferenciados que requieren ser tratados de forma independiente en el diseño de la arquitectura de red.

El diseño de la arquitectura de red es siempre un punto crítico a la hora de implementar un servicio accesible públicamente. En el caso de DNS, al ser por otra parte un elemento común a la estructura interna y externa de una organización es si cabe, aún más importante.

Generalmente, las organizaciones necesitan una infraestructura DNS con dos objetivos: (1) que permita a su red interna acceder a internet, y (2) ofrecer resolución a redes externas de sus recursos públicos.

Para dotar de mayor seguridad a la infraestructura, es necesaria una segmentación de los servidores según su rol e importancia y establecer así distintas capas de exposición. La ventaja de esta implantación es la seguridad y resiliencia ante ataques. La desventaja, el costo y mayor complejidad de administración.

- **Segregación de roles servidor DNS. Autoritativo y Caché**

En una infraestructura DNS se pueden diferenciar dos roles principales de servidor *(Ilustración 18).* Servidores Autoritativos, que se encargan de mantener y distribuir los dominios de nombres y servidores caché recursivos que se encargan solicitar y almacenar temporalmente las resoluciones de dominios que obtienen desde los servidores autoritativos. A su vez, los servidores Autoritativos se dividen en:

Primarios o Máster. Mantiene y administra su base de datos local con los dominios y registros de los que es propietario

Secundario o Esclavo. No tiene base de datos local, sino que obtiene una réplica desde un servidor Master a través de una transferencia de zona

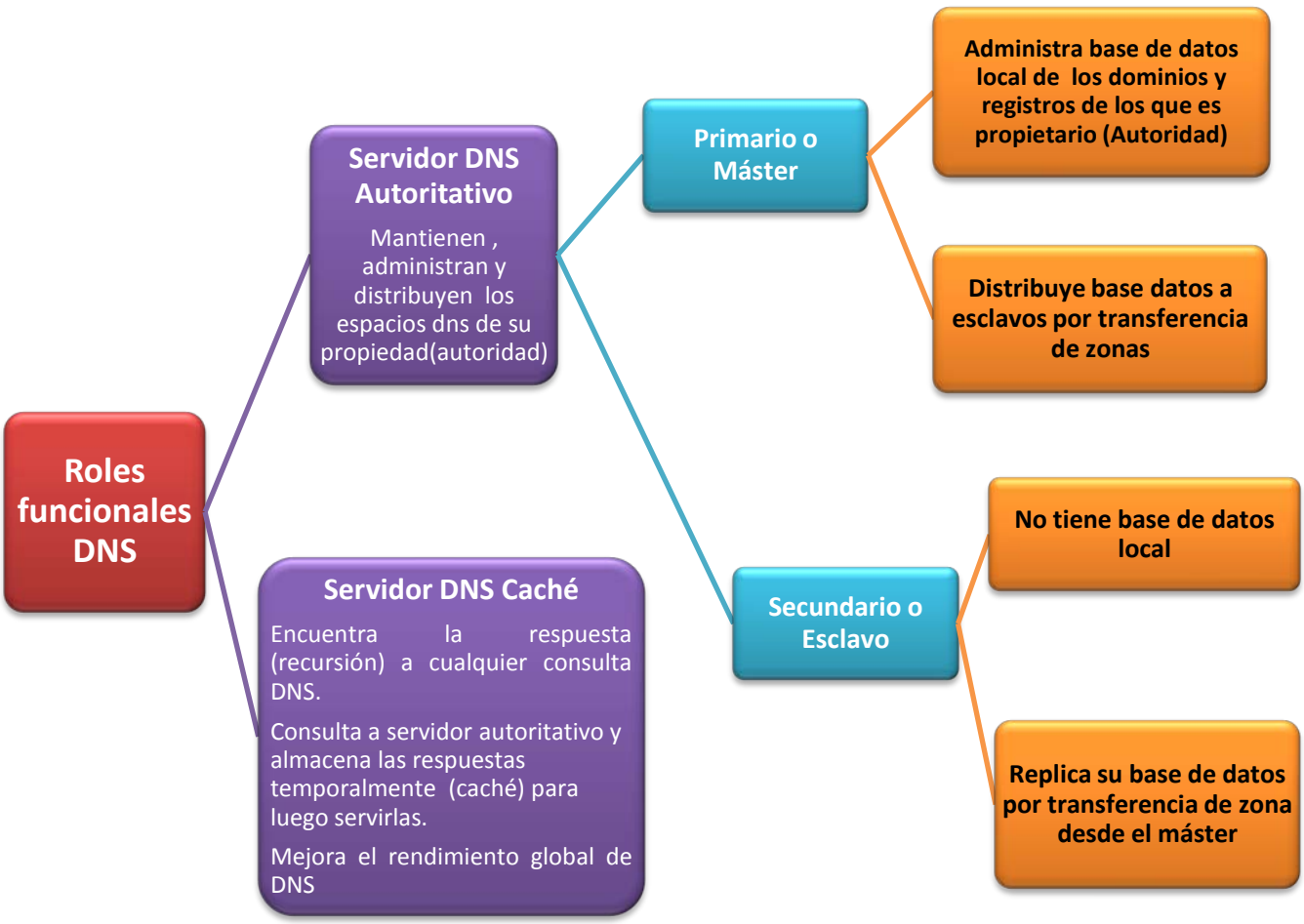

Ilustración 18. Roles funcionales de Servidor DNS. Servidores caché y autoritativos

- **Diseño de la arquitectura de red**

Implementación de Servidores DNS autoritativos.

A la hora del diseño de la arquitectura de red en un servicio DNS hay que contemplar claramente una separación de funciones dependiendo de la información que suministre el servidor autoritativo, con el objeto de separar la información pública de la privada. Además, debe proporcionarse redundancia para afrontar posibles cortes de comunicación ,bien por causas intencionadas o por incidentes técnicos. Estos requisitos pueden cubrirse separando servidores internos y externos dotando a ambos de redundancia geográfica. En ningún caso un servidor Autoritativo permitirá recursión, papel destinado a servidores cachés.

Siguiendo esta estrategia, los servidores autoritativos deben separarse entonces, en dos grupos diferenciados para servir recursos públicos y recursos internos privados:

Servidor Autoritativo Público. Responderá a solicitudes DNS desde una red externa, ofreciendo los registros públicos del dominio de la organización. Puede situarse en una red pública aislada para mayor seguridad, pero generalmente se sitúa en la red DMZ, es decir, el segmento de red que es frontera entre la red interna e internet.

Servidor Autoritativo Privado. Responderá a únicamente a solicitudes procedente de la misma red interna. Situado en la red interna.

Adicionalmente, puede contemplarse la instalación de un servidor máster autoritativo "oculto" , también denominado *Hidden Máster o Stealth.* Un servidor DNS oculto es un servidor primario que no aparece en los registros NS de zona, aunque es máster de la misma. El servidor oculto se sitúa en la red interna tras un firewall y no puede ser alcanzado desde redes externas de internet y tampoco se usa para servir información. Su única función es la de mantener las zonas y transferir zonas a los servidores secundarios.

Esta separación recomendada puede obtenerse por dos vías: bien físicamente, destinando hosts exclusivos para la parte pública y privada o, de forma altenativa usando un host único conocido como Split-DNS, que implementa la separación internamente con el uso la funcionalidad de "vistas" del software DNS. El uso Split-DNS, tiene la ventaja de necesitar menos recursos físicos (hosts) para lograr la separación de zonas, pero cuenta con la contrapartida de que, en el caso de compromiso del servidor, la información sobre redes internas estará expuesta.

Implementación de Servidores caché recursivos.

En la DMZ se situarán los servidores dedicados recursivos. Estos serán los únicos servidores que puedan realizar recursión para solicitudes procedentes de *resolvers* de la red interna y así garantizar la resolución de cualquier recurso de internet a la misma. Jamás deben permitir consultas recursivas procedentes del exterior. Estos servidores no deben usarse como *resolvers* directamente sino que deben ser accedidos a través de un *forwarder*.

Los servidores recursivos internos generalmente se configurarán como redirigir (*forwarder*) las consultas que no conocen hacia los servidores recursivos de la DMZ (como un proxy DNS). Por ejemplo, cuando un servidor recursivo interno no conoce la resolución de un domino externo de internet, dirigirá la consulta al servidor recursivo localizado en la DMZ el cual se encargará de hacer la recursión y devolver la respuesta al servidor interno.

Redundancia y alta disponibilidad.

Para reducir la posibilidad de una pérdida del servicio se proveerá de redundancia geográfica y de red a los servidores autoritativos. Esto supone que estarán en subredes independientes tras routers distintos y físicamente en lugares distintos. Adicionalmente, se puede disponer de un servidor DNS master autoritativo oculto (*hidden master*), y sólo serán visibles masters secundarios como servidores de nombres. Estos tomarán todas sus zonas desde el servidor máster oculto a través de transferencias de zona o actualizaciones dinámicas.

La estrategia de arquitectura de red propuesta puede visualizarse en la *Ilustración 19. Arquitectura de red. Implementación de una infraestructura DNS :*

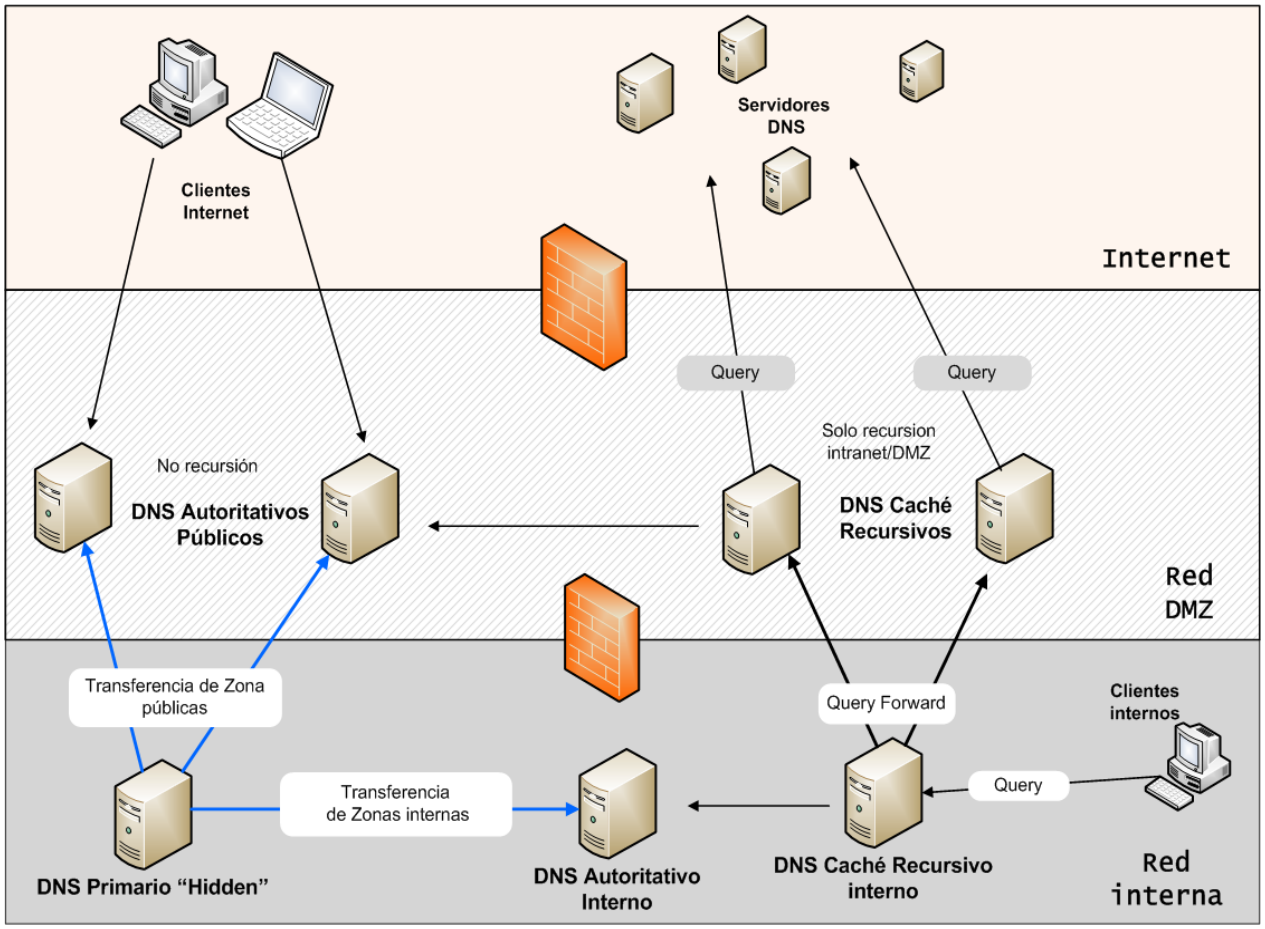

Ilustración 19. Arquitectura de red. Implementación de una infraestructura DNS

MONITORIZACIÓN INTERNA

En un sistema de cierta importancia y/o criticidad es muy recomendable contar con un sistema de monitorización para detectar posibles ataques o eventos que sucedan en la red interna. Con ese fin, se recomienda implementar mecanismos de detección de intrusión (IPS/IDS), monitorización de accesos, concentradores de logs/eventos o SIEM (*Security Information and Event Management*) etc.

RESUMEN DE MEDIDAS EN EL ENTORNO BASE DEL SERVICIO DNS

A continuación se muestra en la *Tabla 5* resumen de las medidas de seguridad en el entorno base del sistema DNS:

Seguridad en el Entorno base y el software DNS	
Sistema Operativo	
1	Parcheado del SO
2	Deshabilitar Servicios innecesarios
Software DNS	
3	Software actualizado y parcheado
4	Ocultar información de versionado
5	Ejecutar como usuario no privilegiado
6	Aislar el entorno del software (chroot)

7	Configuración logging
Topología de Red	
8	Separar Roles en distintos servidores. Autoritativos y Cachés
9	Redundancia geográfica y de red de servidores autoritativos
10	En caso de uso de split DNS, configurar mínimo vista externa e interna
11	Si se usa un servidor master oculto (interno), restringir las transferencias a servidores internos
Monitorización	
12	Contemplar la implementación de sistemas de prevención/detección de intrusiones

Tabla 5. Checklist de medidas de seguridad en el entorno del servidor DNS

MEDIDAS DE SEGURIDAD EN LAS TRANSACCIONES.

SEGURIDAD EN CONSULTAS Y RESPUESTAS DNS.

Las amenazas principales que afectan a las *queries* DNS son las relacionadas con el *spoofing*, las cuales se pueden materializar en **ataques de *DNS cache poisoning***, **denegaciones de servicio** así como **ataques de amplificación DNS**. Adicionalmente y sacando provecho de la debilidad propia del protocolo DNS, es posible la manipulación de respuestas capturadas en un entorno local a través de *sniffing* de tráfico de red.

- **Limitar recursión. Segmentación red / vistas**

Es igualmente importante limitar la recursión a los clientes o redes que formen parte de la organización. Por ejemplo, grandes compañías como los proveedores de Internet o ISP deberían ofrecer sólo recursión a los clientes a los que proporcionan acceso. Este objetivo puede lograrse de distintas formas dependiendo de la topología escogida (ver *Topología de Red.*).

En el caso de BIND, el uso de las vistas proporciona un método para segmentar y separar distintos orígenes a los cuales proporcionar las capacidades de consulta recomendadas. De este modo podemos, por ejemplo, **en un servidor Split-DNS** (que da servicio tanto a zonas internas como clientes externos) permitir la recursión a consultas procedentes de zonas internas y denegarla a clientes externos**. *En un servidor DNS autoritativo la recursión siempre debe ser deshabilitada.***

Un ejemplo de uso de vistas para separar zonas internas de externas puede ser la siguiente:

```
// Ejemplo de configuración “split” con vistas
// Situamos en la acl “trusted” las redes internas (10.2.0.0/24) a las que ofrecer
// recursion
// prevenimos que hosts externos usen este servidor como resolver para otros
// dominios ajenos

acl "trusted" {
// Nuestra red interna
10.0.2.0/24;
localhost;
};
```

```
view "internal-in" {
  // Vista interna. Permitimos a redes internas consultas recursivas y acceso a
  // la cache.
  // IMPORTANTE: Las vistas configuradas se aplican en orden de aparición en la
  // configuración

  match-clients { trusted; };
  recursion yes;
  additional-from-auth yes;
  additional-from-cache yes;
zone "." {
  // Link in the root server hint file.
  type hint;
  file "db.cache";
};
zone "0.0.127.in-addr.arpa" {
  type master;
  file "master/db.127.0.0";
  allow-query { any; };
  allow-transfer { none; };
};
zone "localhost" {
  type master;
  file "db.localhost";

};
zone "interna.ejemplo.com" {
  // Ejemplo de zona interna.
  type master;
  file “etc/interna.ejemplo.com”;

};
};

// Vista para clientes DNS externos (no pertenecen a la acl “trusted”)

view "external-in" {
  // Vista Externa. Permitimos a cualquier cliente.
  // No se permite recursión/cache, evitamos ser open resolver

  // IMPORTANTE: Las vistas configuradas se aplican en orden de aparición en la
  // configuración
  match-clients { any; };
  recursion no;
  additional-from-auth no;
  additional-from-cache no;
// Link in our zones

zone "ejemplo.com" {
  type master;
  file “etc/zona_master.ejemplo.com”;
  allow-query { any; };
};
};
};
```

Configuración 8. Limitación de recursión y zonas. Uso de vistas

- **Defensa contra el IP Spoofing. Filtrado de tráfico**

Dado que incluso los servidores correctamente configurados pueden ser explotados por atacantes que usan una IP origen falsificada para ejecutar sus consultas DNS, se recomienda aplicar las líneas base descritas en las guías *Best Current Practices BCP38* y *BCP84* publicadas por la Internet Engineering Task Force (IETF) para identificar y filtrar tráfico sospechoso de falsear direcciones IP.

Estas guías, altamente recomendables, son:

> **BCP38:** *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing* http://tools.ietf.org/html/bcp38
> **BCP84:** *Ingress Filtering for Multihomed Networks* http://tools.ietf.org/html/bcp84

- **Mejoras en los servidores autoritativos. Response Rate Limiting**

Los servidores autoritativos deben ser accesibles con objeto de ofrecer la información necesaria de los registros de los que son responsables. Es fundamental comprobar que **los servidores autoritativos rechazan siempre consultas recursivas** y sólo ofrecen resolución para los registros de su dominio. Adicionalmente, para combatir los ataques de amplificación, es recomendable implementar en los servidores autoritativos la solución *Response Rate Limiting* (RRL), sobre todo si no se adoptan mecanismos adicionales (BCP38) para detectar posibles falseos de direcciones IP. Así, ante la imposibilidad de determinar si una consulta DNS UDP tiene una dirección IP falseada, se tendrán en cuenta patrones de consultas y respuestas y de este modo, tratar de inferir cuándo se está produciendo un ataque. Esta información puede usarse para descartar las respuestas generadas por consultas sospechosas. Las consultas, a diferencia de las respuestas, no se verán afectadas por el mecanismo RRL.

Escenario ejemplo:

Supóngase que en el servidor autoritativo del dominio *ejemplo.com* se están realizando una gran cantidad de consultas desde una IP 1.2.3.4 con el *flag* de DNSSEC activo, de modo que el tamaño de la respuesta será grande. Este comportamiento es típico de "*flooding*" de peticiones en un ataque de amplificación DNS y hacen sospechar de un ataque dirigido contra la IP 1.2.3.4:

```
13-13-2013    12:27:34.102    queries:    info:    client    1.2.3.4#58540
(host1.example.com): query: testhost.example.com IN A +ED (1.2.3.4)
13-13-2013    12:27:41.606    queries:    info:    client    1.2.3.4#55979
(host1.example.com): query: testhost.example.com IN A +ED (1.2.3.4)
13-13-2013    12:27:59.196    queries:    info:    client    1.2.3.4#47516
(host1.example.com): query: testhost.example.com IN A +ED (1.2.3.4)
```

Ilustración 20. Logs de BIND. Flooding de consultas en un ataque de amplificación DNS

Desde la versión 9.9.4 de BIND se incorpora la funcionalidad RRL y, para su implementación, bastará con especificar con la cláusula ***rate-limit*** los parámetros deseados en la sección *options* o *view* de *named.conf*:

```
//named.conf . Cláusula de rate-limit para combatir ataques de amplificación DNS
// se limitarán las consultas detectadas como sospechosas a 5 por segundo

options {
```

```
  directory "/var/named";
  rate-limit {
    responses-per-second 5;
    // TEST: Para comprobar el funcionamiento se puede descomentar
    // línea siguiente
   // log-only yes;
  };
};
```

Configuración 9. BIND, configuración de Response Rate Limiting contra ataques flooding

Una vez detectados los ataques, las consultas afectadas se reflejarán en los logs, tanto si está activo como en modo *log-only*:

```
13-Dec-2013 12:41:42.336 queries: info: client 1.2.3.4#53459
(host1.example.com): query: host1.example.com IN A +ED (1.2.3.4)
13-Dec-2013 12:41:42.336 query-errors: info: client 1.2.3.4#53459
(host1.example.com): would rate limit slip response to 1.2.3.0/24 for
testhost.example.com IN A (3ee9836b)
```

Ilustración 21. Logs de BIND. Detección de flooding. Rate-limit en log-only

```
13-Dec-2013 12:44:44.868 queries: info: client 1.2.3.4#57114
(host1.example.com): query: host1.example.com IN A +ED (1.2.3.4)
13-Dec-2013 12:44:44.869 query-errors: info: client 1.2.3.4#57114
(host1.example.com): rate limit drop response to 1.2.3.0/24 for
host1.example.com IN A (3ee9836b)
```

Ilustración 22. Logs de BIND. Detección de flooding. Rate-limit activo y descarte de respuestas

Dado que RRL es flexible para combatir distintos escenarios de ataque, es altamente recomendable consultar en profundidad la Guía de Referencia del Administrador de BIND 9.9.4[19]

Finalmente, ha de tenerse en cuenta la circunstancia de que la implementación de RRL puede favorecer en cierto modo los ataques de *DNS caché poisoning* dado que, se pueden estar denegando o poniendo en espera respuestas legítimas, lo que puede brindar a un atacante más oportunidades para conseguir inyectar una respuesta. Como ya se ha discutido en el apartado Vectores de ataque y amenazas en un escenario DNS, la única solución fiable contra ataques de spoofing es la implementación de DNSSEC.

SEGURIDAD EN TRANSACCIONES DE TRANSFERENCIAS DE ZONA

Las transacciones de transferencia de zona se efectúan sobre protocolo TCP y, dadas sus características de conexión, son más difíciles de manipular. Sin embargo aún existe, sobre todo en ataques en red local, la posibilidad de modificar la transacción a través de ARP *spoofing* y ataques *Man in the Middle*.

- **Uso de ACLs y filtrado IP**

BIND posibilita restringir las IP autorizadas a solicitar una transferencia de zona con las directivas *allow-transfer*, pero este método no es eficaz en un ataque de *spoofing* bien elaborado. Puede contemplarse como una medida de fortificación, pero no como una solución.

[19] BIND9 Adminstrator Reference Manual (ARM) https://kb.isc.org/category/116/0/10/Software-Products/BIND9/Documentation/

```
//named.conf
options {
// permite transferencia de zona solo hacia 10.0.2.15
  allow-transfer { 10.0.2.15; };
}
```

Configuración 10. Filtrado IP para transferencia de zona

- **TSIG (Transaction SIGnature)**

TSIG es el método recomendado para la protección de transacciones de transferencia de zona. Con este método la comunicación entre servidores es autenticada usando una clave compartida entre ellos. Originalmente descrito para la protección de actualizaciones dinámicas en el RFC2845, es usado también para evitar la manipulación de transferencias de zona. Una descripción de configuración TSIG puede consultarse en.*Anexo 7.1 TRANSACTION SIGNATURE. TSIG.*

```
zone "ejemplo.com" {
   Type master;
   file "etc/zona_master.ejemplo.com"
   allow-transfer { key "ejemplo.com";}; //transferencia de zona solo permitida
}                                        // con la clave ejemplo.com
```

Configuración 11. TSIG para transferencias de zona

SEGURIDAD EN NOTIFICACIONES

Las amenazas de *spoofing* sobre transacciones de notificación (las que envía el servidor autoritativo a los esclavos cuando se ha producido un cambio en las zonas), son las mismas que para una transferencia de zona. No obstante, el impacto es menor en cuanto la transacción en sí no lleva información sensible. El efecto que en un servidor esclavo podrían causar notificaciones espurias sería la de conseguir que solicitase una transferencia de zona y, en caso extremo, ser víctima de una denegación de servicio si el volumen de notificaciones es muy alto.

- **ACLs y filtrado IP**

A diferencia del caso de transferencias de zona, en el caso de notificaciones y dado que el impacto que esta transacción es menor, el filtrado IP con las correspondientes directivas *allow-notify* puede ser aceptable. En un servidor esclavo se permitirán notificaciones del servidor master especificado con el estamento masters en el estamento de zona. Adicionalmente, pueden permitirse notificaciones de otros servidores con la cláusula *allow-notify*, especificando la IP permitida para la notificación:

```
zone "ejemplo.com" in {
   type slave;
   file "etc/zona.ejemplo.com"
   masters {10.0.2.2}; //servidor maestro, notificaciones permitidas
   allow-notify {10.0.2.3}; //se admiten notificaciones también de 10.0.2.3
};
```

Configuración 12. Filtrado IP para recibir notificaciones

- **Uso de TSIG en notificaciones.**

De igual modo que en transferencias de zona, es la solución eficaz para la protección de la transacción, ya que como se ha reiterado, el falseado de IP (*spoofing*) es una vulnerabilidad

fácilmente explotable. Una posible configuración sería usar ACLs o bien forzar el uso de TSIG con un estamento *server*:

```
server 10.0.2.5 {
  keys { ejemplo.com; };
};
```

Configuración 13. Uso de TSIG en transacciones servidor-servidor

De esta forma, las transacciones, consultas, notificaciones, transferencias de zona y actualizaciones dinámicas dirigidas al servidor 10.0.2.5 serán firmadas con la clave compartida por ambos. Lógicamente una cláusula similar se configurará en el servidor 10.0.2.5 con la IP del servidor recíproco.

SEGURIDAD EN ACTUALIZACIONES DINÁMICAS

Estas transacciones son generalmente utilizadas en entornos que requieren una actualización de las zonas con bastante frecuencia y volumen, lo que dificulta la administración manual de los ficheros de zona. Al tratarse de una tarea administrativa que directamente incide en el contenido de las zonas, es una transacción que es necesario proteger de manipulaciones malintencionadas. De hecho el RFC2845[20] sobre TSIG, que ya se mencionó anteriormente, fue especificado inicialmente con miras a dotar de un mecanismo de autenticación las transacciones de actualizaciones dinámicas de DNS.

- **TSIG para actualizaciones dinámicas**

Como ya se ha visto, del mismo modo que en notificaciones y transferencias de zona, es posible forzar el uso de TSIG en transacciones de actualizaciones dinámicas. En TSIG se hace uso de una clave compartida entre los servidores para autenticar la comunicación, añadiendo una firma generada con misma. La clave se aloja en un fichero en los servidores intervinientes en la transacción y debe ser protegida contra accesos no autorizados. Esto es importante porque la firma autentica la transacción pero no la autenticidad de los datos origen por lo que un host comprometido supone una amenaza para los servidores que administra. Ver anexo *TRANSACTION SIGNATURE. TSIG.*

La configuración necesaria para esta opción se muestra en este ejemplo:

```
zone "ejemplo.com" in {
   Type master;
   file "etc/zona_master.ejemplo.com"
   allow-update { key "ejemplo.com";}; //actualizaciones con clave TSIG
};
```

Configuración 14. Actualizaciones dinámicas con TSIG

- **Seguridad del canal de comunicación.**

El cifrado de la comunicación a través de un canal dedicado o una red aislada o VLAN, o incluso un túnel IPSEC pueden ser medidas complementaria o alternativas a TSIG. En entornos reducidos, si

[20] RFC2845. *Secret Key Transaction Authentication for DNS (TSIG)*. https://www.ietf.org/rfc/rfc2845.txt

una gestión manual es asumible, se sugiere cambiar a esta opción y reducir así este vector de ataque.

RESUMEN DE MEDIDAS EN LA PROTECCIÓN DE LAS TRANSACCIONES

Seguridad las transacciones DNS	
Seguridad en consultas y respuestas DNS	
1	Limitar recursión. Segmentación red / vistas
2	Defensa contra IP spoofing. Filtrado de tráfico
3	Mejoras en servidores autoritativos. Response Rate Limiting
Seguridad en Transferencias de zona	
4	Uso de ACLs y filtrado IP
5	TSIG (Transaction SIGnature)
Seguridad en Notificaciones	
6	Uso de ACLs y filtrado IP
7	TSIG (Transaction SIGnature)
Seguridad en Actualizaciones Dinámicas	
8	TSIG (Transaction SIGnature)
9	Seguridad del canal de comunicación. Redes de administración,VLANs

Tabla 6. Checklist de medidas de seguridad en las transacciones DNS

MEDIDAS DE SEGURDAD EN LA PROTECCIÓN DE LOS DATOS.

En relación a la capa de datos, los objetivos se centran en la protección y disponibilidad de la información de las zonas. Para ello se ha de tener en cuenta parametrizaciones y contenidos en los registros. Es necesario configurar convenientemente las zonas y la información que pueda proporcionar con objeto de evitar fugas de información sobre elementos no deseados como puedan ser, por ejemplo, datos de red interna.

FICHEROS DE ZONA. PARAMETRIZACIÓN EN REGISTROS SOA

El registro SOA de una zona establece los parámetros globales para la misma. Para optimizar la latencia y distribución de los registros de zona y sus actualizaciones, los parámetros que afectan al registro SOA deben ser escogidos cuidadosamente para regular la comunicación entre los servidores primarios y secundarios. Además del parámetro TTL que se asigna al registro (valor recomendado entre 2 y 7 días), se asocian 5 parámetros más. Estos parámetros, recogidos en el RFC1912 junto con los valores recomendados, son:

- **`Serial`** (numero): Este valor en el campo RDATA del registro SOA se usa para indicar cambios de zona. Debe ser incrementado siempre que se realice cualquier modificación en los datos de zona.
- **`Refresh`** (segundos): Comunica a servidores secundarios cuántos segundos se debe esperar entre transferencias de zona. Si se trata de zonas actualizadas frecuentemente, se recomienda un periodo de 20 minutos a 2 horas. En caso de ser infrecuentes las actualizaciones, se aconseja fijar el intervalo entre 2 y 12 horas. En caso de uso de

DNSSEC, el valor siempre será menor que el periodo de validez del registro firmado de zona RRSIG[21]. Si el servidor primario envía un mensaje NOTIFY, la transferencia en los secundarios se hará de inmediato, sin esperar a que se consuma el *Refresh Value.*

- **`Retry`** (segundos): Es el tiempo que un servidor secundario debe esperar para reintentar una transferencia de zona que ha fallado previamente. Debe ser una fracción del Refresh Value especificado, siendo una estimación valores entre 5 minutos y 1 hora.
- **`Expire`**: Tiempo durante el cual un servidor secundario debe considerar válidos los registros de la zona si no se ha podido realizar un refresco de la misma. Se debe fijar a un múltiplo del valor Refresh, preferiblemente entre 2 y 4 semanas.
- **`TTL Mínimo`**: Valor en segundos que un servidor secundario debe guardar en caché un resultado negativo (NXDOMAIN) de un registro. Dependiendo de la frecuencia de actualizaciones de una zona, se recomienda fijar entre 30 minutos (zonas dinámicas) a 5 días para zonas estáticas o de infrecuente actualización. Un valor de 5 minutos se puede adoptar como valor umbral mínimo.
  A continuación se muestra un ejemplo de registro SOA:

```
$TTL 3d ; 3 días TTL
@        IN   SOA   ejemplo.com. root.ejemplo.com. (
                    199609203 ; Serial
                    28800  ; Refresh 8 horas
                    3600  ; Retry  1 hora
                    604800 ; Expire  1 semana
                    86400) ; Minimum TTL 1 dia
```

Configuración 15. Configuración de registros SOA

RESTRINGIR INFORMACIÓN PROPORCIONADA POR TIPOS DE REGISTROS

Entre los tipos de registros DNS existentes, los más habituales para mostrar información están: registros TXT, que almacenan un texto destinado a dar información a personas y aplicaciones sobre redes, hosts, servicios, u otro tipo de información genérica. Los registros HINFO, recogen información sobre el host que alberga el servicio DNS y los registros LOC, sobre la localización geográfica del servidor. El administrador del sistema debe asegurarse de que estos registros no ofrecen información sensible que pueda ser aprovechada por un atacante para reconocer características del entorno o cualquier otro dato que le pueda revelar posibles vectores de ataque.

En la siguiente configuración para Bind, se muestra como ocultar información sensible, en este caso la versión del software.

```
// named.conf
options {
version “Not disclosed”; // Ocultar información de versiones
}
```

Configuración 16. Ocultar versión bind

Como resultado de aplicar la anterior configuración se comprueba que ya no se muestra la versión de Bind:

[21] RRSIG. Tipo de registro DNS utilizado en DNSSEC y que contiene la firma de un conjunto de registros.

```
kuko@DNS:~$dig @ucdns.sis.ucm.es -c CHAOS -t TXT version.bind

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ucdns.sis.ucm.es -c CHAOS -t TXT version.bind
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59251
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "Not disclosed"

;; AUTHORITY SECTION:
version.bind.           0       CH      NS      version.bind.

;; Query time: 51 msec
;; SERVER: 147.96.2.4#53(147.96.2.4)
;; WHEN: Wed Dec 11 17:10:02 2013
;; MSG SIZE  rcvd: 70
```

Ilustración 23. Versión de bind oculta

RESUMEN DE LAS MEDIDAS EN LA PROTECCIÓN DE LOS DATOS

Seguridad en la protección de datos DNS	
Ficheros de zona. Parametrización en registros SOA	
1	Valores recomendados: TTL : 2-7 días Serial: actualizar con cada cambio en los ficheros de zona Refresh: 2-12h ó 20min-2h (actualizaciones frecuentes) Retry: 5min-1h Expire: 2-4 semanas TTL min: 30min-5h (valor umbral 5min)
Ocultar información	
4	Ocultar info TXT y versión bind

Tabla 7. Checklist de Bastionado. Medidas de Seguridad de los datos

6 DNSSEC

QUÉ ES Y CÓMO FUNCIONA

DSNSEC, siglas en inglés de *Domain Name System Security Extensions,* es considerado un mecanismo eficaz para evitar el *spoofing* y manipulación de mensajes en el protocolo DNS, y por extensión, proporcionar una vía de protección contra ataques de caché *poisoning* y similares. DNSSEC se basa en una infraestructura de criptografía de clave pública PKI[22] y *en el uso de firmas digitales* para establecer autenticidad de las fuentes y la validez de los mensajes. Aplicado a las *queries* DNS, asegura la integridad de los mensajes y la autenticidad de la fuente emisora.

DNSSEC, con el uso de una PKI y un conjunto especial de registros de recursos RRs, registros específicos de firma (RRSIGs) y registros de clave DNSKEY, permite a *resolvers* con capacidades DNSSEC comprobar lo siguiente:

- **Autenticidad de origen**: Autenticar que los datos recibidos sólo pueden proceder de la zona solicitada
- **Integridad:** Verificar la integridad de los datos, es decir, que los datos no han sido modificados en el transcurso de la transacción.
- **No existencia:** Verificar, en el caso de una respuesta de dominio no existente (NXDOMAIN), que, efectivamente el registro no existe en la zona solicitada y no ha sido expresamente eliminado en la intercepción de la transacción.

En DNSSEC se valida la clave pública de una fuente con una cadena de verificaciones que empieza en un servidor de confianza (como un *root server*) y bajando por la jerarquía del espacio de nombres DNS verificando sucesivamente la firma de la clave publica de un nodo hijo por su nodo padre. La clave pública de servidores de confianza se denomina "*trust anchor*".

Una vez realizada esta verificación de clave pública de la fuente, el siguiente paso en DNSSEC es autenticar la respuesta. En este caso, las respuestas incluyen no sólo los registros solicitados, sino además, la firma digital de un conjunto de registros encapsulada en un tipo de registro específico, denominado RRSIG. Entonces, el *resolver*, usando la clave pública verificada anteriormente, comprueba la validez de la firma y se asegura de que la respuesta es auténtica. En el caso de una respuesta negativa indicando la no existencia de un registro, se adjunta un registro específico denominado NSEC con su firma correspondiente, cuya verificación asegura la validez de la respuesta y que el registro no ha sido eliminado por una manipulación intermedia.

COMPONENTES Y OPERACIONES

En DNSSEC existen dos procesos principales: firmar y servir, y verificar firma. Estos procesos, se realizan a través de mecanismos basados en criptografía de clave pública. Aunque operativamente no es necesario más que un par de claves pública/privada, es muy común utilizar al menos dos pares para facilitar las tareas de renovación de claves y de re-firmado de zonas. Además, la

[22] PKI. Public Key Infraestructure. Es un conjunto de componentes para establecer comunicaciones cifradas basadas en criptografía asimétrica de clave pública. Muy utilizado para autenticación, cifrado , firmado digital y otros usos donde se establecen relaciones de confianza en certificados digitales.

"separación de claves" es una buena práctica criptográfica, para limitar el alcance de un posible compromiso. De este modo, DNSSEC cuenta con dos pares de claves publica/privada. Un par denominado *Key Signing Key* (KSK) para el firmado de registros de clave DNSKEY y otro, denominado *Zone Signing Key* (ZSK) para el firmado de registros (RRsets) .

- **Registros DNSSEC**

DNSSEC dispone de los siguientes registros específicos para su funcionamiento:

RRSIG (*Resource Record Signature*). Registro de firma. Contiene la información de un conjunto de registros DNS del mismo tipo y la firma del mismo (creada con la clave privada).

DNSKEY. Firma pública. Es usada para verificar las firmas adjuntas en los registros RRSIG.

NSEC (*Next Secure*). Cuando se solicita un registro no existente, se devuelve este tipo de registro y su correspondiente firma (RRSIG) para demostrar al *resolver* la no existencia del mismo. Contiene el listado en orden canónico del siguiente dominio autoritativo o punto delegado (NS) y el conjunto tipos de registros presentes en los mismos. Se acompaña del RRSIG con la firma del mismo. Junto a su registro de firma (RRSIG), constituyen el método de verificar. Ha sido sustituido por la versión NSEC3, puesto que NSEC por su comportamiento posibilita obtener información de zonas (enumeración) solicitando registros inexistentes.

DS. (*Delegation Signer*). Contiene un hash de la clave pública de un nodo hijo. Para tener la certeza de que, en una zona, la clave pública (DNSKEY) y los registros de firma que la misma verifica no han sido ambos manipulados, se genera un hash de la clave pública que se entrega al nodo inmediatamente superior. Este nodo genera el registro DS almacenando ese hash y lo firma con su clave privada, obteniendo el registro RRSIG correspondiente. Esta cadena continúa hacia arriba en la jerarquía hasta llegar al último nodo de confianza en la cadena, típicamente el nodo raíz.

En la *Ilustración 24* se muestra un ejemplo de registro RRSIG y se identifican los campos integrantes del mismo:

Campo	Descripción	Longitud (bytes)
NAME	Nombre del dominio al que pertenece el registro	Cadena variable
TYPE	Código del tipo de registro	2 bytes
CLASS	Código de clase del registro	2 bytes
TTL	Tiempo en segundos durante el cual el registro es cacheado	4 bytes
RDLENGTH	Indica la longitud en bytes del campo RDATA	4 bytes
RDATA	Cadena de longitud variable que describe el registro de acuerdo al tipo y clase del mismo	Cadena variable

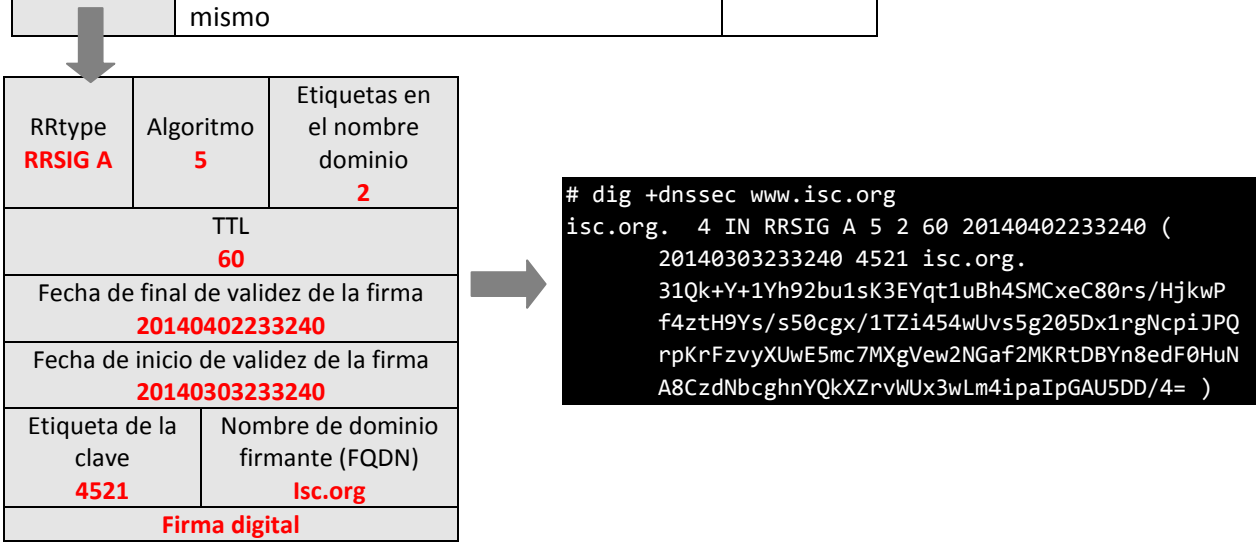

Ilustración 24 Identificación de campos en registro DNSSEC

- **Operaciones DNSSEC**

  Firmado y Servicio de conjuntos de registros (RRsets). El firmado de registros se realiza por conjuntos de registros (RRSet) con el mismo nombre de dominio, clase y tipo. Por ejemplo, conjuntos de tipo A, tipo NS, tipos DNSKEY o DS (específicos de DNSSEC), etc. Recuérdese el formato de registro descrito en *Tabla 1. Formato de registro. Resource Record* **(RR).** El registro RRSIG es el más relevante, pues almacena la firma digital y la información asociada (ID de la clave usada, fechas de inicio y expiración de la firma, etc) para cada grupo de registros o RRset. La firma se realiza con la clave privada correspondiente del par de claves pública/privada generada para el firmado de registros (ZSK) o de claves (KSK).

Verificación de firmas. Un *resolver* puede verificar las firmas digitales contenidas en los registros de firma RRSIG haciendo uso la clave pública aportada en los registros DNSKEY. Igualmente, el registro DNSKEY tiene su correspondiente RRSIG firmado. Para la verificación de las claves públicas en sí, se parte del *trust anchor*, clave pública de confianza del nodo más alto en la jerarquía (que tiene el instalada el resolver) y que, óptimamente, sería la de un nodo raíz para un dominio considerado "globalmente seguro". La cadena de confianza se va

construyendo, verificando sucesivamente las claves públicas de nodos hijo, cuyos hashes se encuentran en los registros DS de los nodos padre, debidamente firmados. El recorrido de la cadena de confianza se denomina “isla de seguridad” (*Ilustración 25*)

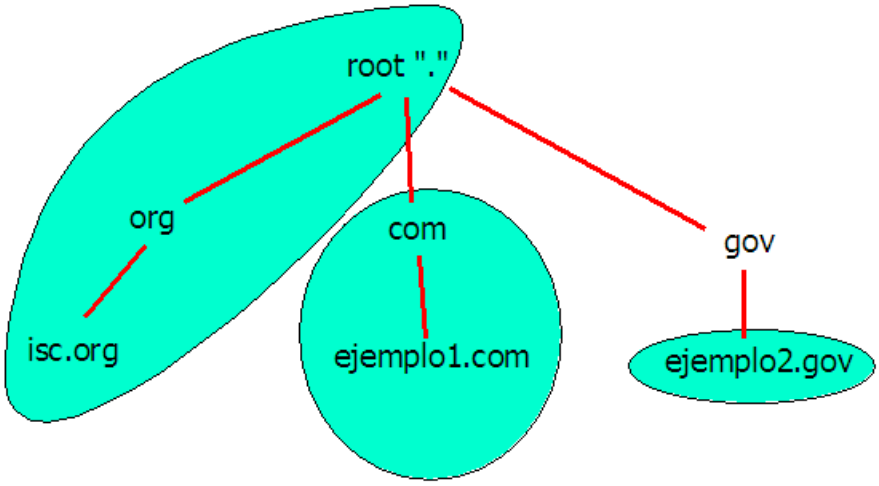

Ilustración 25. Islas de seguridad

En la siguiente tabla se muestra el efecto de la trust anchor en la verificación del dominio:

Trust anchor instalada en el resolver	Estado de la respuesta DNSSEC para consultas dirigidas a		
	www.isc.org	www.ejemplo1.com	www.ejemplo2.gov
ninguna	inseguro	inseguro	Inseguro
root	seguro	inseguro	Inseguro
.com	inseguro	seguro	Inseguro
ejemplo2.gov	inseguro	inseguro	Seguro

Tabla 8. Trust anchor y verificacion de dominio

- **Proceso de resolución y verificación DNSSEC**

Cuando es así requerido por un cliente, el servidor autoritativo añadirá datos adicionales DNSSEC a la respuesta. Estos datos proceden de la firma digital del registro solicitado (RRSIG). En el caso de que el recurso solicitado no exista, se responderá con un registro NSEC3 para autentificar la respuesta. NSEC3 devuelve un hash del siguiente dominio autoritativo del servidor para no revelar en texto claro el nombre del siguiente dominio autoritativo y así evitar ataques de enumeración de dominios utilizando consultas DNSSEC.

NOTA: NSEC3 es una mejora de una primera versión del registro NSEC, que ha sido mejorado para evitar la enumeración de dominios solicitando nombres inexistentes, puesto que, anteriormente con los registros NSEC ante una consulta sobre un dominio inexistente, se devolvía el siguiente dominio autoritativo del servidor y los tipos de registros contenidos en él así como su correspondiente firma RRSIG.

Verificación de una respuesta

El cliente lo primero que hará es verificar los datos recibidos del recurso solicitado .Para ello genera un hash del conjunto de registros de la respuesta (RRset) usando el algoritmo referenciado en la misma. Por otra parte, y usando la clave pública DNSKEY que obtiene en la respuesta (en la sección ADDITIONAL) comprueba la firma incluida en el registro RRSIG, obteniendo el hash que debe coincidir con el anteriormente calculado. Esto confirma la autenticidad de los datos en base a la clave y firma recibidos

Cadena de confianza. Trust Anchor

Pero ¿qué garantiza que ambos, DNSKEY y el RRSIG no han sido modificados? Aquí, es donde entra en juego la verificación de la cadena de confianza desde el *trust anchor* que el cliente *resolver* tiene instalada y en el cual confía. Puede ser un *trust anchor* local para un dominio concreto (una isla de seguridad), o idealmente, la clave pública de un servidor *root* para un dominio globalmente seguro. Por lo tanto, si la DNSKEY no es un *trust anchor*, es necesario verificar su autenticidad. Para ello se pregunta a la zona padre por el registro DS (*Delegation Signer*) del dominio hijo que se está resolviendo.

En la respuesta en nodo padre adjunta:

- DS conteniendo el hash de la clave pública que se quiere validar (el nodo hijo)
- El registro de firma RRSIG correspondiente al DS
- Clave pública DNSKEY (nodo padre) para verificar la firma

Con estos datos y del mismo modo que para validar un registro, el *resolver* debe verificar la validez de la DNSKEY en sí misma que utilizó para verificar la firma del registro. Para ello debe igualmente verificar su firma, y esa operación se realiza con la DNSKEY pública del nodo padre, quien firmó la clave DNSKEY del nodo hijo. De nuevo, e iterativamente, si esta última DNSKEY (la el nodo padre) no es un *trust anchor*, (clave pública en la que un resolver confía y no necesita verificar) el proceso se repite con el nodo padre superior en la jerarquía DNS, y finaliza cuando se llega a un nodo donde encuentra el *trust anchor* (en último caso, el nodo root).

En la se muestra de forma gráfica la cadena. En detalle, siguiendo los pasos numerados del 1 al 8 de la *Ilustración 26*, se puede seguir el proceso de comprobaciones realizadas hasta llegar al trust anchor en la resolución del dominio *www.isc.org*:

Usando la clave ZSK se valida el RRSIG de www.isc.org (1). El RRSIG correspondiente a la clave ZSK se valida con la clave KSK. (2). Esta clave debe ser verificada si no se trata de una *trust anchor*. Si esta clave de dominio no es ya una clave de confianza o *trust anchor*, entonces el cliente debe consultar la zona padre (zona org) solicitando el registro DS de la zona hijo (isc.org). Esta consulta debe devolver un registro DS, un registro RRSIG asociado con el DS y un registro DNSKEY de la zona padre. El registro DS se puede validar contra el RRSIG, utilizando la clave pública contenida en el registro DNSKEY (3, 4, 5). Esta clave pública, a su vez, debe ser validada. Este proceso iterativo construye una cadena de confianza que, finalmente, lleva a obtener una clave coincidente con la clave de confianza configurada localmente, en este caso, la clave de la zona *root zone* (6,7, 8). En ese punto la respuesta de DNS puede considerarse válida.

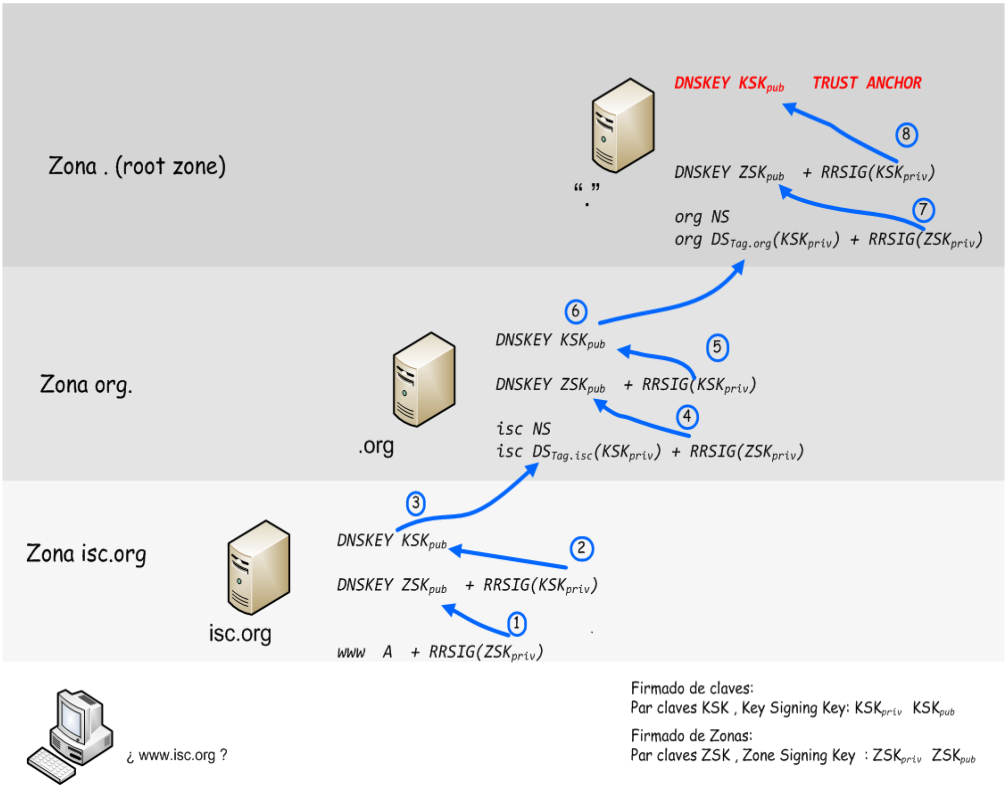

Ilustración 26. Verificación de la cadena confianza para www.isc.org

A través de la utliidad dig, puede comprobarse explícitamente la verificación DNSSEC. Una demostración puede verse en el *Anexo Test de la cadena de confianza DNSSEC usando dig.*

DIFICULTADES EN EL USO DE DNSSEC

DNSSEC tiene como contrapartida ciertos problemas y dificultades que se presentan en su utilización, entre los que se pueden destacar los siguientes:

Dificultad de implementación y mantenimiento. Respecto a DNS, resulta más complejo de implementar y requiere una cuidadosa atención en el mantenimiento de las zonas y las claves. Cualquier problema relacionado con el firmado o claves caducadas causará problemas en la resolución a clientes DNSSEC.

Tamaño de las respuestas. Una respuesta DNSSEC incrementa notoriamente su tamaño respecto a una respuesta DNS convencional, lo que conlleva un mayor uso de recursos de red y proporciona un vector explotable a ataques de amplificación DNS tal y como se explicó en el apartado de ataques de denegación de servicio *ATAQUE DE AMPLIFICACIÓN DNS*

Rendimiento y Resolución DNSSEC. El proceso de resolución y verificación de firmas supone un incremento de procesado por parte del *resolver* que puede impactar en el tiempo de respuesta y causar reintentos por parte de clientes, lo que acentuará la carga**.**

Renovación de claves. La correcta renovación de las claves supone una atención extra a la hora de administrar el servidor.

Sincronización. Puesto que para la verificación de firmas y vigencia de las claves se necesita comprobar los timestamps para determinar los periodos de validez, es necesaria una correcta sincronización de tiempo respecto a la referencia del firmante. Esto puede constituir un vector de ataque si se consigue cambiar la referencia de sincronización de tiempo de un servidor que puede provocar una denegación de servicio por problemas al verificar los tiempos de validez.

DESPLEGANDO DNSSEC

Cada organización es distinta y cada una debe estudiar y trazar un plan propio de despliegue. En principio, algunas recomendaciones básicas para llevar a cabo una implementación de DNSSEC serían:

- **Establecer una política de seguridad para DNSSEC**
  - Determinar qué zonas necesitan ser firmadas. Generalmente la mayoría de organizaciones comienzan por firmar únicamente sus zonas públicas de internet
  - Decidir que servidores servirán zonas firmadas y actualizar y adaptar convenientemente el software.
  - Establecer procedimientos de generación de claves y renovación. Almacenamiento seguro y periodicidad de renovación
  - Escoger convenientemente los parámetros criptográficos. Algoritmos de cifrado, longitud de clave, tiempo de validez, etc.

- **Despliegue y test**

Diseñar la implementación de DNSSEC siguiendo manuales de buenas prácticas, como el descrito por la ENISA23 en su publicación *Good Practice Guide for deploying DNSSEC*[24]

Empezar una prueba piloto con una réplica de una zona existente y firmarla. Si se crea un subdominio de la misma, configurar el trust anchor y probar la validación

[23] ENISA: European Union Agency for Network and Information Security. http://www.enisa.europa.eu
[24] Good Practice Guide for deploying DNSSEC: http://www.enisa.europa.eu/activities/Resilience-and-CIIP/networks-and-services-resilience/dnssec/gpgdnssec

INDICES Y REFERENCIAS

REFERENCIAS TÉCNICAS

[RFC1034] *Mockapetris, P., "Domain names - concepts and facilities," STD 13, RFC 1034, November 1987.*

[RFC1035] *Mockapetris, P., "Domain names - implementation and specification," STD 13, RFC 1035, November 1987.*

[RFC2671] *Vixie P. "Extension mechanisms for DNS (EDNS0) ,August 1999.*

[RFC4033] *R. Arends, R. Austein, M. Larson,D. Massey, S. Rose "DNS Security Introduction and Requirements", March 2005*

[RFC2827] *Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," BCP 38, RFC 2827, May 2000.*

[RFC3013] *Killalea, T., "Recommended Internet Service Provider Security Services and Procedures," BCP 46, RFC 3013, November 2000.*

[RFC3833] *Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)," RFC 3833, August 2004.*

[RFC4033] *Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements," RFC 4033, March 2005.*

[vu-457875] *United States CERT, "Various DNS service implementations generate multiple simultaneous queries for the same resource record," VU 457875, November 2002.*

[RFC1034] *Mockapetris, P., "Domain names - concepts and facilities," STD 13, RFC 1034, November 1987.*

[RFC1035] *Mockapetris, P., "Domain names - implementation and specification," STD 13, RFC 1035, November 1987.*

[RFC2671] *Vixie P. "Extension mechanisms for DNS (EDNS0) ,August 1999.*

[RFC4033] *R. Arends, R. Austein, M. Larson,D. Massey, S. Rose "DNS Security Introduction and Requirements", March 2005*

DOCUMENTACIÓN

ENISA	*European Union Agency for Network and Information Security*
US-CERT	*Unites States Computer EmergencY Readiness Team*
DHS	*US Homeland Security Department*
NIST	*National Institute of Standards and Technology*
ISC	*[RFC1034]*

INDICE DE ILUSTRACIONES

INDICE DE CONFIGURACIONES

TRANSACTION SIGNATURE. TSIG.

TSIG, acrónimo de *Transaction SIGnatures*, es un mecanismo para verificar la identidad de los servidores DNS con quienes se está comunicando. TSIG fue descrito en el RFC 2845[25]. Este mecanismo fue diseñado antes de la aparición de DNSSEC para dotar de autenticación e integridad a las transacciones DNS y originalmente pensado para la protección de actualizaciones dinámicas.

Message Authentication Code (MAC)

TSIG hace uso de MAC (Message Authentication Code) y se utiliza de una clave compartida entre los servidores primarios y esclavos para cifrar criptográficamente los mensajes intercambiados. Las claves, ya que son compartidas, deben distribuirse por vías o canales seguros y cambiarse con relativa frecuencia para reducir el riesgo de compromiso de la clave.

Registro RRSIG

Una transacción TSIG incluye un registro RRSIG con el MAC el cual se obtiene a partir de un hash del registro DNS a trnasmitir (lo que aporta integridad) que está a su vez se cifra con la clave compartida (aportando autenticación). De este modo, el servidor consultado enviará el registro solicitado y su correspondiente MAC en el registro RRSIG. En el receptor, el MAC contenido en el RRSIG se verifica usando su copia de la clave compartida y aplicando la misma operación que se realizó en el emisor. Si la verificación es correcta y ambos coinciden, la transacción es aceptada.

Configuración de TSIG en BIND

Para la configuración de TSIG se necesita generar la clave compartida para el cifrado. Esto se realiza mediante la utilidad de BIND *dnssec-keygen*, indicando el algoritmo de cifrado deseado para la generación. Se recomienda el algoritmo HMAC-MD5 al ser uno de obligado soporte en la especificación TSIG de DNS. La generación se muestra a continuación (*Cuadro 1*):

```
# dnssec-keygen –a hmac-md5 –b 128 –C –n host ejemplo.com
# ls
  Kejemplo.com.+157+31456.key
  Kejemplo.com.+157+31456.private <- - - Contiene la clave a utilizar

# cat Kejemplo.com.+157+31456.private
 Private-key-format: v1.2
 Algorithm: 157 (HMAC_MD5)
 Key: JuhsyAfsdsRiW4fs90==    <- - Clave generada
 Bits: AAA=
```

Cuadro 1. Generación de clave compartida TSIG

Esta clave es la que se usará en los ficheros de configuración *named.conf*. Ésta puede escribirse directamente en la directiva *key* de la clausula *options* del fichero *named.conf* pero es más

[25] RFC2845: Secret Key Transaction Authentication for DNS (TSIG) http://www.ietf.org/rfc/rfc2845.txt

recomendable guardarla en un fichero local en todos los servidores y protegido contra acceso no autorizado y referenciar este fichero en la configuración *named.conf*, como se muestra en la siguiente configuración (*Cuadro 2*):

```
# mv Kejemplo.com.+157+31456.private /chroot/named/keys/ejemplo.com.key

# chown named:named /chroot/named/keys/ejemplo.com.key
# chmod 0400 /chroot/named/keys/ejemplo.com.key

# cat /chroot/named/keys/ejemplo.com.key
 Private-key-format: v1.2
 Algorithm: 157 (HMAC_MD5)    <- - - Algoritmo
 Key: JuhsyAfsdsRiW4fs90==  <- - - Clave en base64
 Bits: AAA=
```

Cuadro 2. Protección del fichero de clave compartida TSIG

Antes de incluirlo en la configuración de *named.conf*, es necesario editar el fichero de la clave y dar el formato de claúsula "key" utilizado por BIND. Como se puede ver en el Cuadro 2, el fichero contiene 4 líneas, de las cuales se necesitan 2: el algoritmo usado y la clave en base64 para construir la claúsula "key" que se configura en named.conf. El resultado puede verse en el *Cuadro 3* a continuación:

```
# vi /chroot/named/keys/ejemplo.com.key

# cat /chroot/named/keys/ejemplo.com.key

key "ejemplo.com" {
   algorithm hmac-md5;
   secret JuhsyAfsdsRiW4fs90==;
};
```

Cuadro 3. Fichero de clave con formato de cláusula key

Una vez el fichero de clave tiene el formato necesario, puede referenciarse directamente en el fichero de configuración *named.conf,* como se ve a continuación:

```
options {
   directory /chroot/named
  include "keys/ejemplo.com.key"; // incluir el fichero que contiene la
                                  // cláusula de la clave
  server { 10.1.2.3 ;};  // especificamos que las transacciones con 10.1.2.3 se
   keys {"ejemplo.com";}; // firmen con la clave denominada "example.com"

}
zone "ejemplo.com" in {
   Type master;
   file "etc/zona_master.ejemplo.com"
   allow-transfer { key "ejemplo.com";}; //transferencia zona solo permitida
}                                        // con clave ejemplo.com
```

Configuración 17. Configuración TSIG. Clave almacenada en fichero

EJEMPLOS PRÁCTICOS DE USO DE DIG

En esta sección se presentan algunos ejemplos de uso de la aplicación dig. Dig es una herramienta distribuida con el software BIND orientada a interaccionar con servidores DNS y de gran utilidad para tareas de diagnóstico o simplemente para obtener información sobre recursos DNS.

- **Sintaxis básica**

```
dig [@servidor_dns] [dominio] [tipo_recurso] [clase] [opciones]
```

tipo_recurso: A, TXT, MX, NS, SOA, ANY etc (véase apartado 2.3)
clase : IN, CH
opciones: dig dispone de un amplio abanico de opciones. Pueden consultarse con *dig -h*
Algunas de las opciones mas habituales son:

+short → modo abreviado, solo muestra respuesta
+tcp → usar tcp en la consulta
+recurse → Forzar consulta recursiva
+nostats → No mostrar estadísticas

- **Uso básico**

Se hace uso del dominio example.com para los ejemplos

Consultas sobre recursos:

Registro A

```
kuko@DNS ~ dig example.com A

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> example.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36611
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.          IN    A

;; ANSWER SECTION:
example.com.      4935  IN    A    93.184.216.119
```

Registro A, modo abreviado:

```
kuko@DNS ~ dig example.com A +short
93.184.216.119
```

Consultar servidores de nombres (NS):

```
kuko@DNS ~ dig example.com NS +short
b.iana-servers.net.
a.iana-servers.net.
```

Registros TXT:

```
kuko@DNS ~ dig example.com TXT +short
"v=spf1 -all"
"$Id: example.com 1921 2013-10-21 04:00:39Z dknight $"
```

Transacción de transferencia de zona

```
kuko@DNS ~ dig example.com AXFR
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> example.com AXFR
;; global options: +cmd
; Transfer failed.
```

- **Consultas especificando buffer EDNS0**

Si el servidor DNS implementa EDNS0 (*véase 0 Mensajes DNS*) es posible de enviar mensajes DNS de tamaño superior a 512 bytes sobre UDP si así se solicita.

Por defecto, si la respuesta a una consulta sobre UDP es mayor de 512 bytes, el servidor insta al cliente a solicitar de nuevo la petición por TCP con una señal de 'truncation'.

En la consulta siguiente, no se especifica buffer UDP. De este modo, si la respuesta es mayor de 512 bytes (valor por defecto para UDP) se provocará la retransmisión por TCP. Obsérvese en la siguiente ilustración (*Ilustración 27*), cómo la respuesta, de tamaño 2628 bytes, se recibe tras reintenarse por TCP:

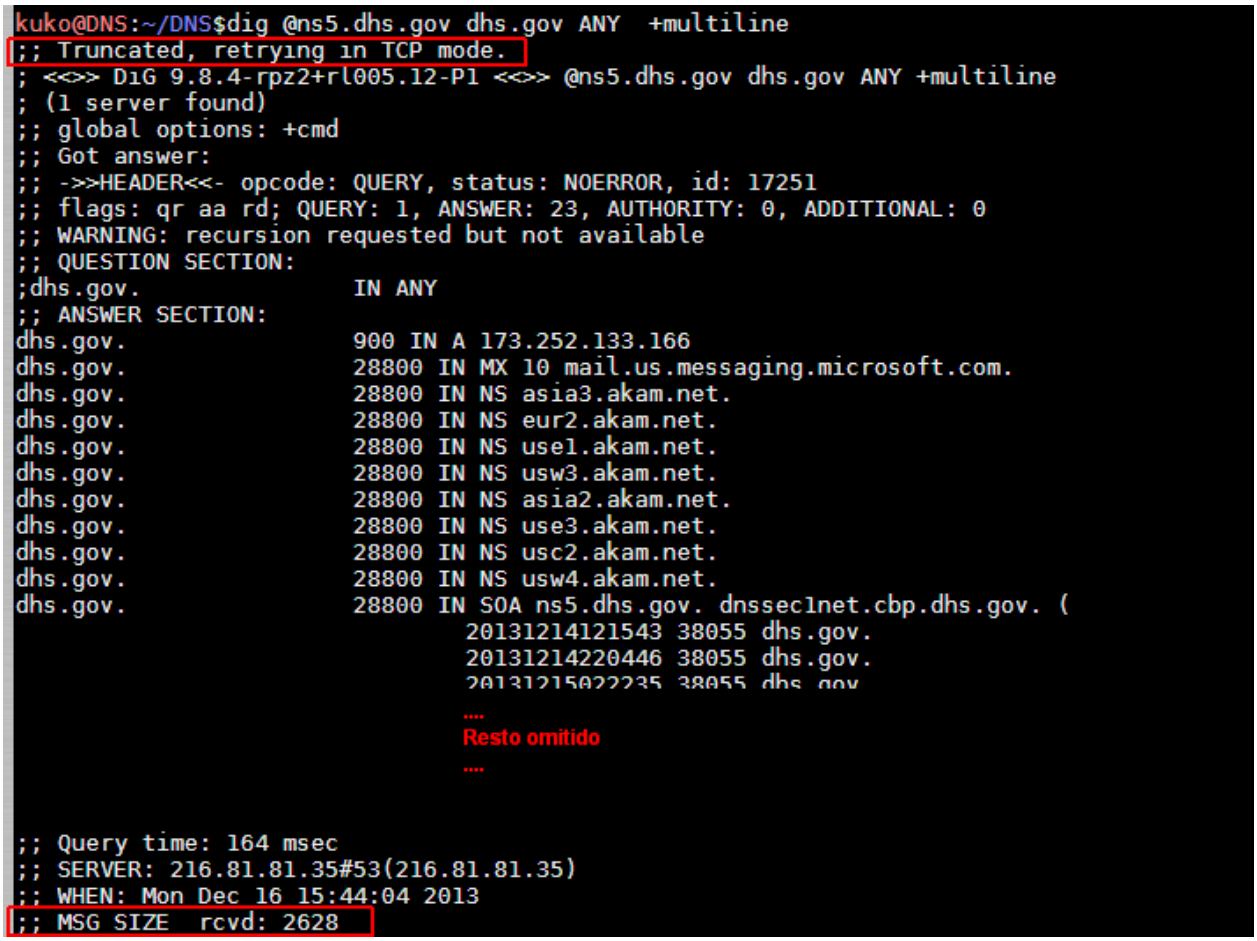

Ilustración 27. Respuesta mayor 512 bytes. Truncation *y cambio a TCP*

A continuación se repite la consulta, pero esta vez se fuerza UDP especificando un buffer de 4096 bytes. Préstese atención a la sección OPT (Ilustración 28):

```
kuko@DNS:~/DNS$dig @ns5.dhs.gov dhs.gov  ANY +multiline  +bufsize=4096

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ns5.dhs.gov dhs.gov ANY +multiline +bufsize=4096
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13545
;; flags: qr aa rd; QUERY: 1, ANSWER: 23, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;dhs.gov.                IN ANY

;; ANSWER SECTION:
dhs.gov.                900 IN A 173.252.133.166
dhs.gov.                28800 IN MX 10 mail.us.messaging.microsoft.com.
dhs.gov.                28800 IN NS use1.akam.net.
dhs.gov.                28800 IN NS usc2.akam.net.
dhs.gov.                28800 IN NS eur2.akam.net.
dhs.gov.                28800 IN NS asia2.akam.net.
dhs.gov.                28800 IN NS usw3.akam.net.
dhs.gov.                28800 IN NS usw4.akam.net.
dhs.gov.                28800 IN NS asia3.akam.net.
dhs.gov.                28800 IN NS use3.akam.net.
dhs.gov.                28800 IN SOA ns5.dhs.gov. dnssec1net.cbp.dhs.gov. (
                .... Resto omitido ....
;; Query time: 167 msec
;; SERVER: 216.81.81.35#53(216.81.81.35)
;; WHEN: Mon Dec 16 15:55:51 2013
;; MSG SIZE  rcvd: 2639
```

Ilustración 28. Consulta especificando buffer UDP. Obsérvese la pseudosección EDNS0

- **Test de la cadena de confianza DNSSEC usando dig**

Para verificar la cadena de confianza dnssec, se parte de un "*trust anchor*" o clave de confianza, que será la base de la verificación. En este ejemplo se usan las claves de los servidores *root* que son descargados y almacenados en un fichero:

Se obtienen las claves DNSSEC de los servidores raíz. Trust anchors:

```
dig . DNSKEY
```

```
kuko@DNS ~ dig . DNSKEY | grep -Ev "^($|;)" > root.keys
kuko@DNS ~
kuko@DNS ~ ls root.keys
root.keys
kuko@DNS ~
kuko@DNS ~ cat root.keys
.                       9514    IN      DNSKEY  257 3 8 AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gc
MVDxP/VHL496M/QZxkjf5/Efucp2gaD X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz W5hOA2hzCTMjJPJ8LbqF
knNnulq QxA+Uk1ihz0=
.                       9514    IN      DNSKEY  256 3 8 AwEAAb8sU6pbYMWRbkRnEuEZw9NSir707TkOcF+UL1XiK4NDJOvXRy
v00xWExKQjbvnRPI4bqpMwtHVzn6Wyb BZ6kuqED
kuko@DNS ~
```

Ilustración 29. Trust anchors servidores root

A continuación tomando las claves obtenidas de los servidores root como trust anchor, se lanza la cadena de verificación con dig: *dig +sigchase +trusted-key=./root.*

```
kuko@DNS ~ dig +sigchase +trusted-key=./root.keys www.isc.org A +multiline
;; RRset to chase:
www.isc.org.            59 IN A 149.20.64.69


;; RRSIG of the RRset to chase:
www.isc.org.            59 IN RRSIG A 5 3 60 20140402233240 (
                            20140303233240 4521 isc.org.
                            VGyObmU7Arxur6ygjb/KXh/3GBRbbolWMn7xZCaY9/Rz
                            0mPmRsqI42CBI9vXdvdho7ZLu+4/EcZsnAzF4YltTSLM
                            vmGRneE4IFq6wRHC3UCmYh0GAqt0chJ0IbjGTFGFYKzM
                            RXWOSuQHvVztep/wsMR+Reth2Ab4PRQhG4Vc9dI= )



Launch a query to find a RRset of type DNSKEY for zone: isc.org.

;; DNSKEYset that signs the RRset to chase:
isc.org.                7199 IN DNSKEY 257 3 5 (
                            BEAAAAOhHQDBrhQbtphgq2wQUpEQ5t4DtUHxoMVFu2hW
                            LDMvoOMRXjGrhhCeFvAZih7yJHf8ZGfW6hd38hXG/xyl
                            YCO6Krpbdojwx8YMXLA5/kA+u50WIL8ZR1R6KTbsYVMf
                            /Qx5RiNbPClw+vT+U8eXEJmO20jIS1ULgqy347cBB1zM
                            nnz/4LJpA0da9CbKj3A254T515sNIMcwsB8/2+2E63/z
                            ZrQzBkj0BrN/9Bexjpiks3jRhZatEsXn3dTy47R09Uix
                            5WcJt+xzqZ7+ysyLKOOedS39Z7SDmsn2eA0FKtQpwA6L
                            XeG2w+jxmw3oA8lVUgEf/rzeC/bByBNsO70aEFTd
                            ) ; key id = 12892
                        7199 IN DNSKEY 256 3 5 (
                            AwEAAbJpDF4RemdHHE/HrJJhR3zpzAQ6zsHqFv0i4lCW
                            TUf4sX+cq3vSu7fKO4QJtm97S1sbcnmHonVE3QPzLOsq
                            sY630Wy5JzrPK3gUvQLgfIsovo2v+dosITL8WbvjU1mE
                            XhIwfuuBhYmYSKySZ0X9gpHGhdxRd+J8M7riPfN7kHLP
                            ) ; key id = 4521


;; RRSIG of the DNSKEYset that signs the RRset to chase:
isc.org.                7199 IN RRSIG DNSKEY 5 2 7200 20140402230130 (
                            20140303230130 4521 isc.org.
                            NS2bTHzpgatqh4LVtU0x9aVi5yJ4gpIg6lmSB2VJFoJL
                            P+ySeG/srlHqecMni9UxmmmR6qNlCwXqjLmd5IHOrIG0
                            yniepDycwYkkX1jDnF+LDcIgS8YFxURUjDWM24WHpGsM
                            oj0pV15/aZ8LB7MVikNOiysDcIvXjgLDJxGgWuo= )
                        7199 IN RRSIG DNSKEY 5 2 7200 20140402230130 (
                            20140303230130 12892 isc.org.
                            AMnoRGL1DxDFNZTv0GVxmIu2J0kZeiLo8kjVHbn873rM
                            KxzOzxCsnWEU2fBqm3lhXU/w7NnUFNHvswNN9SkroLk4
                            DxBrlkVTxorkTdlsiC6S0rSALzN5Brr/sqshkBywfIKD
                            c00UgAqTCuhfTdYfLwghpR4i3Z0SImvs6hFZw0zCSTMd
                            Wdx+l/rLAwt/GFnwQuc0q6SVOMk6A3FWzKp0JIBysD3R
                            Q2kiRunNEb0il/3KU7bA3V74AMWJufZP7awgRZaV/k1M
                            vMSQ6G9xoC1TETPMe+uR3x3MymwHChZ6XfGQ7CfmcpTu
                            7P3HPbAi+SaVj6S1wkU85GCg3f/3AypGLA== )
```

```
Launch a query to find a RRset of type DS for zone: isc.org.

;; DSset of the DNSKEYset
isc.org.                21599 IN DS 12892 5 1 (
                                982113D08B4C6A1D9F6AEE1E2237AEF69F3F9759 )
                        21599 IN DS 12892 5 2 (
                                F1E184C0E1D615D20EB3C223ACED3B03C773DD952D5F
                                0EB5C777586DE18DA6B5 )


;; RRSIG of the DSset of the DNSKEYset
isc.org.                21599 IN RRSIG DS 7 2 86400 20140318154949 (
                                20140225144949 24209 org.
                                DGeds8tNSuQd3z5DNHwCqtHncjVshmP5iy1FwALF+l58
                                GxYVHigZBDwXBFDejW+QIxH8rfkrr8X0hDu19X/URemn
                                igsfmxFZS2gZYg+610WsaKwwMncJF5I/2fMxV0eTYDb+
                                mQLoioZ48ri7/xQJ1VW2Hp4tMTa/NRDqA6wZ248= )



;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING A RRset for www.isc.org. with DNSKEY:4521: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Now, we are going to validate this DNSKEY by the DS
;; OK a DS valids a DNSKEY in the RRset
;; Now verify that this DNSKEY validates the DNSKEY RRset
;; VERIFYING DNSKEY RRset for isc.org. with DNSKEY:12892: success
;; OK this DNSKEY (validated by the DS) validates the RRset of the DNSKEYs, thus th
;; Now, we want to validate the DS :  recursive call


Launch a query to find a RRset of type DNSKEY for zone: org.

;; DNSKEYset that signs the RRset to chase:
org.                    52 IN DNSKEY 257 3 7 (
                                AwEAAZTjbIO5kIpxWUtyXc8avsKyHIIZ+LjC2Dv8naO+
                                Tz6X2fqzDC1bdq7HlZwtkaqTkMVVJ+8gE9FIreGJ4c8G
                                1GdbjQgbP1OyYIG7OHTc4hv5T2NlyWr6k6QFz98Q4zwF
                                IGTFVvwBhmrMDYsOTtXakK6QwHovA1+83BsUACxlidpw
                                B0hQacbD6x+I2RCDzYuTzj64Jv0/9XsX6AYV3ebcgn4h
                                L1jIR2eJYyXlrAoWxdzxcW//5yeL5RVWuhRxejmnSVnC
                                uxkfS4AQ485KH2tpdbWcCopLJZs6tw8q3jWcpTGzdh/v
                                3xdYfNpQNcPImFlxAun3BtORPA2r8ti6MNoJEHU=
                                ) ; key id = 9795
                        52 IN DNSKEY 256 3 7 (
                                AwEAAXIJ6PWJHkI4gLfGoHDPxQwS1kVKhYcjEwKn76TM
                                EQgw3mr2rDMsKiC7vfTnaGxIbqCodo4xNixVp8MgAuUA
                                +YLrSPft5ivGLkTXmZnxmKTaJocMVsyGjLiQL0oJTFJC
                                H25xf/wBHJ7PAeqbaQvgRGLTR8Jmqyjfrg UxZw0qRhGN
                                ) ; key id = 24209
                        52 IN DNSKEY 256 3 7 (
                                AwEAAa+yHvpOo3f7XS1vtKPGH6AD1OkmYUtnRlkkCO9B
                                KJ0OCCvYSWh5NWLJjIMXRzVpituqoLtiYfhdDYQH5JzR
                                VW6lCtT+2SiWmEx+7GnSyMT48858uC02AYlJVfbitCpo
                                GGdzyLTiMxtMlztpRyCAvaDujnx+2GBo7zgb50f5gQJp
                                ) ; key id = 1829
```

```
                        52 IN DNSKEY 257 3 7 (
                                AwEAAYpYfj3aaRzzkxWQqMdl7YExY81NdYSv+qayuZDo
                                dnZ9IMh0bwMcYaVUdzNAbVeJ8gd6jq1sR3VvP/SR36mm
                                GssbV4Udl5ORDtqiZP2TDNDHxEnKKTX+jWfytZeT7d3A
                                bSzBKC0v7uZrM6M2eoJnl6id66rEUmQC2p9DrrDg9F6t
                                XC9CD/zC7/y+BNNpiOdnM5DXk7HhZm7ra9E7ltL13h2m
                                x7kEgU8e6npJlCoXjraIBgUDthYs48W/sdTDLu7N59rj
                                CG+bpil+c8oZ9f7NR3qmSTpTP1m86RqUQnVErifrH8Kj
                                DqL+3wzUdF5ACkYwt1XhPVPU+wSIlzbaAQN49PU=
                                ) ; key id = 21366


;; RRSIG of the DNSKEYset that signs the RRset to chase:
org.                    52 IN RRSIG DNSKEY 7 1 900 20140318154949 (
                                20140225144949 9795 org.
                                OfT8srp+lEzCcGDHOX0xPb0Z0Vmht+w2yxPJnTfkOqBJ
                                yVkhWV0lDfe8wfPH+xHzKtAw0E+91Hz9H1S4t+6C+Yp8
                                Ls/+F+z80h5Gkjq0tTOYN8kEfCSGYK3rAL6tq4lq6ZPZ
                                +cv9GizeOmWX7wUxkVETUU2+KVoGHLRpgEGQOI6IGR1Y
                                8gsA7NAY2AYCPJ+IekQPqfj/Y4up1dCdI51jhbe1n1zL
                                RDCYIYYBKX6xtaXqjbQUEQ4yv/H2thCdzjosLBvzVj1d
                                W+qRywhNrKWw30bWZYZxSerxSrpivrzikjVnWBfetIbo
                                OeO56u0D7Bp0oH8ACc6ocPIdAd+H3JFuBw== )
                        52 IN RRSIG DNSKEY 7 1 900 20140318154949 (
                                20140225144949 21366 org.
                                hhrN8ugInW5Zag9cjMDdPgbk5nbUp1aXDGQPXRSTZUoK
                                Ctsg2CzpDqN3wx9ly+0aeO5SUexTJM/i9hNChQayLmTj
                                Ts42kpQxV4pgKXBxxL6dOaFIOPv3SPssQlKPm3OoEBrb
                                WW1rpE3E/nXbAaFMnSQoHzil9xVxhoJt382e8uBYIL8/
                                zrGTrmgyEwl9eubam9zXS++GVHN+5CbqaT3S/39I7SAF
                                iusoT0qv1+mz9SVy1IJSezNh+sLCiJIhiZtlGTSwEFXF
                                HwlAlZuKoJ4cl37n+4nTGnxm13r0U/LKxXItNQ80GVfB
                                8tPCbES+/ujS69t0/f/Se43fEa7qPQLtpw== )
                        52 IN RRSIG DNSKEY 7 1 900 20140318154949 (
                                20140225144949 24209 org.
                                OEfP9P1qN0fytGJrqmcOqnMWjVM4rrXntrfDSTvoUSwu
                                6Gmdh9aLPU9uxoS1WBrS9FBLih5KJPj6JzcsLL4kFzjw
                                7z9F59bREx6dcY9XhiK/w0yf7O1ABB3COwpPcCjOM2Db
                                tZu1LC/aXLK7P4QRhB95D3UcJl1zI3tepkxCBu0= )



Launch a query to find a RRset of type DS for zone: org.

;; DSset of the DNSKEYset
org.                    8732 IN DS 21366 7 2 (
                                96EEB2FFD9B00CD4694E78278B5EFDAB0A80446567B6
                                9F634DA078F0D90F01BA )
                        8732 IN DS 21366 7 1 (
                                E6C1716CFB6BDC84E84CE1AB5510DAC69173B5B2 )
```

```
;; RRSIG of the DSset of the DNSKEYset
org.                    8732 IN RRSIG DS 8 1 86400 20140314000000 (
                                20140306230000 33655 .
                                B1FZwn3bjwCiBfiMlpLRlEIukKZJjALVYFhWSzhYrLuj
                                8pe4B1t7f1iFJTkCBJOd5N9B0QbDJhfP0Nz5rgv82v6f
                                eSlHbkSBp1jzG1M/o6EQweODF4kU5PqdBEEga73bNgT5
                                r+g/28lHxrXj9aQDeYWFH+M4Wp4/d8WaqTSg+uA= )



;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING DS RRset for isc.org. with DNSKEY:24209: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Now, we are going to validate this DNSKEY by the DS
;; OK a DS valids a DNSKEY in the RRset
;; Now verify that this DNSKEY validates the DNSKEY RRset
;; VERIFYING DNSKEY RRset for org. with DNSKEY:21366: success
;; OK this DNSKEY (validated by the DS) validates the RRset of the DNSKEYs, thus th
;; Now, we want to validate the DS :  recursive call


Launch a query to find a RRset of type DNSKEY for zone: .

;; DNSKEYset that signs the RRset to chase:
.                       9238 IN DNSKEY 257 3 8 (
                                AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQ
                                bSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh
                                /RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWA
                                JQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXp
                                oY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3
                                LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGO
                                Yl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGc
                                LmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=
                                ) ; key id = 19036
                        9238 IN DNSKEY 256 3 8 (
                                AwEAAb8sU6pbYMWRbkRnEuEZw9NSir707TkOcF+UL1Xi
                                K4NDJOvXRyX195Am5dQ7bRnnuySZ3daf37vvjUUhuIWU
                                AQ4stht8nJfYxVQXDYjSpGH5I6Hf/0CZEoNP6cNvrQ7A
                                FmKkmv00xWExKQjbvnRPI4bqpMwtHVzn6WybBZ6kuqED
                                ) ; key id = 33655


;; RRSIG of the DNSKEYset that signs the RRset to chase:
.                       9238 IN RRSIG DNSKEY 8 0 172800 20140316235959 (
                                20140302000000 19036 .
                                ia/OHAcEyCgkaADe5cO7DSGpNyNskxaDQQXvivOUe4Wo
                                PXye82dEgObGKqtHc2TSJ/PNsczGdUPr48ZbujbM6mBr
                                NKqSloX7fl0eCciXcKXKLNzoCG1WDFxKPno/VQS3+Ev+
                                6ll9jaNPu/sCIq0ziSh/7hw8sNzG2XBmBq1lJRZ/SZBX
                                VrzH0E8kNjW/BQAk/D8K0VMKvR7JAzd/rXUkiSYkj8h2
                                0lW6Zi+5mTm5c51Ujn+CzUPsudVwnUesD+Um7wAXgV0J
                                F/Rv5/Pd21VppsLQG25qf4KkItHpNXc1Re005J7Sr5jH
                                ouhgIo4Duc+4NtinY/D49jjr2+YU0Fza8w== )
```

```
Launch a query to find a RRset of type DS for zone: .
;; NO ANSWERS: no more

;; WARNING There is no DS for the zone: .



;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING DS RRset for org. with DNSKEY:33655: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Ok, find a Trusted Key in the DNSKEY RRset: 19036
;; VERIFYING DNSKEY RRset for . with DNSKEY:19036: success

;; Ok this DNSKEY is a Trusted Key, DNSSEC validation is ok: SUCCESS

kuko@DNS ~
```

ENLACES Y HERRAMIENTAS ÚTILES

En esta sección se ofrecen recursos y herramientas útiles para la gestión, resolución de problemas y auditoría de sistemas DNS

- **Herramientas online**

FUNCIONALIDAD	URL
Tests de validación DNSSEC	http://dnssec.vs.uni-due.de/
Tráfico DNS malicioso. Umbrella Labs & OpenDNS	http://dnsviz.net/
	http://labs.umbrella.com/global-network/
Tests generales de estado servicio DNS	http://www.dnsstuff.com/
	http://www.dnsinspect.com/
	http://dr.xoozoo.com/
	http://www.simpledns.com/lookup.aspx
Información sobre dominios	https://www.robtex.com

Tabla 9. Herramientas online de diagnóstico DNS

- **Herramientas software**

FUNCIONALIDAD	HERRAMIENTA
Herramientas clientes para resoluciones DNS	*Dig (bind).* http://www.isc.org/downloads/
	Nslookup. Misma uso que dig, aunque obsoleta a favor de dig
Herramientas para escanear, y obtener de información de dominios	*Fierce.* http://ha.ckers.org/fierce/
	Dnsenum. https://code.google.com/p/dnsenum/
	Dnsrecon. https://github.com/darkoperator/dnsrecon/blob/master/dnsrecon.py
Múltiples pruebas para comprobar la consistencia y validez DNS de un dominio.	*Dnswalk* http://sourceforge.net/projects/dnswalk/

Tabla 10 Software y útiles DNS