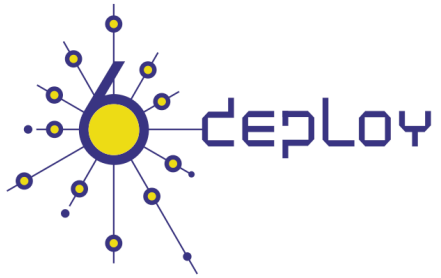
DNS / DNSSEC en IPv6 e IPv4



Workshop IPv6 – 8-10 de agosto 2011

Santiago de Chile

Carlos Martínez (carlos @ lacnic.net)

Motivación DNS (1)

- El protocolo IP asigna direcciones individuales a todos los hosts en una cierta red
- Estas direcciones son simplemente números binarios sin mayor estructura
- Para enviar tráfico IP de un host a otro esto es técnicamente lo único que hace falta
- Sin embargo, para los usuarios de la red es prácticamente imposible recordar o manejar estos números, es preferible contar con identificadores textuales

Introducción (2)

- DNS: *Domain Name System*
- Propósito básico:
  - Traducir números IP en nombres textuales mas amigables para los usuarios “humanos” de la red
- Propósitos adicionales:
  - Soporte a diferentes servicios a dar sobre la red
    - Correo electrónico
    - Sub-delegaciones de nombres
    - Resolución reversa
      - Reverso: correspondencia nombre -> número IP

Introducción (3)

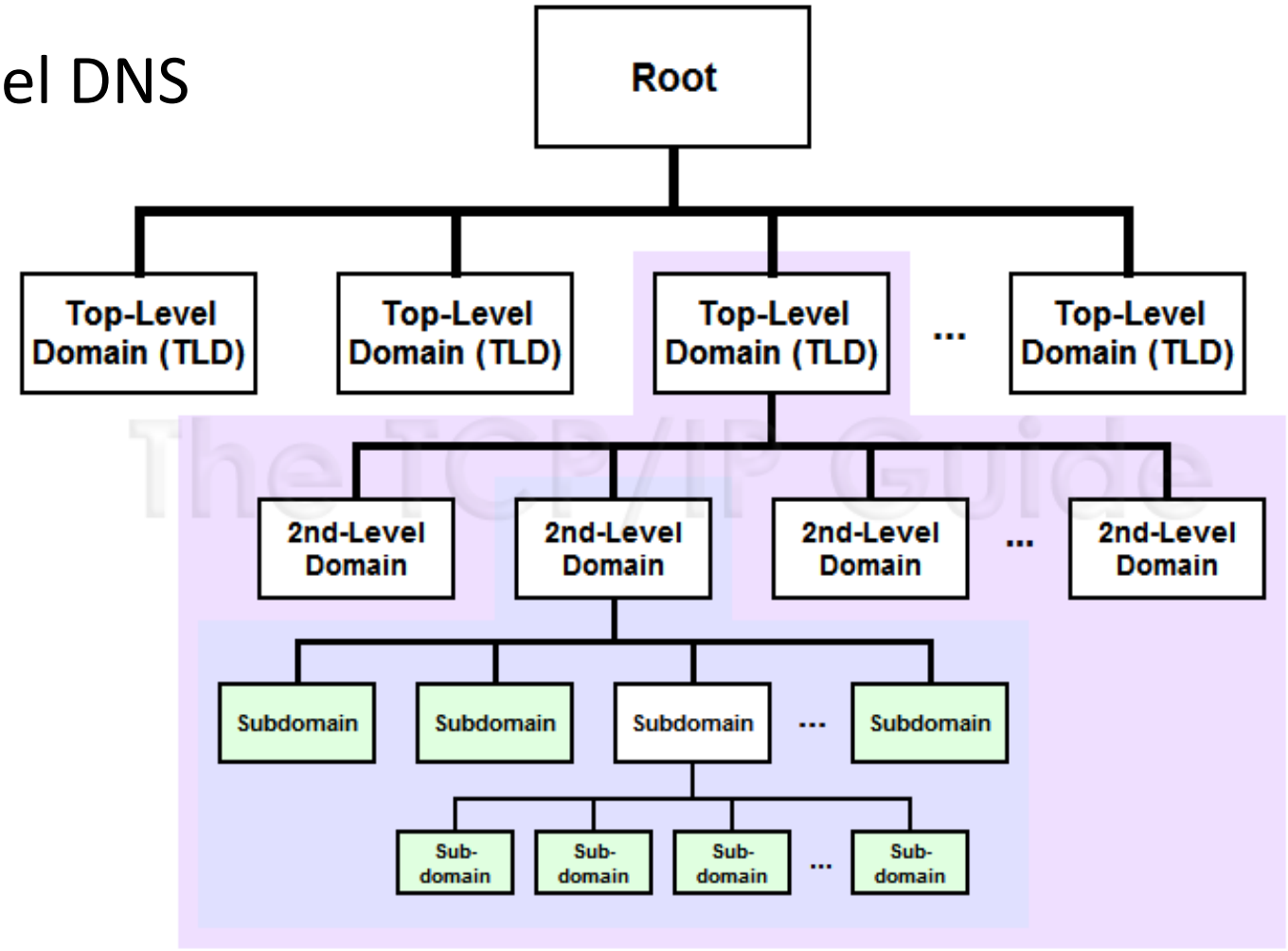
- Requerimientos del sistema:
  - Apoyo a diferentes consultas y aplicaciones
    - Nombres directos, reversos, apoyo a aplicaciones
  - Distribución de la administración
    - En Internet no hay administración centralizada sino que todo es por naturaleza distribuido. El DNS debe soportar y apoyar esta forma de trabajo.
  - Performance adecuada
    - Las consultas deben responderse lo mas rápidamente posible.
  - Confiabilidad adecuada

Introducción (4)

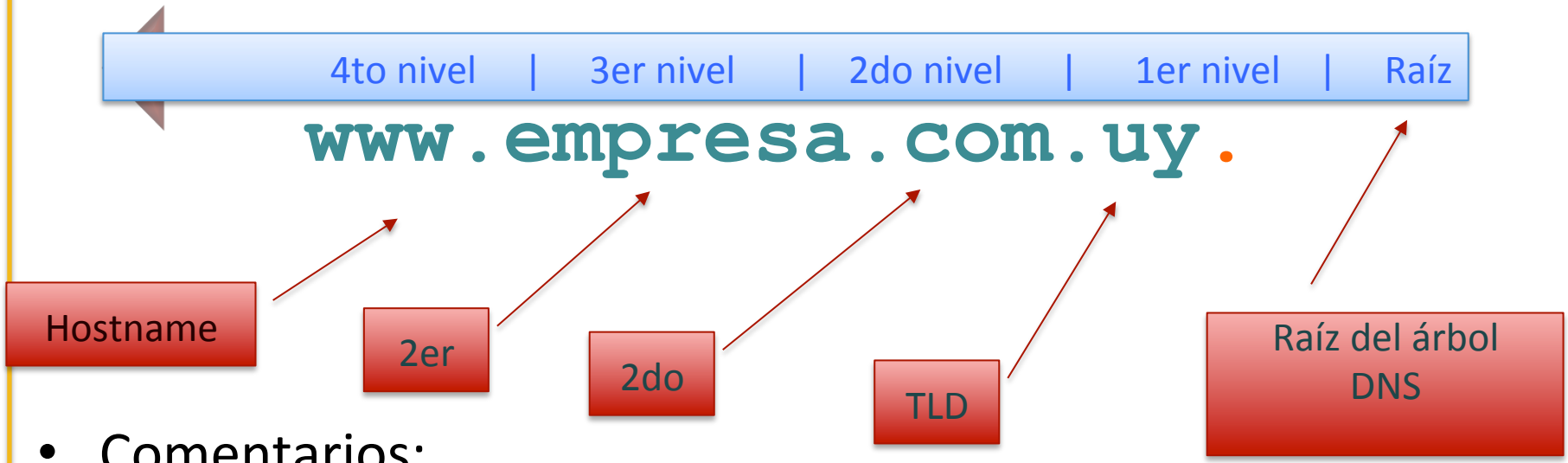
- El DNS como base de datos:
  - El objetivo principal del DNS es entonces almacenar información de mapeo entre nombres y números IP
    - Directa y reversa
  - El sistema opera entonces como una base de datos distribuida en la que existe la posibilidad de delegar la administración de sectores del espacio de nombres a diferentes organizaciones

Introducción (5)

- El árbol del DNS



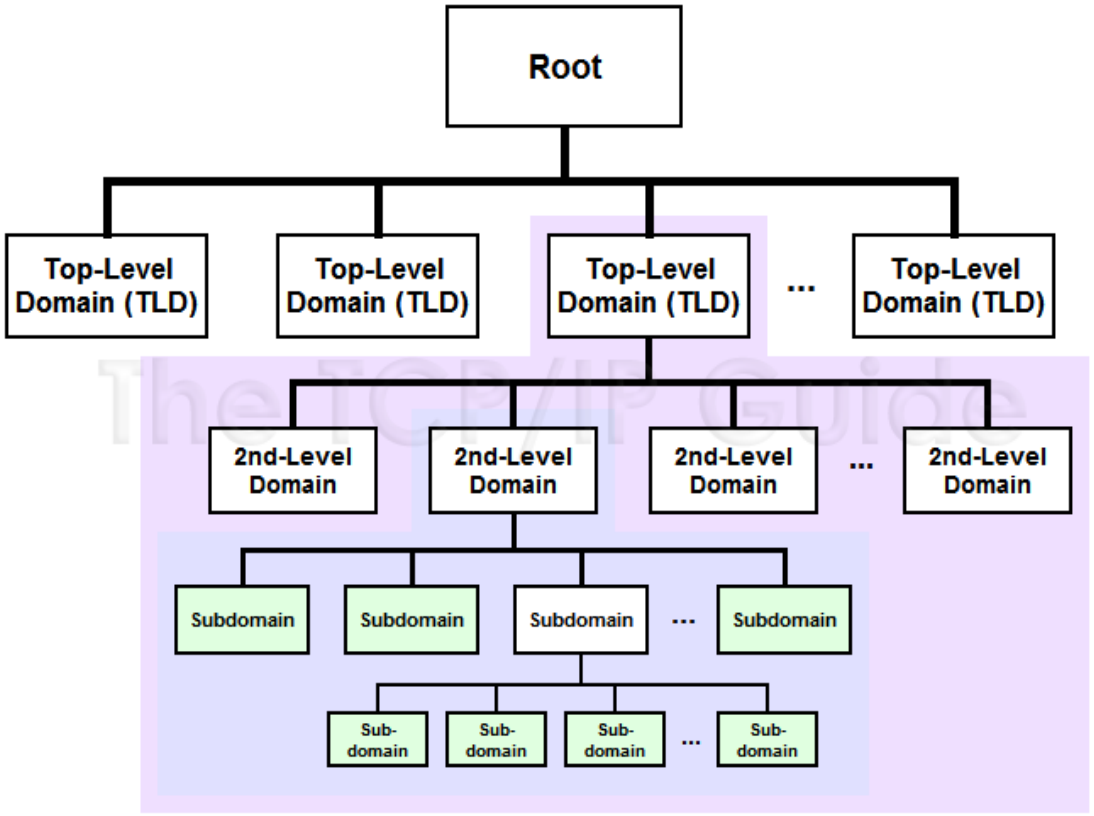
Estructura de un nombre de dominio



- Comentarios:
  - Los niveles del árbol reflejan las divisiones administrativas
  - El root del arbol esta siempre presente de forma ímplicita
  - Restricciones:
    - 127 niveles, 63 caracteres por etiqueta
  - Los niveles superiores "*delegan*" hacia los inferiores

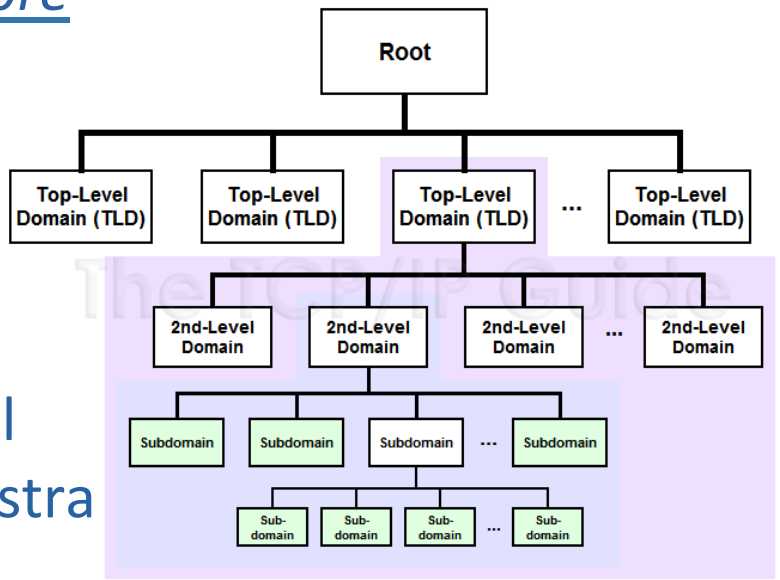
División administrativa

- DNS es una base de datos distribuida
  - Repositorio de pares (clave, valor)
- Delegación de niveles superiores a inferiores



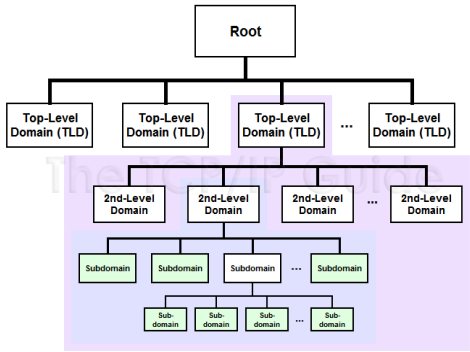
Zonas y Autoridad

- Zonas
  - A cada dominio (*incluyendo siempre al root*) le corresponde lo que se denomina una *zona* de DNS
- Autoridad
  - Cada zona define una región de *autoridad* donde se le reconoce el derecho organización que administra la misma
    - Respuestas *autoritativas*



Primarios y secundarios

- Primarios y secundarios nos permiten brindarle alta disponibilidad a una zona
- Primarios contienen informacion configurada sobre una zona
- Los secundarios la copian
  - AXFR
- Las respuestas de los secundarios
  - También son autoritativas



Resource Records

- La información en la base de datos del DNS está estructurada en un conjunto de *resource records*:
  - SOA, A, AAAA, NS, MX, PTR, TXT, etc.
- Cada RR representa un ítem de información en la base de datos de DNS que puede ser consultado
- Un RR está definido por cinco campos
  - Class, Type, Value, Name, TTL

Resource Records: *"A" y "AAAA"*

- RR "A": *Address*
  - Los registros A establecen las correspondencias entre direcciones IP y nombres de dominio
  - Idem IPv6: "AAAA"
- Formato "Zone File" de BIND
  - `www IN A 200.7.85.220`
  - `www IN AAAA 2001:db8::2`

Especificacion del protocolo

- Formato de paquetes DNS
  - Header
    - Encabezado del protocolo
  - Question Section
    - La pregunta que hacemos al DNS
      - Tuplas (*Name, Type, Class)*
  - Answer Section
    - RRs que responden la pregunta (si es que hay), también en (N, T, C)
  - Authority Section
    - RRs que apuntan a una autoridad (opcional)
  - Additional Section
    - RRs que a juicio del DNS pueden ser útiles para quien está preguntando, y que pueden no ser autoritativos

Formato del paquete

- Cada sección es una lista de RRs
- Cada RR se identifica por:
  - Class
    - Identifica la aplicación, en Internet es siempre **IN**
  - Type
    - El tipo de RR
      - SOA, MX, A, AAAA etc.
  - Name
    - El nombre completo por el que estamos preguntando o por el que se está respondiendo
  - Value
    - El valor que el registro representa
  - TTL
    - Tiempo de vida del registro

DNS Header

- Campos
  - Query ID
    - Enlaza preguntas con sus respuestas
    - (recordar que uno de los transportes posibles es UDP)
  - OpCode: Operation code
  - Rcode: Response code
    - NOERROR, NXDOMAIN**, REFUSED, NOTAUTH
- Flags:
  - Dan informacion y proporcionan semántica
  - Flags usuales:
    - QR (query / response), AA (auth. answer), TC (truncation), RD (recursion desired), RA (recursion available)

DNS Header (II)

- Mas campos:
  - QDCount:
    - Numero de preguntas en Q.S.
  - ANCount
    - Numero de preguntas en A.S.
  - NSCount
    - Numero de RR en Auth. S.
  - ARCount
    - Numero de RR en Add. S.

Primarios / secundarios

- Primarios y secundarios
  - Cada zona tiene que tener *al menos* un servidor de nombres que sea *autoritativo* para ella
  - Este es el *primario* de la zona
  - Por motivos de redundancia, se recomienda tener uno o más servidores *secundarios* para la misma
    - Los secundarios también son autoritativos
- Transferencia de zonas
  - Para no tener que configurar la misma información dos o tres veces, y para facilitar la operación, existe un protocolo de transferencia de zonas (AXFR)

Consultas directas / reversas

- Terminología sobre consultas
  - Consultas directas
    - De nombre a dirección
      - A
  - Consultas reversas
    - De dirección a nombre
      - PTR
    - Cuidado con el concepto de “inversas”
      - Nunca estuvo bien definido y se abandonó

Consultas reversas

- El mapeo reverso permite asignar un nombre a una dirección IP
- Se mapean sobre dos dominios especiales y usando el RR “PTR”
  - in-addr.arpa (para IPv4)
  - ip6.arpa (para IPv6)
- Ejemplo (IPv4)
  - La consulta reversa por 200.2.13.45 se traduce en una consulta por el PTR de 45.13.2.200.in-addr.arpa

Transporte

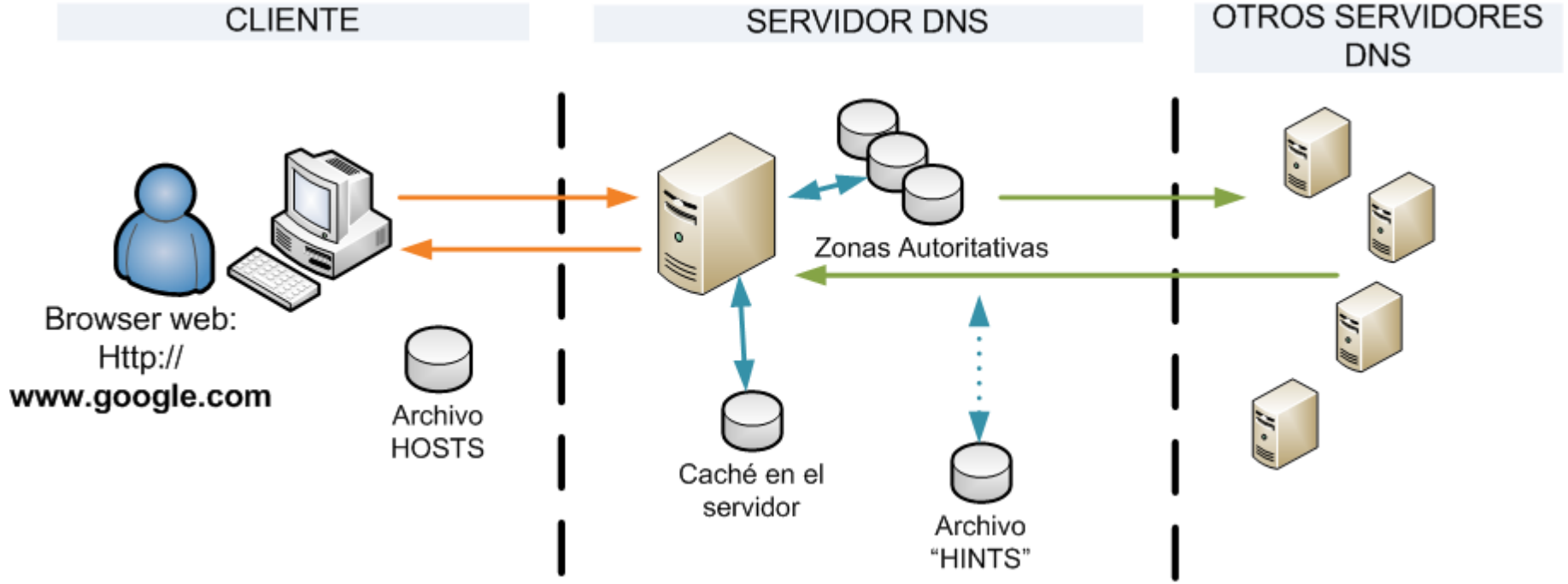
- Clientes y servidores pueden elegir entre TCP y UDP, los servidores DEBEN atender en ambos
  - UDP puerto 53
  - TCP puerto 53
- UDP:
  - El preferido por clientes finales
    - Las consultas pueden truncarse (MTU)
- TCP:
  - El preferido para las consultas recursivas servidor-servidor
    - Las consultas no se truncan
    - Existe la posibilidad de un failover de una a otra

Transporte (ii)

- IPv4 vs IPv6
  - Los paquetes de consultas y respuestas DNS pueden ser transportados indistintamente por IPv4 o por IPv6
- Las consultas que se pueden realizar son completamente independientes del transporte
  - TCP vs UDP
  - IPv4 vs IPv6

Operación: Consultas

- Esquema de una consulta DNS



"resolver" local

DNS recursivo local

Otros servidores autoritativos

Operación

- Esquema de una consulta DNS (II)
  - El PC final tiene un *resolver* local
    - Archivo /etc/hosts
      - Si aquí hay una entrada, se responde desde aquí
    - Apunta a un servidor DNS
  - Cada DNS trata de responder de:
    - Sus *hints*
    - Su caché
    - Sus zonas autoritativas
  - Cada DNS almacena en caché de forma agresiva todo los RRs que sean posibles
    - ¿Hasta cuando? Se guía por los tiempos establecidos en los registros SOA y en los TTLs

Archivo de zona directa

- En los archivos de zona directa encontramos mayormente registros A, AAAA y CNAME

```
@   IN SOA  www.example.org.   root.freebsd.www.example.org.
(
                        961230  ; Serial
                        3600    ; Refresh
                        300 ; Retry
                        3600000 ; Expire
                        3600  ) ; Minimum
    IN NS   freebsd.www.example.org.

; hosts solo v4
www.example.org.        IN A    10.0.0.10
nt2.www.example.org.    IN A    10.0.0.2

; host en dual stack
mail.example.org.       IN A    10.0.0.25
mail.example.org.       IN A    2001:db8:13c7::25
```

Archivo de zona reversa (v4)

- Las zonas reversas usan los registros PTR

```
@   IN SOA  X.Y.in-addr.arpa.  root.freebsd.www.example.org.
(
                        961230  ; Serial
                        3600    ; Refresh
                        300 ; Retry
                        3600000 ; Expire
                        3600  ) ; Minimum

        IN NS   freebsd.www.example.org.
10      IN PTR   freebsd.www.example.org.
1       IN PTR  nt1.www.example.org.
2       IN PTR  nt2.www.example.org.
```

Archivo de zona reversa (v6)

- Los reversos de IPv6 usan la dirección completa desarmada a nivel de nibbles

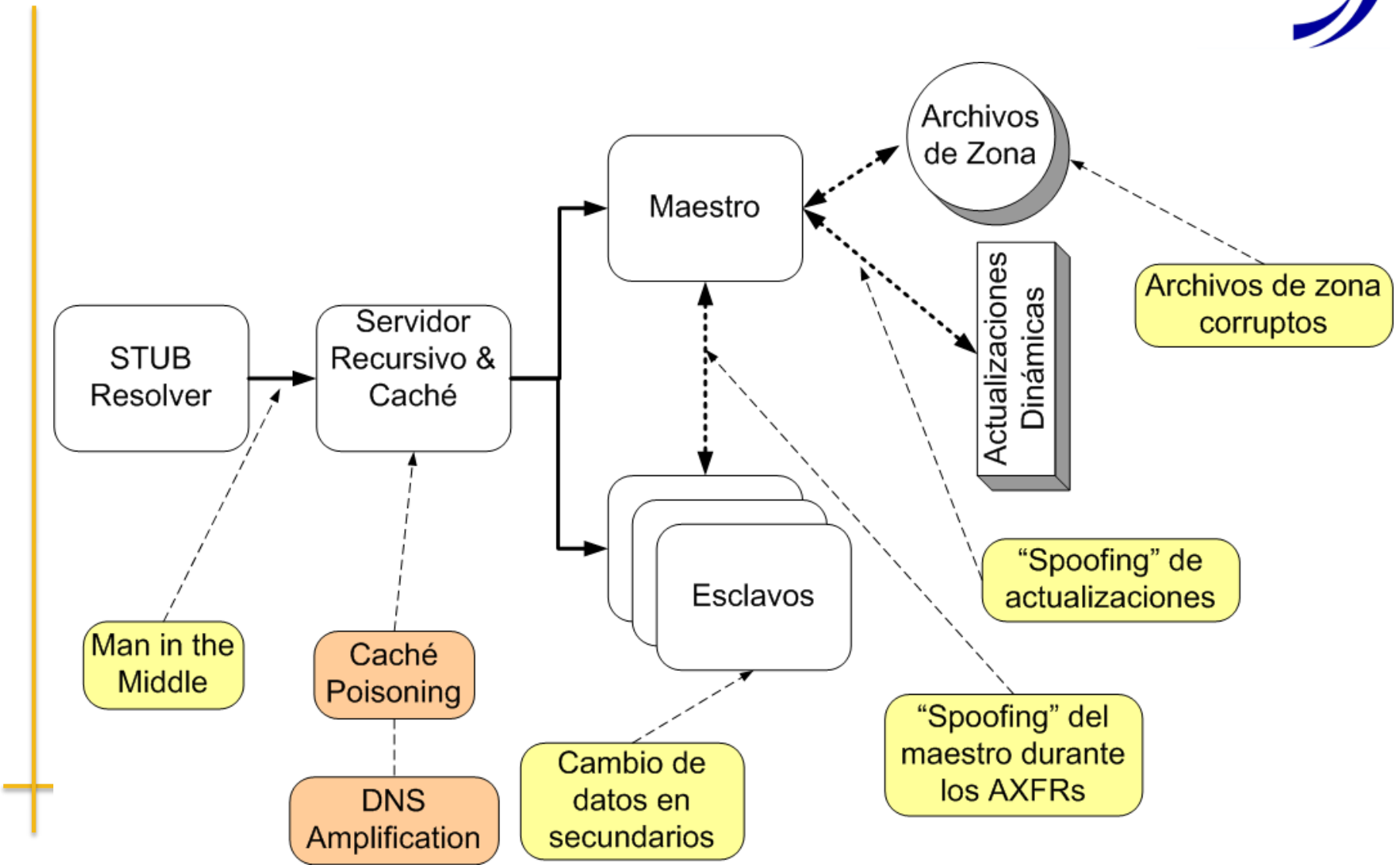
```
; reverse IPV6 zone file for example.com
$TTL 2d    ; default TTL for zone
$ORIGIN 0.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA.

@          IN      SOA    ns1.example.com. hostmaster.example.com. (
                          2010121500 ; sn = serial number
                          12h        ; refresh = refresh
                          15m        ; retry = refresh retry
                          3w         ; expiry = expiry
                          2h         ; nx = nxdomain ttl
                          )
; DNS servers
     IN       NS       ns1.example.com.
     IN       NS       ns2.example.net.
; PTR RR maps a IPv6 address to a host name

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN      PTR     ns1.example.com.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN      PTR     mail.example.com.
```

DNSSEC

Vectores de ataque en DNS



Vulnerabilidades del protocolo DNS

- La información transmitida en DNS puede ser "*spoofed*"
  - Entre maestro y esclavo (AXFR)
  - Entre maestro y sus clientes "*resolver*"
- Actualmente el protocolo DNS no permite validar la información contenida en una respuesta
  - Vulnerable a las diferentes técnicas de *poisoning*
  - Datos envenenados siguen causando problemas por un tiempo (potencialmente grande, TTL)
- Tampoco los secundarios tienen manera de autenticar al primario con el que están hablando

Introduciendo DNSSEC

- Análisis de vulnerabilidades en DNS
  - RFC 3833: "*Threat Analysis of the Domain Name System (DNS)*"
- DNSSEC:
  - "*DNS Security Extensions*"
  - RFC 4033, 4034, 4035
  - ~ Marzo 2005
    - Aunque DNSSEC viene siendo tratado desde hace mucho mas tiempo en el IETF

¿De que nos protege DNSSEC?

- DNSSEC nos protegerá de corrupción y del *spoofing* de datos
  - Proporciona un mecanismo para poder validar la autenticidad y la integridad de los datos contenidos en una zona DNS
    - DNSKEY/RRSIG/NSEC
  - Proporciona un mecanismo para delegar la confianza en ciertas claves públicas (cadena de confianza)
    - DS
  - Proporciona un mecanismo para autenticar las transferencias de zona entre primarios y secundarios
    - TSIG

Introducción a DNSSEC

- DNSSEC *no* es un nuevo protocolo
- Es un conjunto de **extensiones** al protocolo DNS tal como lo conocemos
  - Cambios en el "*wire protocol*" (EDNS0)
    - Extensión del tamaño máximo de una respuesta UDP de 512 a 4096 bytes
  - Agregado de nuevos *resource records*
    - RRSIG, DNSKEY, DS, NSEC
  - Agregado de nuevos flags
    - Checking Disabled (CD)
    - Authenticated Data (AD)

Introducción a DNSSEC (2)

- Nuevos RR
  - RRSIG: *Resource Record Signature*
  - DNSKEY: *DNS Public Key*
  - DS: *Delegation Signer*
  - NSEC: *Next Secure*
- Nuevos Flags:
  - AD: indica que la respuesta esta autenticada
  - CD: indica que no se realiza chequeo (deshabilitado)

Introducción a DNSSEC (3)

- (Repaso) Un *resource record* en DNS es una tupla de cinco valores
  - *(nombre, clase, tipo, TTL, valor)*
- El registro:
  - `www.empresa.com. 86400 IN A 200.40.100.141`
  - Esta representado por la tupla:
    - Nombre (www.empresa.com)
    - Clase (IN)
    - Tipo (A)
    - TTL (86400 segundos)
    - Valor (200.40.100.141)

Introducción a DNSSEC (4)

- *Resource Record Sets (*RRSets)
  - DNSSEC opera firmando *RRSet*s (no RR individuales)
  - Un RRSet es un conjunto de resource records que comparten igual:
    - Clase
    - Tipo
    - Nombre
- Ejemplo de RRSet (TTL omitido):
  - `www IN A 200.40.241.100`
  - `www IN A 200.40.241.101`

Introducción a DNSSEC (5)
Firma de zona

- Se genera un par de claves (publica y su correspondiente privada) para cada **zona**
  - El par de claves es propio de cada zona y no del servidor autoritativo
  - La parte privada se debe mantener bajo custodia
    - La privada firma los RRSets de la zona
  - La publica se debe publicar en DNS mediante un registro DNSKEY
    - La privada permite verificar las firmas de los RRSets
  - Un RRSet puede tener multiples firmas generadas con diferentes claves

Introducción a DNSSEC (6)

- La firma digital de un RRSet se devuelve en forma de un registro RRSIG que es parte de la respuesta
- Ejemplo:

```
~ carlosm$ dig +dnssec www.nic.se

;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1

;; ANSWER SECTION:
www.nic.se.       60   IN   A    212.247.7.218
www.nic.se.       60   IN   RRSIG    A 5 3 60 20101021132001
20101011132001 23369 nic.se. HeeUZ5h5iExK5uU1SuNRIf2Dbmh2/
aWV8FkjmzixUzTAVrHv39PfmfnG DHdHoZxoz85hqqYiWb
+t9EZh5+iqxQk8AxRDic9Nn6WxifOoWeS+IUKQ
rVyqXf1NtkZvu1A325vwa8obtbeVGVkhqg6bDIjKYeHixjlQ4cRoFcEW Izk=

;; AUTHORITY SECTION:
nic.se.           2974 IN   NS   ns3.nic.se.
nic.se.           2974 IN   NS   ns2.nic.se.
nic.se.           2974 IN   NS   ns.nic.se.
nic.se.           3600 IN   RRSIG    NS 5 2 3600 20101021132001
20101011132001 23369 nic.se. GSzAUC3SC3D0G/
iesCOPnVux8WkQx1dGbw491RatXz53b7SY0pQuyT1W
eb063Z62rtX7etynNcJwpKlYTG9FeMbDceD9af3KzTJHxq6B+Tpmmxyk
FoKAVaV0cHTcGUXSObFquGr5/03G79C/YHJmXw0bHun5ER5yrOtOLegU IAU=
```

Cadena de confianza

- ¿Como puede un cliente verificar un RRSet de una cierta zona?
  - Hace una consulta por el DNSKEY correspondiente
  - Realiza los calculos correspondientes y los compara con el RRSIG
    - Si coinciden, la firma verifica, de lo contrario, no
- Pero ¿como se puede confiar en la DNSKEY si sale de la misma zona que queremos verificar?
  - Necesitamos verificar la **cadena de confianza**

Cadena de confianza (ii)

- Registro DS "*Delegation Signature*"
  - Los registros DS "firman" claves de zonas **hijas**
  - De esta forma uno puede verificar el DNSKEY de una zona buscando un registro DS en la zona padre
- El registro DS contiene un hash de la una clave pública
  - Es decir, del contenido de un registro DNSKEY
- Los registors DS en la zona padre están firmados con la(s) claves de esa zona
- Para completar la cadena de confianza tiene que estar firmada la **raíz del DNS**

Cadena de confianza (iii)

- Pero ¿que pasa con la zona raíz?
  - La zona raíz no tiene "padre" a quien ir a pedirle un registro DS
  - La raíz del DNS esta firmada desde julio de 2010
    - *[http://www.root-dnssec.org]*
  - El registro DS para "." se puede obtener fuera de banda
    - [http://data.iana.org/root-anchors/root-anchors.xml]
    - . IN DS 49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5

¡Gracias!

carlos @ lacnic.net