



**agesic**

agencia de gobierno electrónico  
y sociedad de la información



# Seguridad en DNS y DNSSEC

Ciclo de charlas 2010 – CERTificate!

Santiago Paz

Nicolás Antoniello

Carlos Martínez-Cagnazzo

# Agenda

- Introducción
- Conceptos básicos y operación del DNS
- Vulnerabilidades del sistema DNS
- Kaminsky Bug 2008
- DNSSEC
- Referencias

Seguridad en DNS y DNSSEC

# INTRODUCCIÓN AL DNS

## Introducción (3)

- 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 (4)

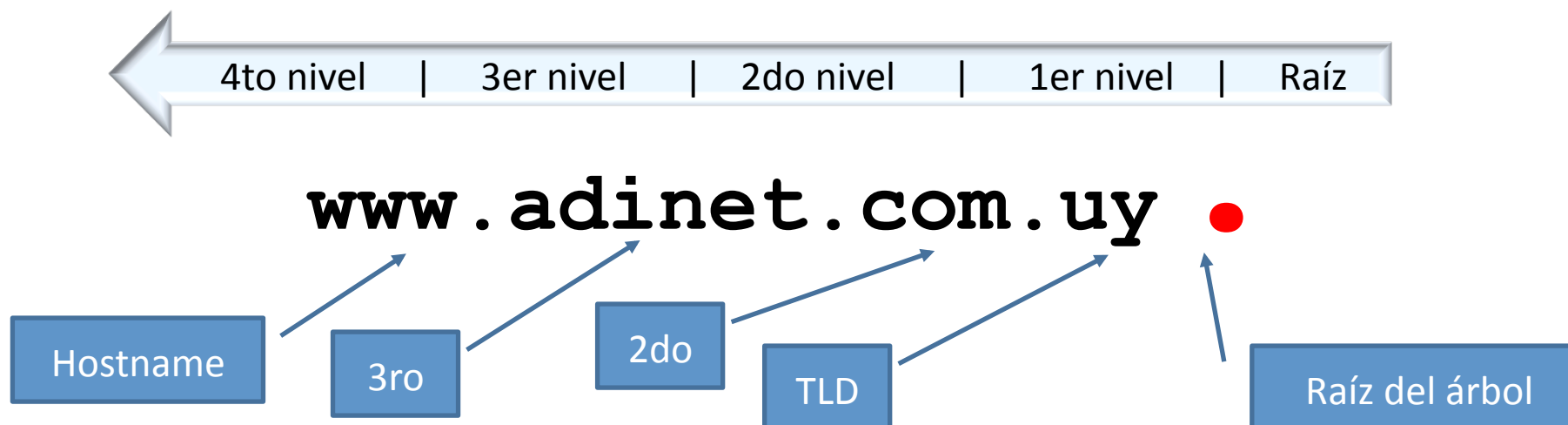
- Propiedades del sistema DNS:
  - Diferentes consultas y aplicaciones
    - Nombres directos, reversos, apoyo a aplicaciones, alias (CNAMEs)
  - Distribución de la administración
    - En Internet no hay administración centralizada sino que todo es por naturaleza distribuido. El DNS debe soportar y apoyar esta forma de trabajo.
  - Performance adecuada
    - Las consultas deben responderse lo mas rápidamente posible.
  - Confiabilidad adecuada
    - El DNS es obviamente una pieza crítica de la infraestructura de Internet, por lo que debe ser altamente confiable.

## 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 e inversa
    - *Cuidado: en Internet “resolución inversa” != “resolución 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 (4)

- Estructura de los nombres de dominio:



- Comentarios:
  - Los niveles del árbol reflejan las divisiones administrativas
  - El root del arbol esta siempre presente de forma ímplicita
  - No hay restricciones a la cantidad de niveles
  - Los niveles superiores “delegan” hacia los inferiores

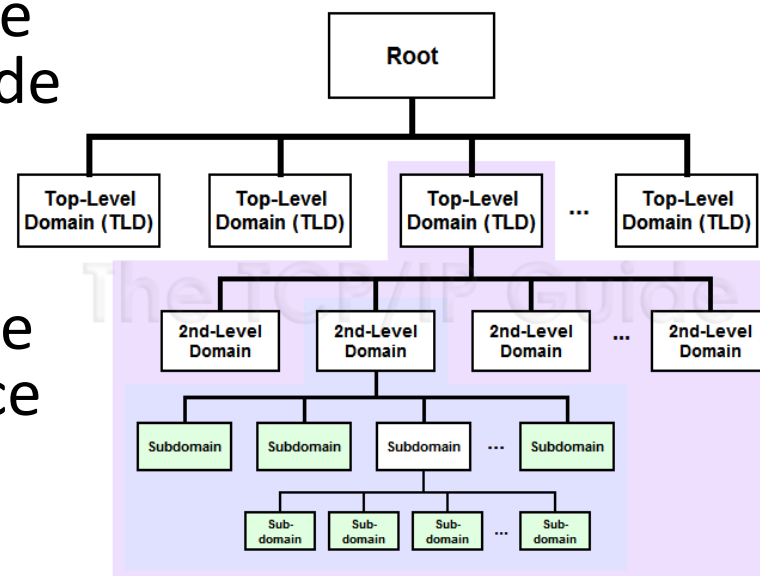
DNS: Doman Name System

# CONCEPTOS BÁSICOS Y OPERACIÓN



# Conceptos básicos

- 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
- Transporte
  - TCP y/o UDP puerto 53



## Conceptos básicos

- 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)

## Conceptos básicos: Resource Records

- Registros (*Resource Records*)
  - La información en la base de datos del DNS está estructurada en un conjunto de *resource records*:
    - SOA, A, CNAME NS, MX, PTR, TXT, etc.
  - Cada RR representa un ítem de información en la base de datos de DNS que puede ser consultado
  - **A, CNAME**: resolución directa; **PTR** resolución reversa
- SOA: “*Start of Authority*”
  - Delimita una zona
  - Incluye a todos los RRs de la misma

**SOA: [adinet.com.uy](http://adinet.com.uy)**

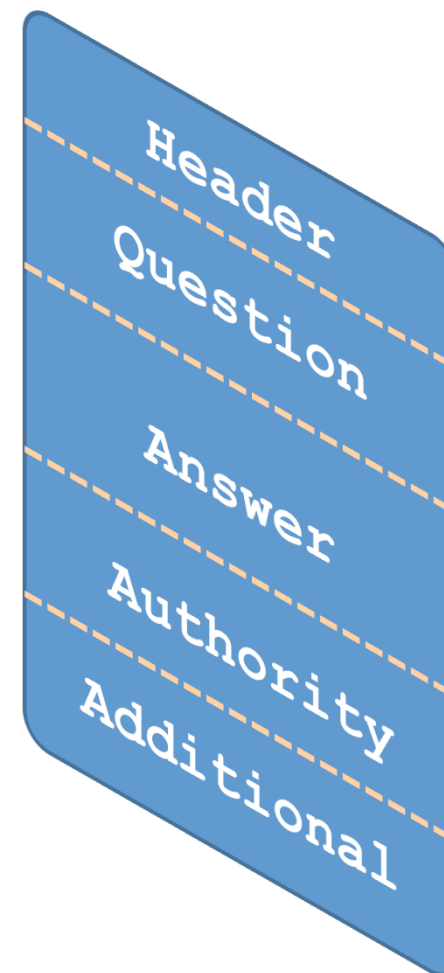
IN A ....

IN A ....

IN MX ...

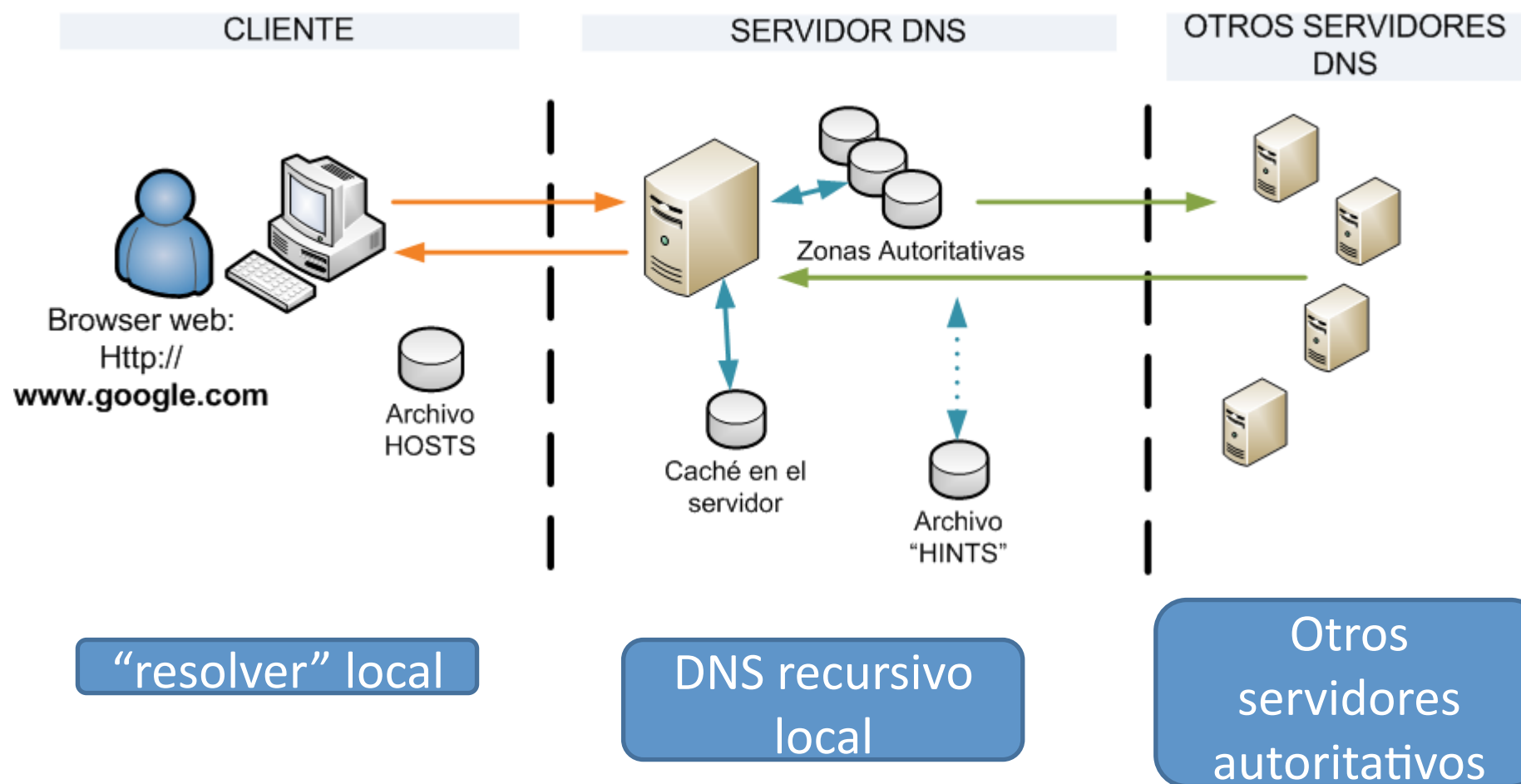
# Formato del paquete

- Formato de paquetes DNS – “Secciones”
  - Header
    - Encabezado del protocolo
    - Query ID
  - Question: pregunta realizada
  - Answer: respuesta directa a lo preguntado
  - Authority
    - Servidores autoritativos
  - Additional
    - Glue records: en que IP encontrar a los servidores autoritativos



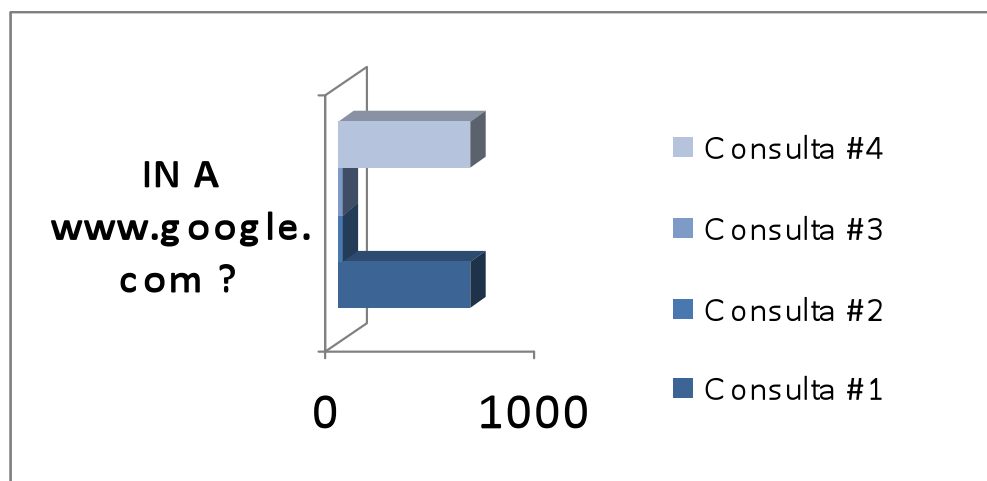
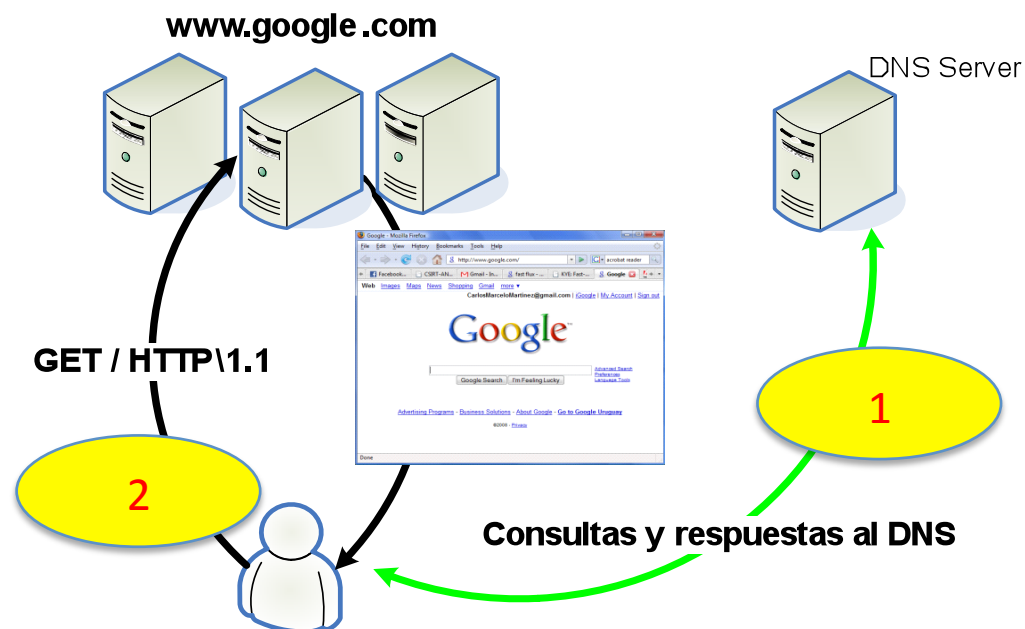
# Operación: Consultas

- Esquema de una consulta DNS



# Operación: *Time-to-Live*

- Cada consulta al DNS es “costosa”
  - Consulta a servidores remotos
  - Consultas recursiva
- Los resultados se almacenan en *caché* local
- ¿Por cuánto tiempo?  
*Time-to-Live*
- Típicamente
  - 86400 segundos (1 día)



# Operación: Root Servers

- ¿Como arranca el proceso? Buscando la *raíz*
- Root servers
  - Son una serie de servidores bien conocidos repartidos en el mundo
  - Todos los servidores DNS cuando uno los instala vienen un un *hint file* de los root servers
- ¿Como se sigue a la autoridad?
  - Cada zona puede delegar sub-zonas a otros servidores
  - *Glue records*
    - Son registros NS (*name server*) que apuntan a una sub-zona, realizando una delegación de autoridad

# Operación: Root Servers

- (Fuente: *Wikipedia*)

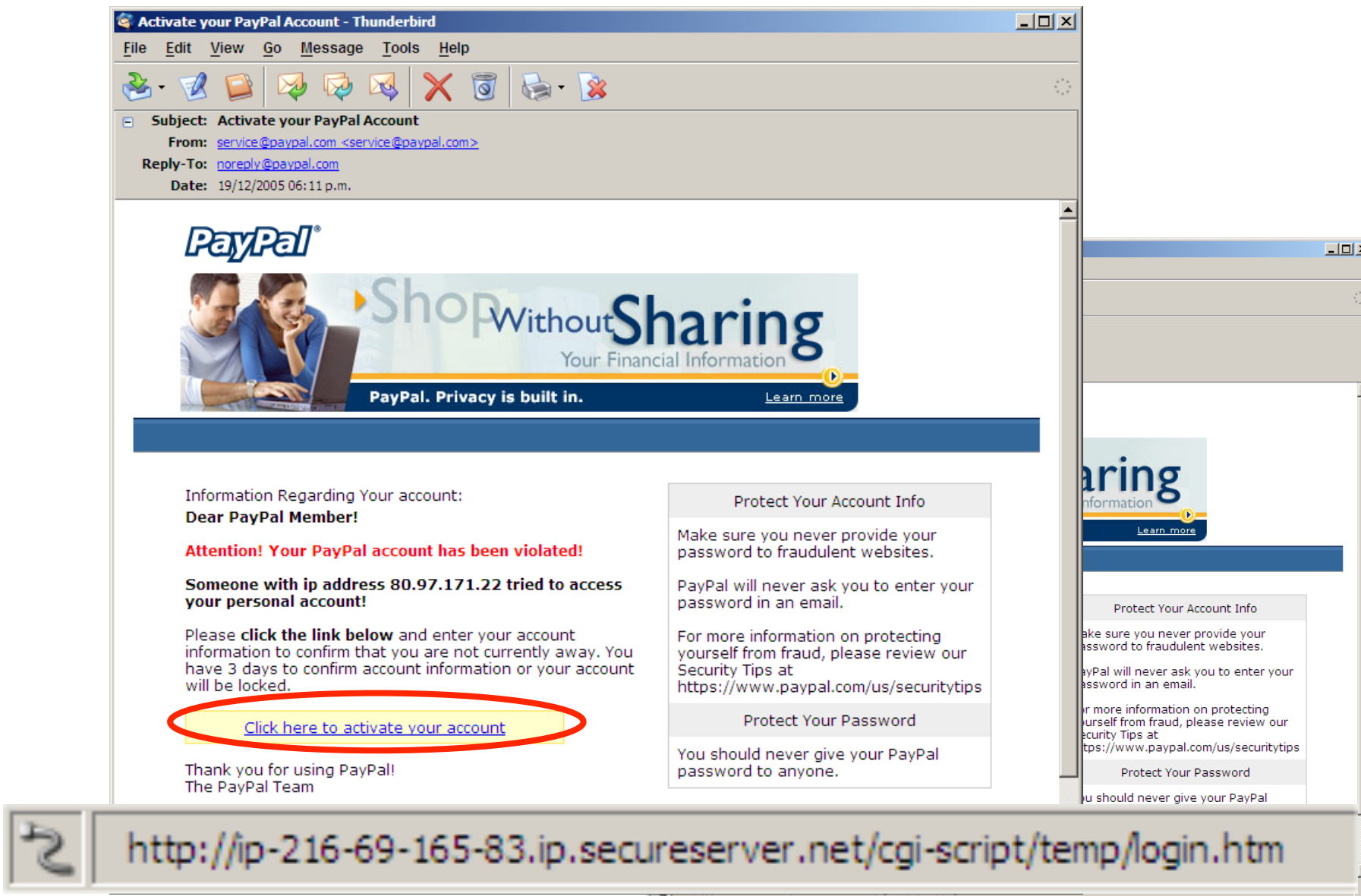




Seguridad en DNS y DNSSEC

# **ATAQUES Y VULNERABILIDADES EN DNS**

# ¿Porqué atacar el DNS?



**Activate your PayPal Account - Thunderbird**

File Edit View Go Message Tools Help

Subject: **Activate your PayPal Account**  
From: [service@paypal.com](mailto:service@paypal.com) <[service@paypal.com](mailto:service@paypal.com)>  
Reply-To: [noreply@paypal.com](mailto:noreply@paypal.com)  
Date: 19/12/2005 06:11 p.m.

**PayPal®**

**Shop Without Sharing**  
Your Financial Information  
PayPal. Privacy is built in. [Learn more](#)

Information Regarding Your account:  
**Dear PayPal Member!**

**Attention! Your PayPal account has been violated!**

**Someone with ip address 80.97.171.22 tried to access your personal account!**

Please **click the link below** and enter your account information to confirm that you are not currently away. You have 3 days to confirm account information or your account will be locked.

[Click here to activate your account](#)

Thank you for using PayPal!  
The PayPal Team

**Protect Your Account Info**

Make sure you never provide your password to fraudulent websites.

PayPal will never ask you to enter your password in an email.

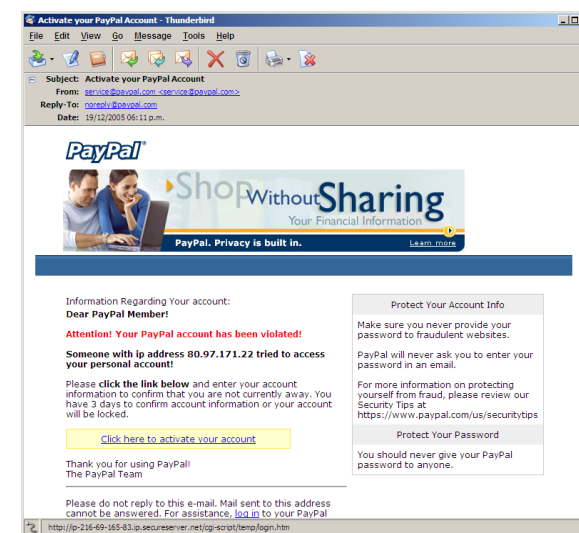
For more information on protecting yourself from fraud, please review our Security Tips at <https://www.paypal.com/us/securitytips>

**Protect Your Password**

You should never give your PayPal password to anyone.

<http://ip-216-69-165-83.ip.secureserver.net/cgi-script/temp/login.htm>

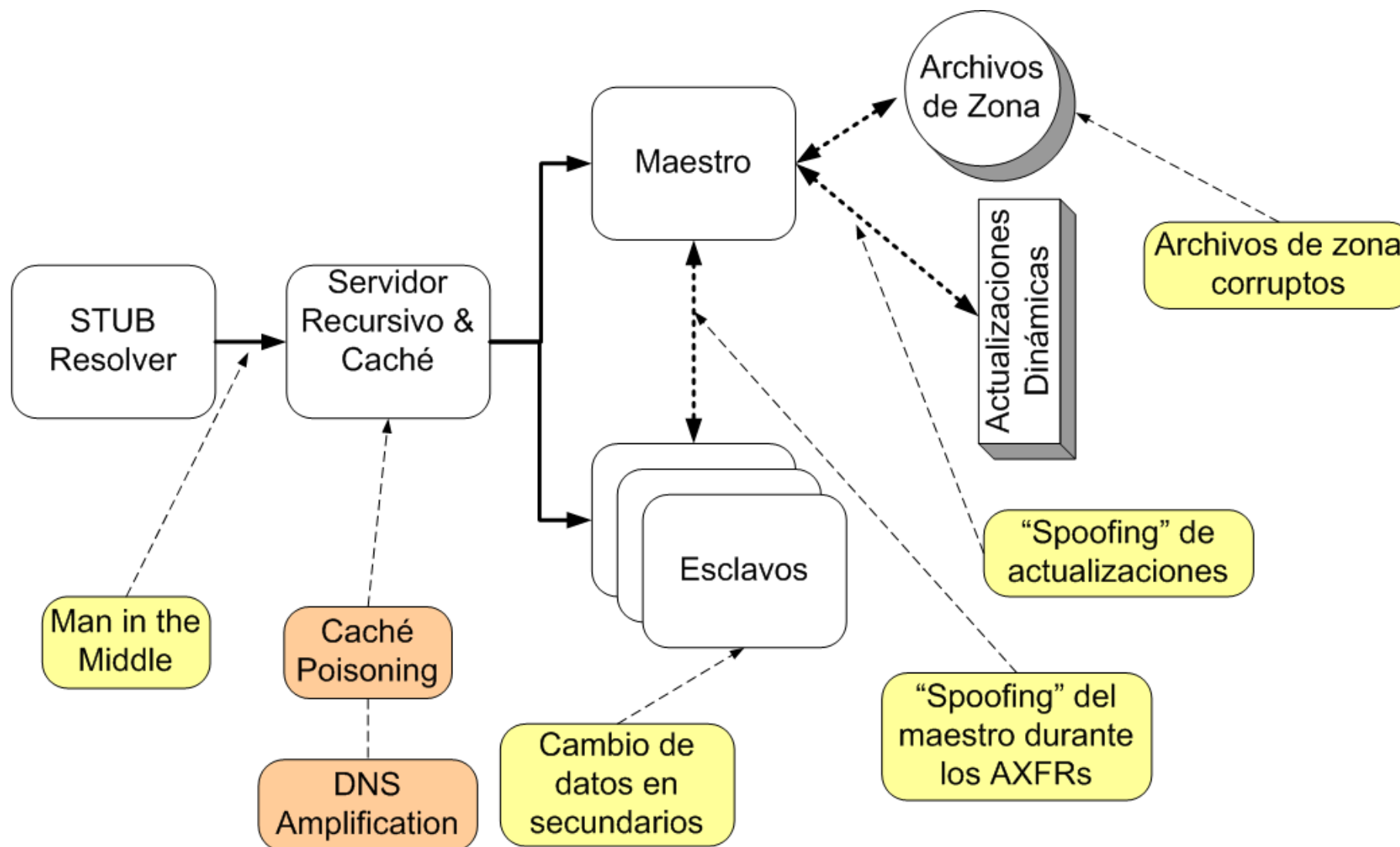
- Para que el *phishing* opere hacen falta:
  - Un sistema comprometido donde alojar las páginas web que simulan al sitio “real”
  - Una forma de direccionar (nombre o IP), para dirigir a los usuarios al mismo
    - En general, las IPs son variables, hacen falta nombres
  - Un agente de recolección de datos
- Rastros:
  - Artefactos en web servers comprometidos



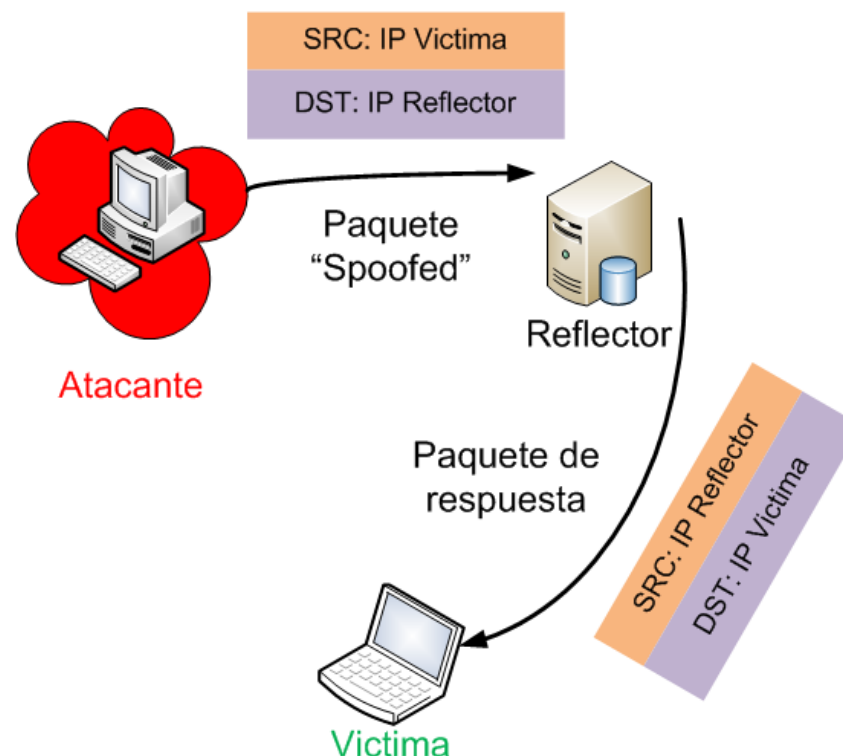
## (In) Seguridad en DNS

- Múltiples posibles vectores de ataque
- Aspectos principales:
  - Ataques con amplificación
  - Caché Poisoning
  - Aseguramiento de las transferencias de zona
  - Certificación de autoridad
- Aspectos no directamente relacionados con el protocolo
  - Vulnerabilidades en el software que implementa DNS

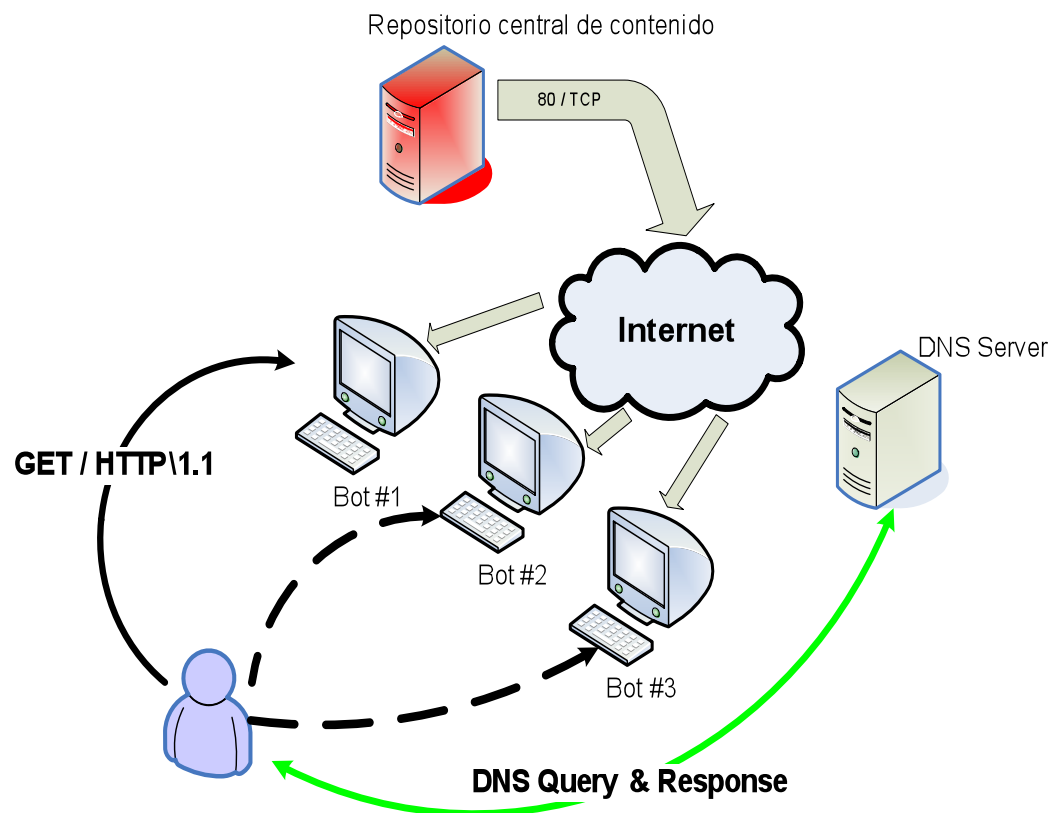
# Vectores de ataque en el DNS



- Vector -> servidores recursivos **abiertos**:
  - El atacante hace que el reflector almacene registros grandes en caché (TXT típicamente)
    - Simplemente instala un DNS que sirva esos registros y consulta a los DNS reflectores
  - Realizar las consultas *spoofed* con IP de origen la de la víctima
- Algún numero:
  - 20 servidores recursivos
  - Amplificación 100X
    - 20 bytes -> 2 Kbytes
  - ADSL de 512 Kbps
  - Tráfico de ataque: **1 Gbps**
    - $512 \text{ Kbps} * 100 * 20$



- ¡Alta disponibilidad para botnets!
  - Múltiples registros “A” devueltos por el DNS
  - TTLs muy pequeños
  - Los “servidores” son en general computadores personales comprometidos
  - Registros “A” van cambiando con el tiempo
  - Los propios bots actúan como DNS servers
- Registros NS
  - Pocos y estáticos (single flux)
  - Muchos y dinámicos (double flux)



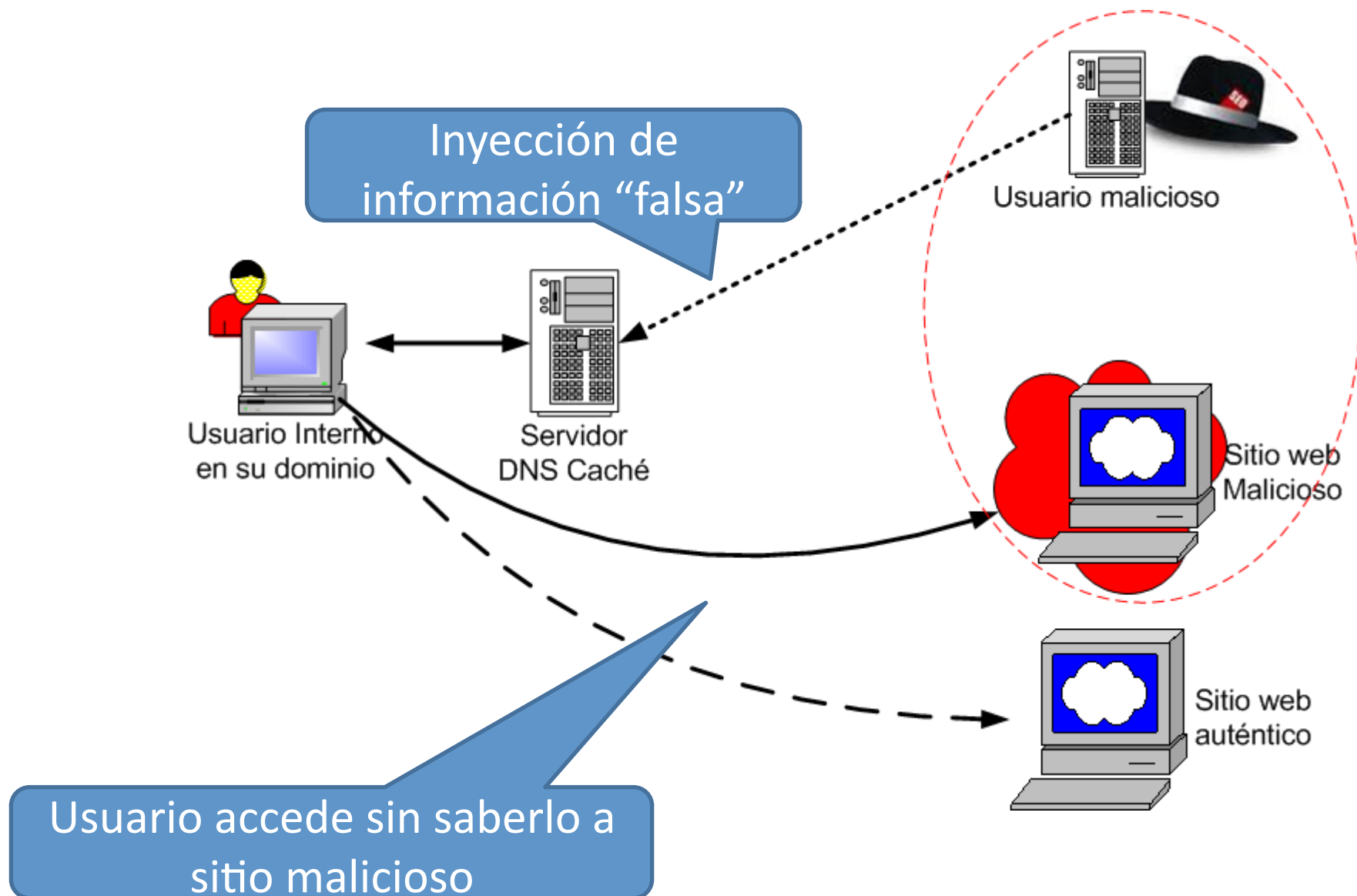
- El objetivo es mantener la alta disponibilidad de la red de máquinas comprometidas

# Seguridad en DNS: *Caché Poisoning*

- El *caché poisoning* es una técnica por la cual es posible engañar a un servidor DNS y hacerle creer que recibió información auténtica y válida
- El servidor luego cachea esa información y la utiliza para responder otras consultas hasta la duración el TTL de los RRs cacheados
- De esta forma propaga el engaño aguas abajo
- ¿Para qué?
  - Redirigir tráfico a sitios tomados, *pharming*
  - Robo de información



# Caché Poisoning (II)

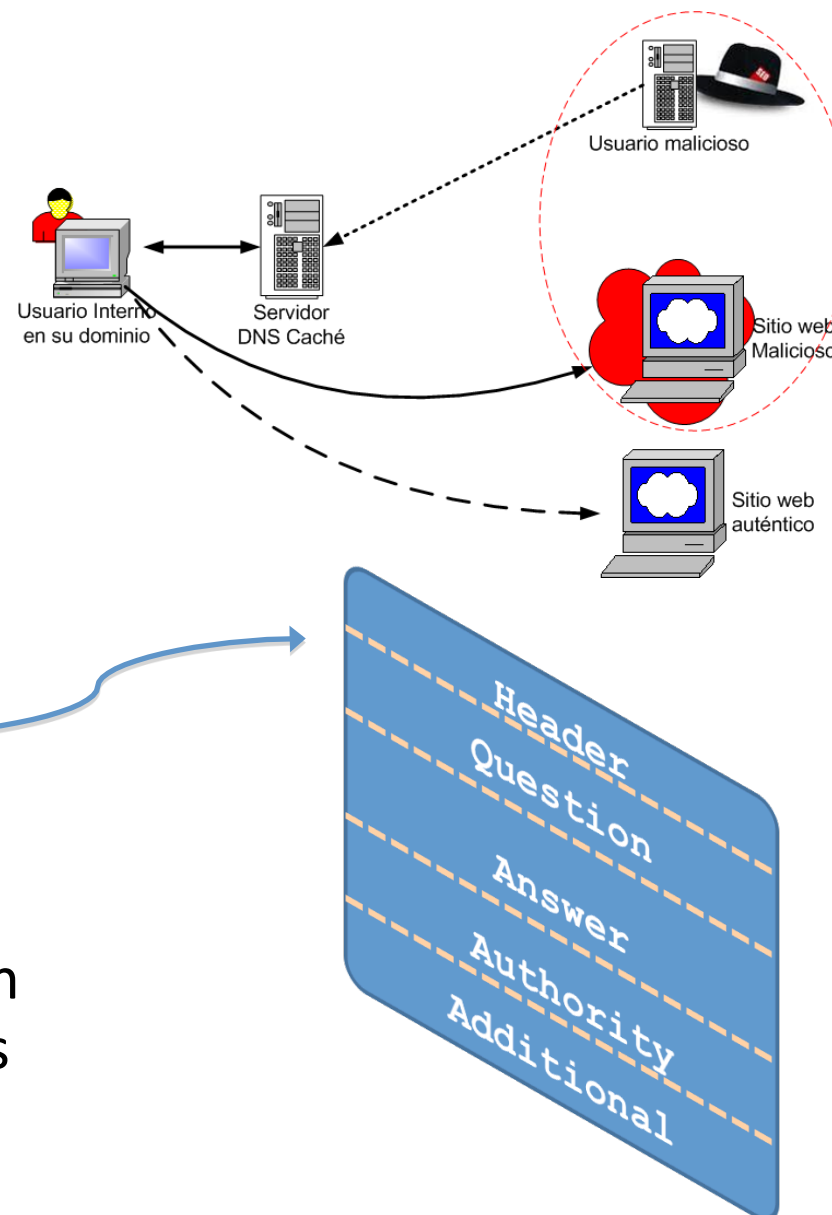


## *Caché Poisoning (III)*

- ¿Cómo?
  - Recordar que los servers cachean agresivamente la sección *Additional* de la respuesta
  - Truco: devolver el engaño en esta sección
    - El atacante debe tener un DNS server bajo su dominio, con una zona autoritativa.
- Ejemplo:
  - Un cliente pregunta al DNS de good.org por el MX de bad.org
  - La respuesta trae el MX de bad.org y además trae, por ejemplo:
    - **ns.banco.com.uy IN A X.Y.Z.W**
    - X.Y.Z.W es la IP del DNS de bad.org o cualquier otro equipo malicioso
- Esta es la modalidad clasica de envenenamiento; en general ya no funciona

# Caché Poisoning (IV)

- *DNS packet forgery*
  - Inyectar paquetes de respuesta antes que el verdadero servidor de nombres en una consulta recursiva
  - Implica adivinar un QueryID de 16 bits
    - Es lo que se usa para atar preguntas con respuestas
    - El puerto de origen UDP en muchas implementaciones es muy fácil de adivinar

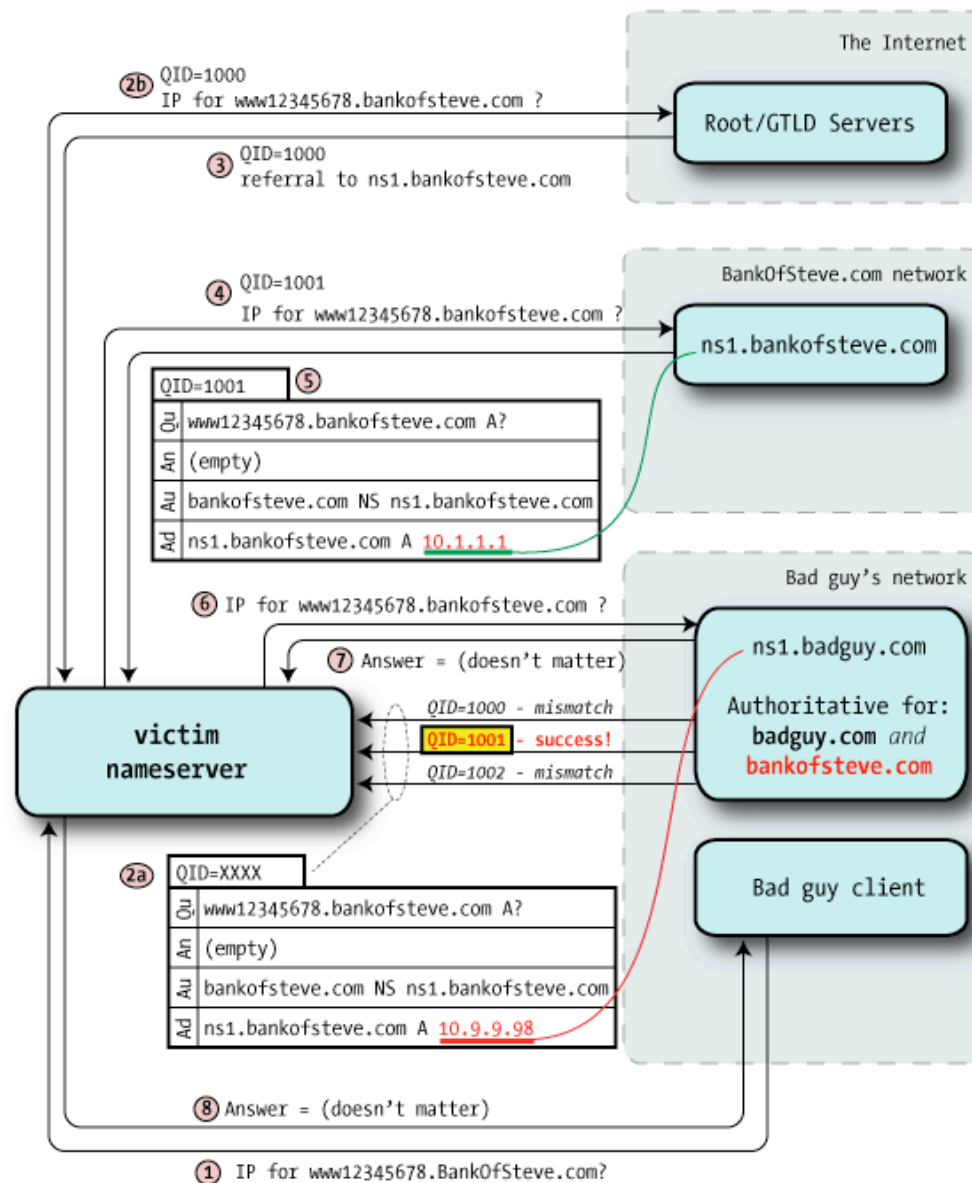


# Kaminsky Bug 2008

- Premisa: *“envenenar” la autoridad*
  - Si en vez de envenenar registros “A” individuales, envenenamos los “NS” de una red podemos redirigir las consultas a un DNS controlado por nosotros.
  - Alguien se anima a especular sobre las consecuencias de algo así? Variadas!
- El mecanismo es similar al *“packet forgery”* que vimos antes
  - Pero permite el control de la zona completa
- Raiz del problema:
  - Facilidad de adivinar dos números de 16 bits (queryID y src port)

# Kaminsky Bug 2008 (II)

- Fuente: “*An Illustrated Guide to the Kaminsky DNS Vulnerability*”
- <http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html>



Seguridad en DNS y DNSSEC

# DNSSEC

# DNSSEC

- ¿Qué es DNSSEC? *Domain Name System Security Extensions*
- Objetivo principal: proteger al sistema de DNS de inyección de datos falsificados
- Provee *firmas digitales* de las respuestas positivas y negativas



# DNSSEC

- ¿Qué es DNSSEC?

Las Extensiones de Seguridad para el Sistema de Nombre de Dominios (DNSSEC) consiste en un conjunto de protocolos desarrollados por la IETF para securizar cierto tipo de información parte del servicio del Sistema de Nombres de Dominio (DNS).

- Este conjunto de extensiones, provee a los clientes DNS (resolvers) de la posibilidad de **validar la autenticidad de quien origina la respuesta** a una consulta DNS, **indicación autenticada de que la no existencia de información** para los datos solicitados y de **integridad de los datos** transferidos.



# DNSSEC

- ¿Qué NO es DNSSEC?

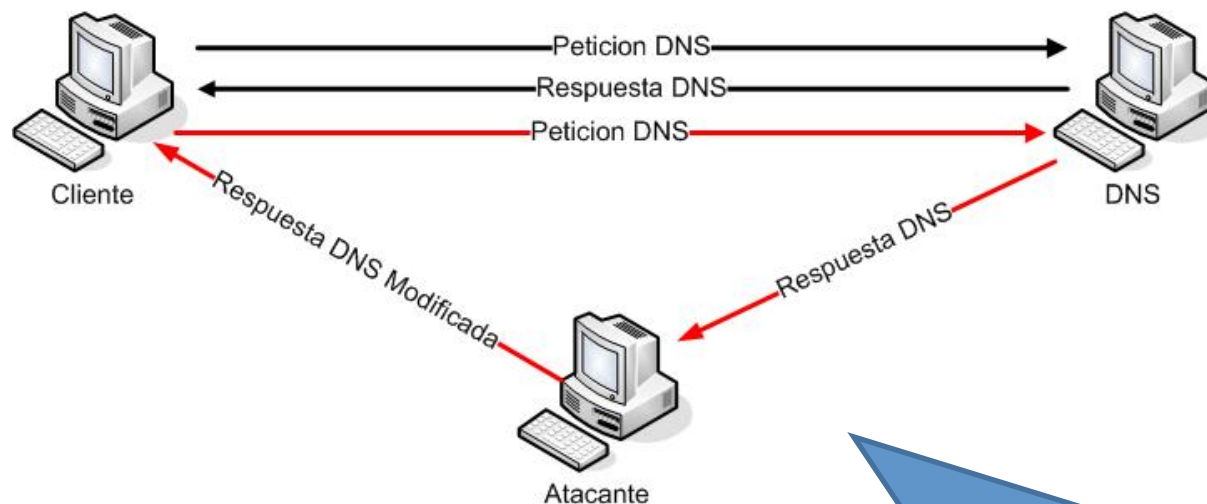
Es importante recordar que DNSSEC no incrementa ni provee disponibilidad o alta disponibilidad del sistema DNS ni provee confidencialidad en la información intercambiada; esto último en el entendido de que los mensajes intercambiados entre los clientes (resolvers) y los servidores (servers) no es encriptada por el protocolo DNSSEC.



DNSSEC no busca ser la solución robusta y definitiva para todos los problemas de seguridad y posibles ataques al Sistema de Nombre de Dominios (DNS)

# DNSSEC

- *Validar la autenticidad de quien origina la respuesta a una consulta DNS e integridad de los datos transferidos*



Notar que DNSSEC no elimina la posibilidad de inyectar información falsa en una transacción de DNS, sino que proporciona un mecanismo para autenticar el mensaje de respuesta transferido

# DNSSEC

- *Indicación autenticada de la no existencia de información para los datos solicitados*



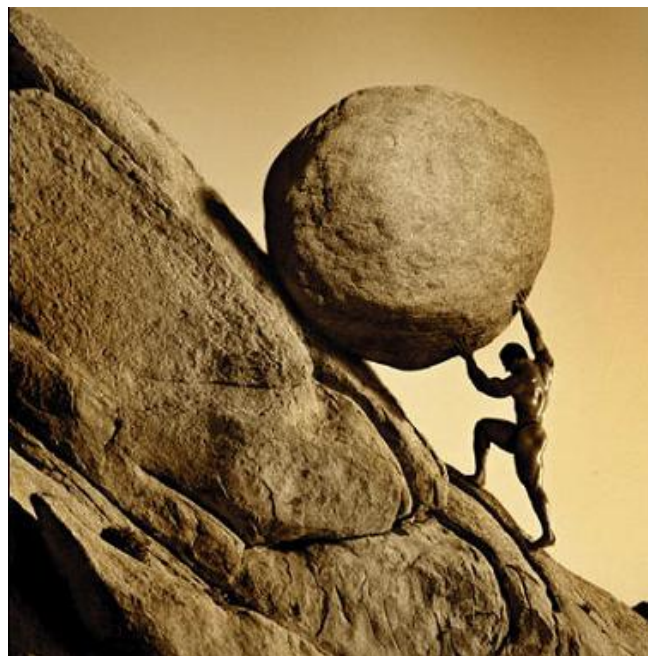
Seguridad en DNS y DNSSEC

# Rem> **CRIPTOGRAFÍA**

# Criptografía

- La piedra en el camino...

Problema general: lograr intercambiar información entre dos partes que están separadas físicamente y que eventualmente no confían entre si



# Criptografía

- El problema mejor detallado:
  - ¿Cómo se yo que el mensaje que recibí lo enviaste tú?
    - *autenticación*
  - Suponiendo que tú lo enviaste, ¿Cómo se yo, que ese mensaje no fue interceptado y alterado por un tercero, y luego reenviado hacia mi?
    - *integridad*
  - Nuevamente suponiendo que tú lo enviaste, ¿Cómo puedo estar seguro de que tú no me dirás luego?: – ¿Yo? Yo no te envié eso; jeso debió enviarlo alguien haciéndose pasar por mi!
    - *No repudio*



# Criptografía

- Surge el problema...
- Y finalmente, aunque justamente como se dijo anteriormente, la siguiente problemática no es contemplada por DNSSEC:
  - Si se trata de un intercambio de información privado: ¿Cómo evitar que un tercero que captura los datos en tránsito pueda conocer la información contenida en el mensaje?
  - O visto desde el otro extremo de la comunicación ¿cómo puedes estar seguro tú, que un mensaje dirigido exclusivamente a mí, solo sea posible ser leído por mí, aún cuando sea interceptado por un tercero?



# Criptografía

## ¿De que se trata eso de Criptografía de Clave Privada?

Uno de los mecanismos ancestrales para resolver la mencionada problemática es lo que denominamos comúnmente ***El Secreto***.

Ese Secreto, se trata de algo que en principio solo tú y yo sabemos, de algo que solo tú y yo tenemos (o incluso de algo que solo tú y yo somos).



En este caso, hablaremos de dos tipos de Secretos, uno que compartiremos y otro que ni siquiera compartiremos entre nosotros.



# Criptografía

## Secreto Compartido (Shared Key)



Tomaremos una caja de seguridad (que supondremos construida de un material indestructible e impenetrable) en la que pondremos tanto el mensajes a enviar, como también nuestro nombre y datos que nos identifiquen como el origen de la información. Todo eso dentro de la caja.

Luego fabricaremos solamente dos llaves que pueden abrir dicha caja; yo me guardaré una y tú la otra.

Esto es lo que se conoce en criptografía como mecanismo de ***clave compartida***, donde la *clave* que se comparte entre las partes será la llave; la caja de seguridad se implementa con *algoritmos matemáticos de cifrado* más o menos complejos, cuya cerradura solo se puede abrir si se posee la *clave*.

# Criptografía

## Secreto NO Compartido (Private/Public Key)



Ahora tomaremos una nueva caja de seguridad con las mismas características que la anterior excepto por la cerradura.

En este caso, también tenemos dos llaves, una para ti y otra para mí, pero la tuya solamente sirve para cerrar la caja y una vez cerrada, necesitas mi llave para poder abrirla, pues la mía solo sirve para abrir la caja.

Esto es ***casi*** lo que se conoce en criptografía como mecanismo de ***clave pública y clave privada***, donde el par de llaves implementan una la *clave pública* (la llave que cierra la caja) y otra la *clave privada* (la llave que abre la caja), y el secreto implementado por el par de llaves, ***no*** es compartido.

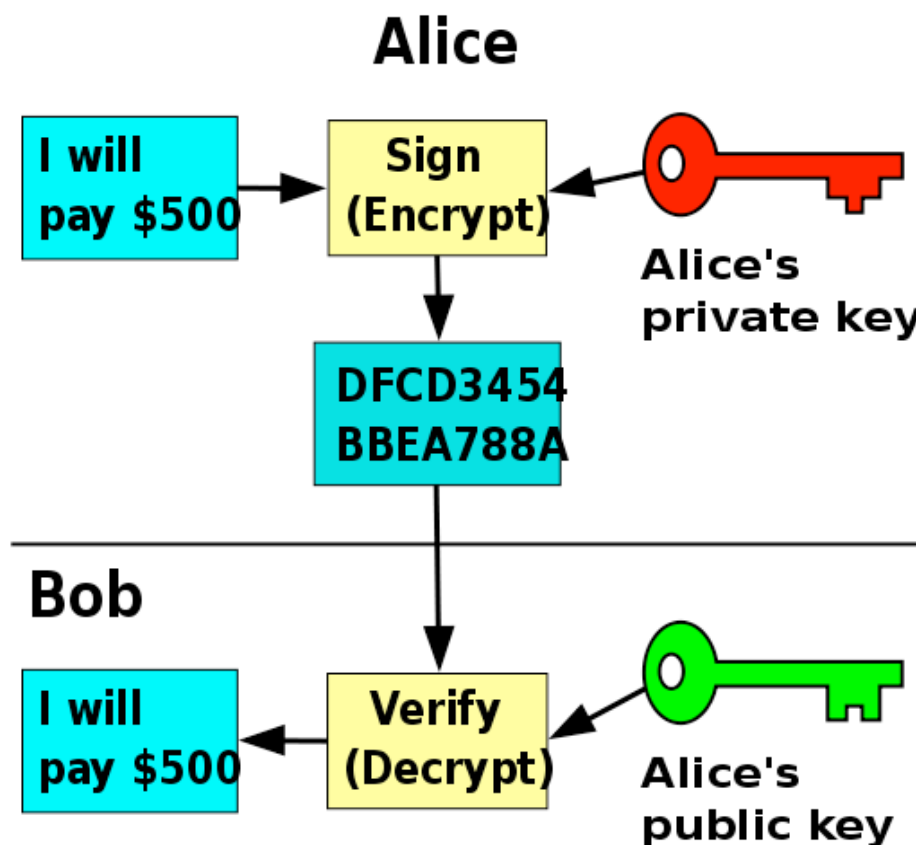
# Criptografía

## Firma digital (Private/Public Key)



Para completar el esquema, ahora agregamos la siguiente funcionalidad al par de llaves:

De esta forma, al mecanismo de *clave pública y clave privada* que teníamos, le añadimos la posibilidad de “*firmar*” los datos en los dos sentidos.



# Criptografía

## Cadena de confianza...



Finalmente, solo resta un aspecto por resolver:

¿Cómo se realiza la entrega de las llaves, de forma que ambos sepamos que quien tiene una en su poder, es realmente quien debe poseerla, cuando no podemos entregarlas de forma presencial?

Lo que podemos hacer es establecer una secuencia de pasaje de información, mediante terceros en los que confiamos, donde el intercambio entre cada uno de los participantes se realiza de forma segura, utilizando los mecanismos mencionados.



Esto es lo que denominamos comúnmente “cadena de confianza”.

Seguridad en DNS y DNSSEC

**¡ Más DNSSEC !**

# DNSSEC

## ¿Qué implica DNSSEC para un administrador de Zona?

Firmar los RRSets utilizando la clave privada (una clave para cada Zona)

Publicar dichas firmas para cada RRSets en el archivo de Zona correspondiente

Publicar la clave pública de la Zona en el archivo de Zona



# DNSSEC

¿Qué implica DNSSEC para un administrador de Zona?

Adicionalmente se deberá tener la clave pública de cada Zona, firmada por la Zona del nivel inmediatamente superior.  
De esta forma se asegura mantener la  
***cadena de confianza.***



# DNSSEC

## ¿Qué implica DNSSEC para un cliente DNS?

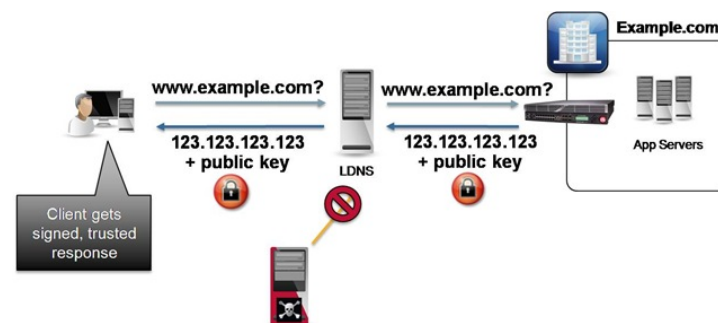
Capacidad de poder verificar la firma contenida en las respuestas DNS





# DNSSEC

## ¿Cómo logra esto DNSSEC?



Se añaden los siguientes registros:

- DNSKEY: Guarda la clave pública con la que se firman las Zonas.
- RRSIG: Guarda un Hash encriptado del RRSset (encriptado con la clave privada de la Zona).
- NSEC: Guarda una respuesta para los casos en los que el nombre requerido o el RR no existan en el archivo de Zona.
- DS: Contiene el Hash de la clave pública de la zona hija firmado por la clave privada del padre.

# DNSSEC

## Como seguir...

- Detalle de los nuevos registros de DNSSEC y su funcionamiento
- Ejemplo de funcionamiento de una consulta DNSSEC
- Configuración de un Servidor DNS y Cliente DNS con soporte DNSSEC



## Consideraciones...

- ¿UDP o TCP para consultas/respuestas DNS?
- ¿Qué sucede con el tamaño de los mensajes de respuesta DNS?
- ¿Nuevos vectores de ataque?
- Tiempos de consulta y tamaño de los archivos de Zona
- ¿Cómo se firma la raíz y cómo se asegura la cadena de confianza?
- ¿Rotación de claves?

# PREGUNTAS



Seguridad en DNS y DNSSEC

**¡GRACIAS POR SU ATENCIÓN!**