Administración de DNS

Alex Callejas

Marzo 2014

Agenda
Día 1

- Presentación
- Introducción al DNS
- Conceptos básicos de redes y criptografía
- Teoría básica del DNS

Presentación

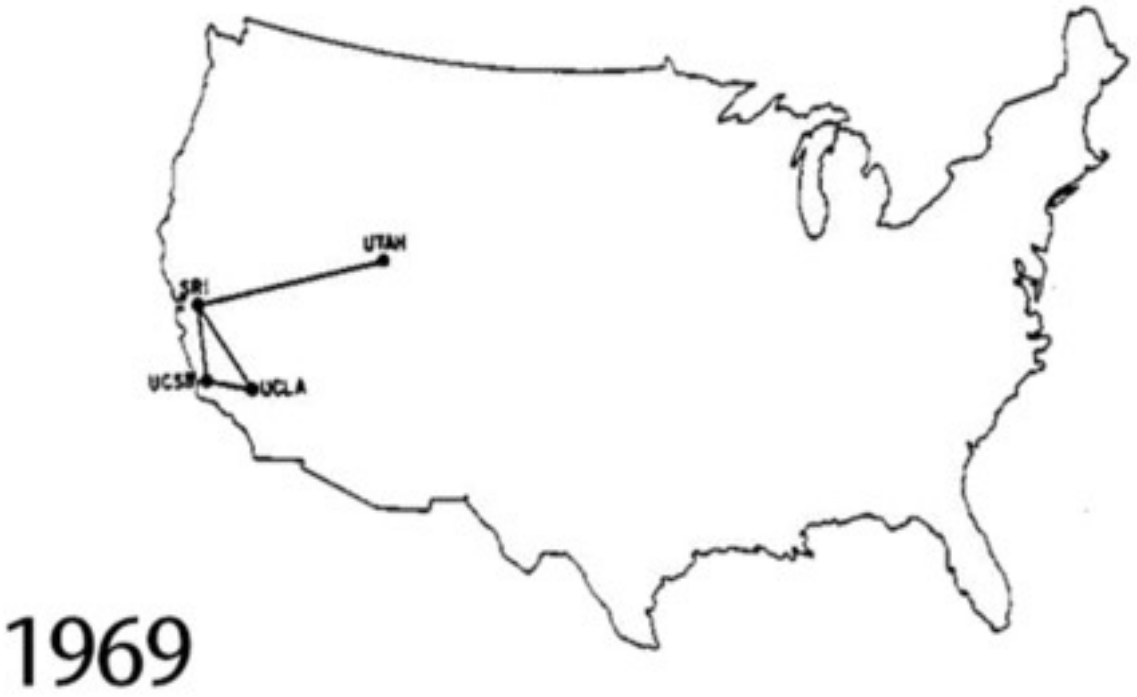
Introducción al DNS

- Breve historia de Internet
- Historia del DNS
- Estructura y características del DNS
- Historia de BIND

Arpanet

- A finales de 1960, la *Advanced Research Projects Agency* (ARPA) del Departamento de Defensa de USA creó una red experimental de computadoras denominada ARPANET
- A principios de 1980 fue desarrollada la suite de protocolos TCP/IP que rápidamente se convirtió en el protocolo estándar para la red ARPANET
- Al incluirse TCP/IP en el BSD Unix, muchas organizaciones que ya estaban conectadas a ARPANET conectaron también sus redes LAN para comunicarse con otras redes a través de ARPANET
- La red ARPANET fue creciendo hasta que se convirtió en la columna vertebral (backbone) de un conjunto de redes locales basadas en TCP/IP (lo que ahora conocemos como Internet)

Arpanet



Internet

- En 1988 el Departamento de Defensa decidió que la red experimental había llegado a su fin, así que empezó a desmantelar la infraestructura de ARPANET. Así fue como la red de la Fundación Nacional de la Ciencia (NSFNET) reemplazó a ARPANET como el backbone de Internet
- En 1995 hubo una transición más en Internet: comenzaron a usarse backbones comerciales, además del de NSFNET. El resto es historia …

Historia del DNS

- Durante 1970, la red ARPANET estaba conformada por una pequeña comunidad de computadoras. Había un único archivo denominado hosts.txt que contenía pares de nombre de computadora y su correspondiente dirección IP
- Este archivo era mantenido por el NIC del Instituto de Investigación de Stanford (SRI) y distribuido desde un punto central: la computadora sri-nic. Los administradores de ARPANET simplemente enviaban por correo electrónico sus cambios al NIC y periódicamente transferían vía FTP el archivo hosts.txt, el cuál se actualizaba una o dos veces a la semana.
- Aunque todo parecía ir muy bien, los problemas comenzaron conforme la red fue creciendo, pues el tamaño del archivo hosts.txt fue creciendo también, lo cuál a su vez provocó que las actualizaciones generaran cada vez más tráfico en la red. Peor aún, cuando ARPANET adoptó TCP/IP como su protocolo de comunicación y la red fue creciendo aún más, surgieron los siguientes problemas:
  - El tráfico de la red y la carga del procesador de la máquina sri-nic se volvió muy pesado
  - Aunque era imposible que dos computadoras tuvieran el mismo nombre, no podía evitarse la duplicidad en los nombres
  - Cada vez era más difícil mantener la consistencia del archivo hosts.txt, pues la red crecía más rápido de lo que este archivo podía actualizarse
  - Debido a estos problemas, los creadores de ARPANET comenzaron a pensar en un reemplazo para hosts.txt.

Historia del DNS

- Debido a estos problemas, los creadores de ARPANET comenzaron a pensar en un reemplazo para hosts.txt. Su idea era crear un sistema que permitiera:
  - Resolver los problemas inherentes a un sistema unificado
  - Administrar localmente los datos, pero al mismo tiempo hacerlos disponibles a todos
  - Asegurar la unicidad de los nombres

- El responsable de diseñar la arquitectura del nuevo sistema fue Paul Mockapetris, del Instituto de Ciencias de la Información de la Universidad del Sur de California. En 1984, liberó los RFCs 882 y 883, que describen el Sistema de Nombres de Dominio o DNS (Domain Name System). Estos RFCs fueron reemplazados más tarde por el RFC 1034 y 1025, respectivamente. Estos RFCs, a su vez, han sido aumentados en los RFCs 1535 (problemas potenciales de seguridad en el DNS), 1536 (problemas de implantación) y 1537 (aspectos administrativos)

Estructura y Características del DNS

- El DNS es una **base de datos distribuida de forma global, jerárquica, y redundante**
- Es *distribuida y global* porque se encuentra repartida en servidores de nombres de todo el mundo, lo cuál permite controlar de forma local los segmentos que conforman la base de datos completa, y al mismo tiempo los datos de cada segmento están disponibles en todo el mundo gracias al esquema cliente-servidor en que se basa el DNS
- Es *jerárquica* desde el punto de vista de su estructura
- Es *redundante* porque la misma información se encuentra en varios servidores repartidos en todo el mundo. Precisamente ahí radica la complejidad del DNS: está completamente distribuida en todo el mundo, en millones de computadoras administradas por otros tantos millones de personas, y aún así debe comportarse como una base de datos única e integrada
- El DNS es una base de datos única en el mundo por el número de peticiones que recibe cada segundo y por el número de actualizaciones que sufre todos los días

Historia de BIND

- La primera implantación del DNS se llamó JEEVES, escrita por Paul Mockapetris. BIND fue una implantación posterior
- BIND (Berkeley Internet Name Domain) fue escrito por Kevin Dunlap para el sistema operativo Unix 4.3 BSD de la Universidad de Berkeley
- Actualmente, BIND es mantenido por Paul Vixie, auspiciado por el ISC (Internet Software Consortium) cuyo WebSite es http://www.isc.org
- BIND se encuentra disponible para múltiples plataformas, incluyendo las basadas en Unix y Windows
- Actualmente se encuentran en uso básicamente dos ramas de BIND: BINDv8 y BINDv9 (la rama BINDv8 se considera obsoleta y debiera ser migrada a BINDv9 a la brevedad posible)
- Las últimas versiones de BIND al momento de escribir este material son:
  - BIND 9.9.5 (current release)
  - BIND 9.10.0a1 (development)

Conceptos Básicos de redes y criptografía

- Concepto de red
- Nombres y direcciones IP
- Protocolos DNS y ARP
- Concepto de Criptografía
- Criptografía de llave simétrica
- Criptografía de llave pública
- Funciones hash
- Firma digital

Conceptos básicos de redes

- Una red es un conjunto de computadoras interconectadas entre sí con el propósito de intercambiar información y compartir recursos de hardware (como una impresora) y recursos de software (como una base de datos) en un ámbito local, regional o mundial
- Con el propósito de distinguir entre sí a las computadoras que conforman una red, a cada una de ellas se le asigna un nombre y una dirección numérica única
- El nombre de la computadora está conformado por un conjunto de caracteres alfanuméricos
- Para un ser humano es relativamente fácil recordar el nombre de una computadora o cualquier otro dispositivo de red. Sin embargo, dado que las computadoras internamente utilizan el sistema binario, para ellas resulta más fácil comunicarse usando direcciones numéricas en lugar de nombres, de manera que a cada una de ellas se le asigna una dirección única denominada dirección IP o dirección de Internet (IP significa Internet Protocol)
- Estas direcciones son números de 32 bits normalmente expresadas como 4 octetos separados por el carácter punto (.). Una típica dirección IP es 192.168.0.1. Cada uno de los 4 números de una dirección IP se denomina octeto porque pueden tener valores decimales de entre 0 y 255 (2x2x2x2x2x2x2x2 = 256 posibles valores por octeto)

Conceptos básicos de redes

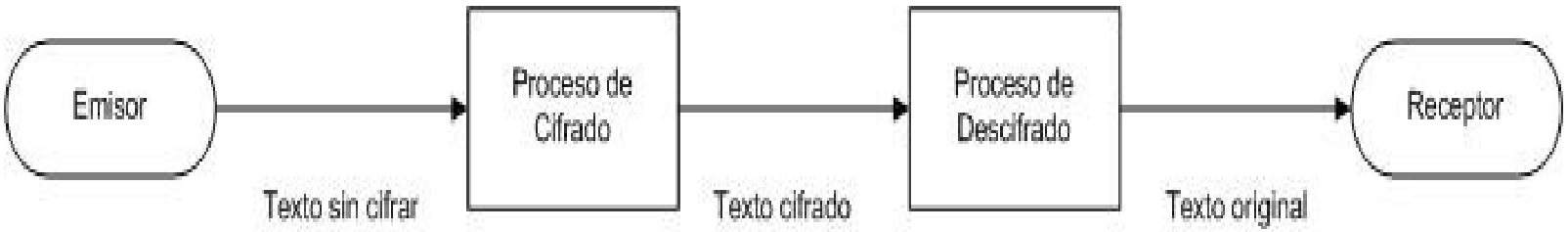
- Para que dos computadoras se comuniquen entre sí en una red es necesario que cada una de ellas posea un nombre y una dirección IP que las identifique de forma única. Sin embargo, también es necesario el uso de un lenguaje común por ambas partes (emisor y receptor), un conjunto de reglas que permita establecer la comunicación de forma correcta. A este conjunto de reglas se le conoce como protocolo
- Ahora bien, si las computadoras se comunican usando su dirección IP, que son números difíciles de recordar para los seres humanos, entonces se hace necesario un protocolo que se encargue de realizar la conversión de un nombre de computadora a una dirección IP y viceversa. Dicho protocolo se denomina DNS (Domain Name System)

Conceptos básicos de redes

- Las computadoras en red se comunican aún a un nivel más bajo, el cuál implica conocer la dirección física de cada una de ellas. Esta dirección física se denomina **MAC Address** y se refiere al número único que el fabricante ha asignado a la tarjeta de red de la computadora
- Una MAC Address típica se ve como **00:06:5b:88:4c:e8**, esto es, 6 pares de números en sistema hexadecimal separados por el carácter dos puntos (:). De este modo, se hace ahora necesario un protocolo capaz de convertir una dirección IP en una dirección física
- Tal protocolo es conocido como *ARP* (**Address Resolution Protocol**). El protocolo que realiza la operación inversa, esto es, convertir una MAC Address en una dirección IP, se conoce como RARP (Reverse ARP)

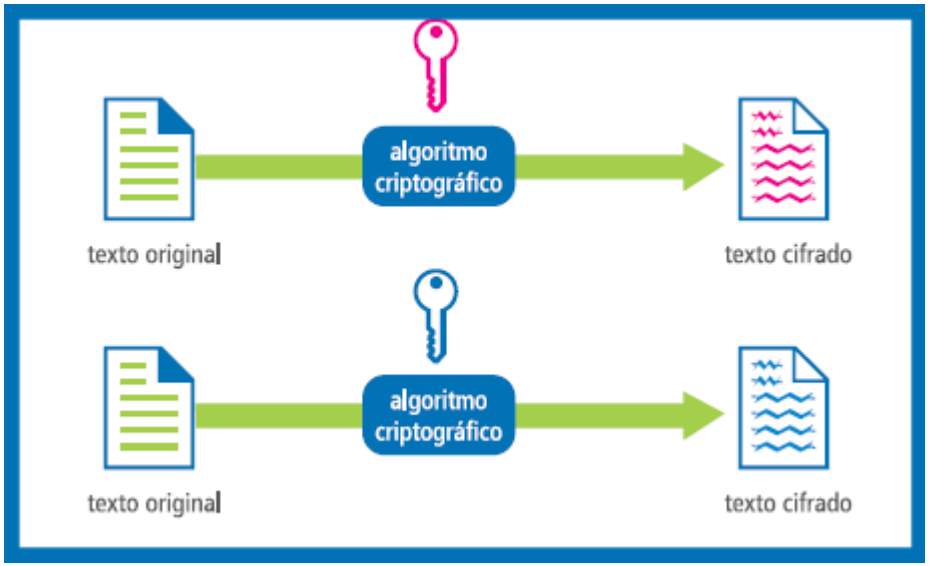
Criptografía

La criptografía es la ciencia que estudia los diversos métodos que permiten convertir un mensaje en un texto ininteligible mediante un procedimiento conocido como cifrado con el propósito de ocultar el mensaje original, y posteriormente recuperar dicho mensaje mediante un proceso conocido como descifrado. La siguiente figura muestra el proceso descrito:



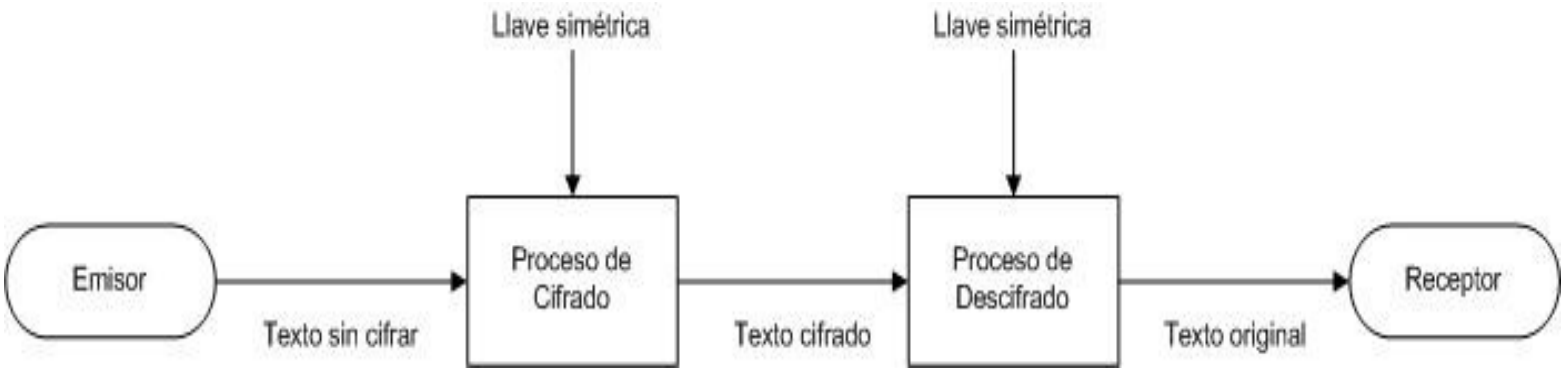
Criptografía

Las técnicas criptográficas se dividen en dos tipos genéricos: criptografía de llave simétrica y criptografía de llave pública

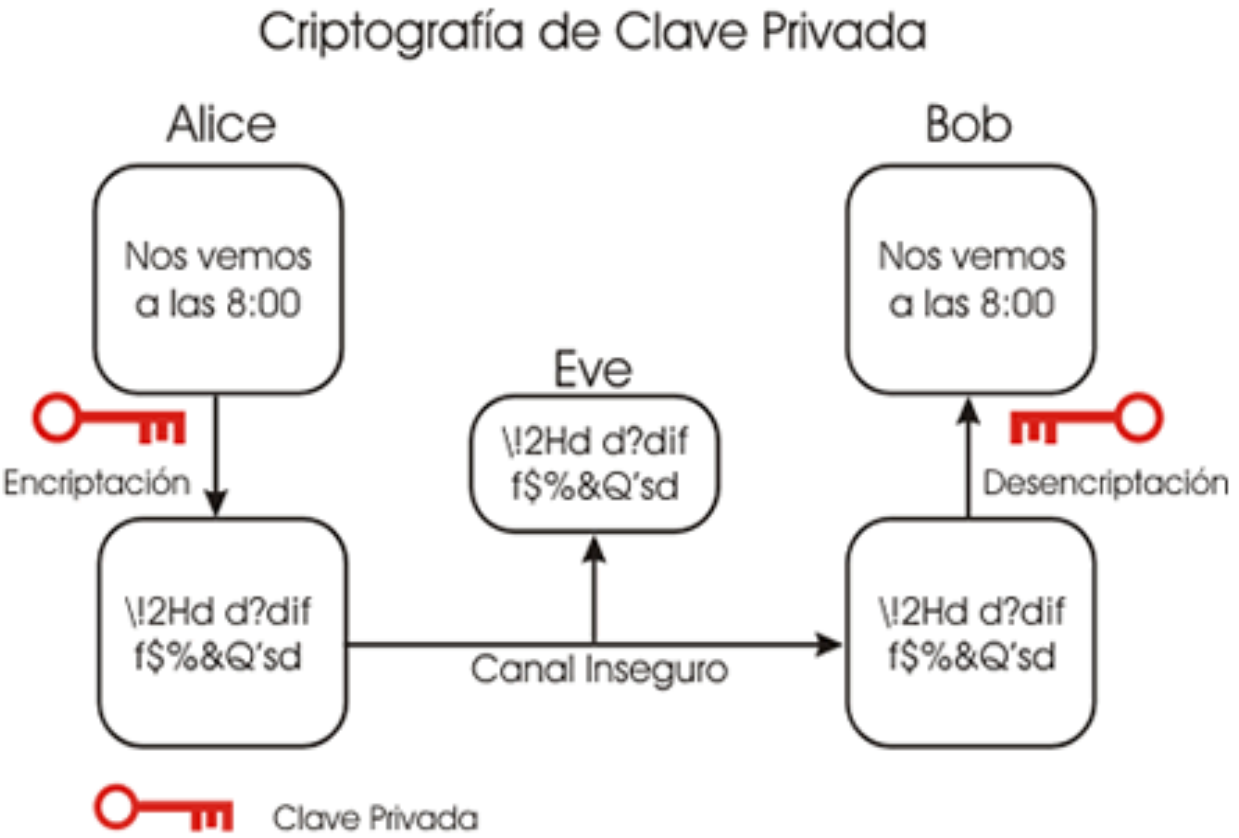


Llave simétrica

- La criptografía de llave simétrica utiliza una misma llave para cifrar y descifrar un mensaje, mientras la criptografía de llave pública utiliza una llave para cifrar y otra para descifrar
- La siguiente figura muestra el esquema de cifrado de llave simétrica:

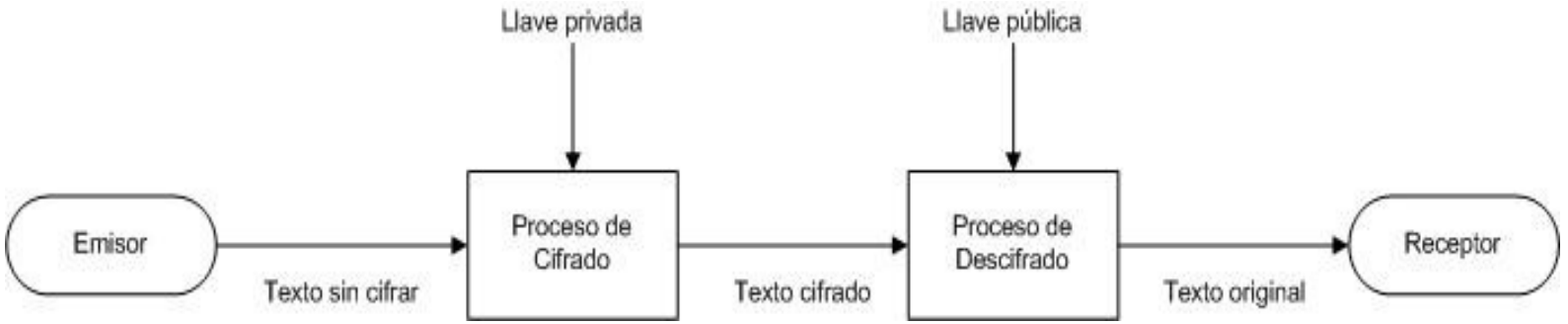


Llave simétrica

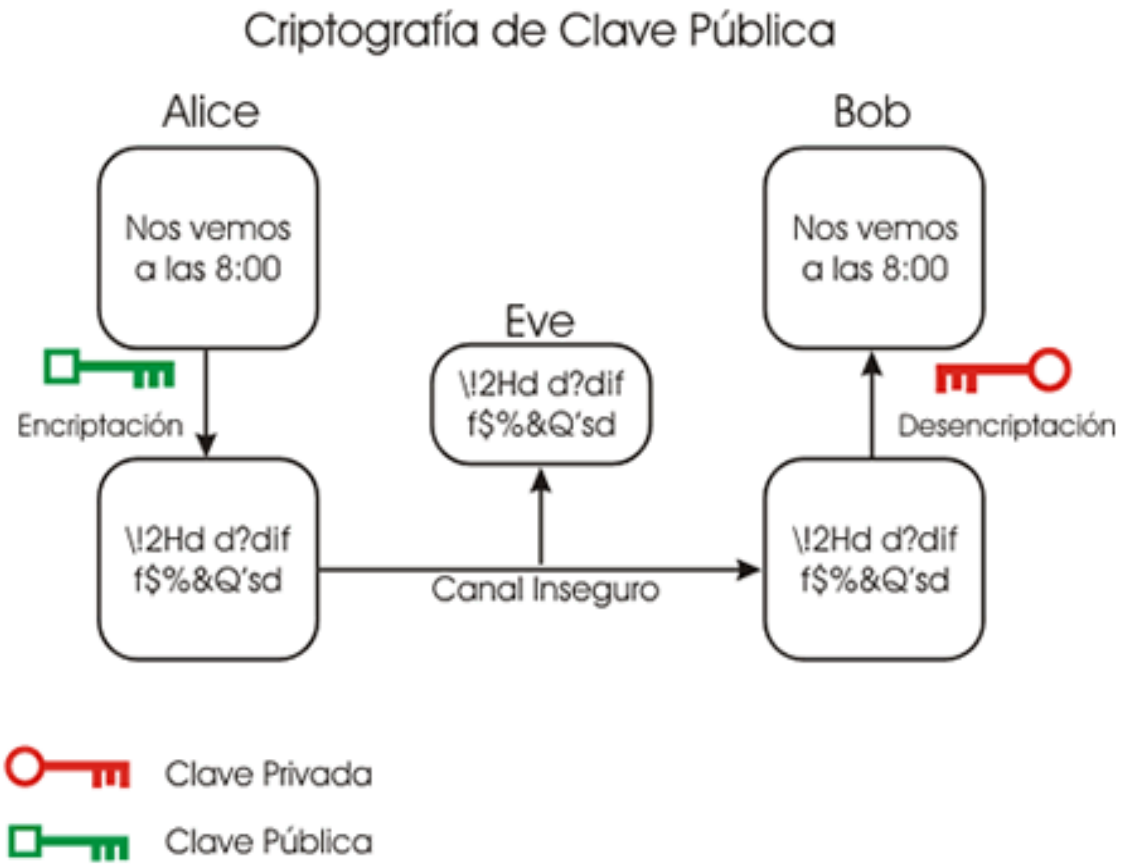


Llave pública

- La criptografía de llave pública resuelve el problema de la distribución de llaves mediante el uso de criptografía asimétrica. La idea detrás de la criptografía de llave pública es un par de llaves, que en realidad es un par de números generados al mismo tiempo de acuerdo a una fórmula matemática, pero dada una llave no es posible determinar la otra. Los datos que son cifrados con la llave privada (que es mantenida en secreto) pueden ser descifrados usando la llave pública (que se da a conocer públicamente) y viceversa. La siguiente figura ilustra el esquema de cifrado de llave pública:



Llave pública



Funciones hash

- Una de las contribuciones más importantes de la criptografía de llave pública es la firma digital y esta a su vez está basada en funciones hash
- Los algoritmos de hash generan un número de tamaño fijo (por ejemplo de 128 bits) independientemente del tamaño del mensaje. Ese número se denomina valor de hash y es generado de acuerdo a los bits de entrada; si uno sólo de los bits de entrada cambia, el valor de hash cambia también. Otra característica de este valor de hash es que “no es computacionalmente viable” que dado un valor de hash pueda determinarse el mensaje que lo generó, ni tampoco hallar dos conjuntos de datos que generen el mismo valor de hash. En otras palabras, el valor de hash es una especie de “huella digital” que representa exactamente los datos, pero en una forma abreviada

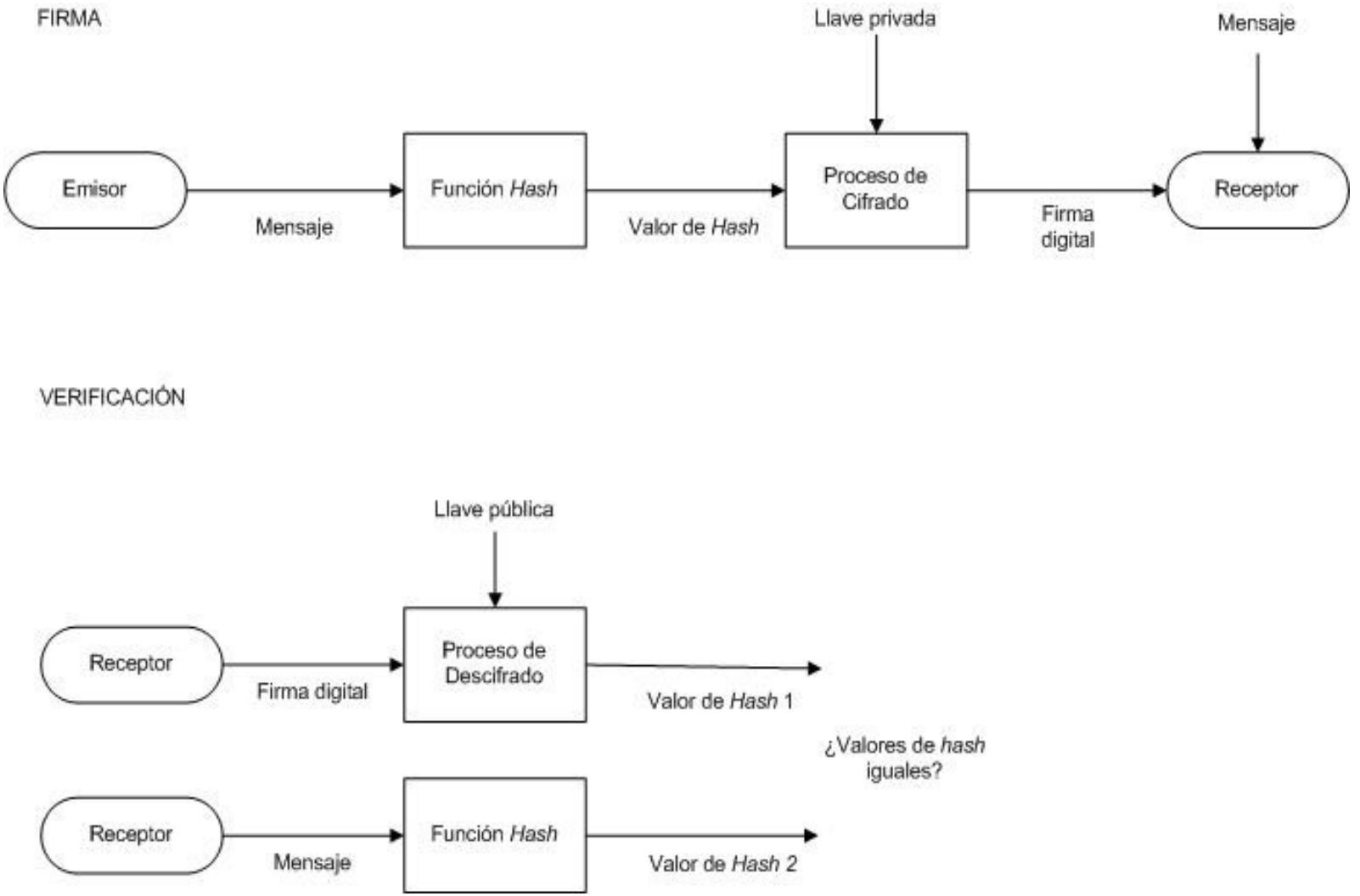
Firma digital

- Si se cifran datos con la llave privada, cualquiera que tenga acceso a la llave pública podrá descifrar los datos. Esto quiere decir que los datos cifrados no son privados, pero prueba que los datos realmente fueron cifrados por el dueño de la llave privada. Esto se conoce como proceso de firma
- Desafortunadamente, el hecho de cifrar una gran cantidad de datos con algoritmos de cifrado asimétrico es lento e implica más carga en la red, pero si el cifrado de llave pública es usado con propósitos de autenticación y no de confidencialidad, no es necesario cifrar el mensaje completo, sino que dicho mensaje alimenta a una función hash de una sola vía, de manera que únicamente se cifra el valor de hash derivado de dicha función, que a final de cuentas representa los datos originales. Este valor de hash cifrado se conoce ahora como firma digital y se envía al destinatario junto con el mensaje original para que aquél pueda autenticarlo, descifrando la firma digital, recalculando la función hash en base al mensaje recibido y comparando el valor obtenido y el que se ha recibido

Firma digital

- Para verificar una firma digital, el destinatario tiene que conocer la llave pública del remitente; una vez que la conoce, procede a descifrar la firma con la llave pública correspondiente para conocer el valor de hash. Ahora el destinatario calcula el valor de hash de los datos que recibe y lo compara con el valor de hash obtenido anteriormente. Si son iguales, la verificación es exitosa; de lo contrario, los datos se encuentran incompletos o corrompidos
- ¿Qué es lo que prueba la verificación de la firma digital? Prueba que los datos realmente fueron firmados con la llave privada del remitente, porque de lo contrario los datos no hubiesen podido ser descifrados con la llave pública de dicho remitente. El proceso de verificación prueba también que los datos recibidos junto con la firma digital no fueron modificados desde que el remitente los firmó, pues de lo contrario el valor de hash obtenido al descifrar el mensaje no hubiera coincidido con el valor de hash calculado. La siguiente figura resume el proceso de firma y verificación de un mensaje:

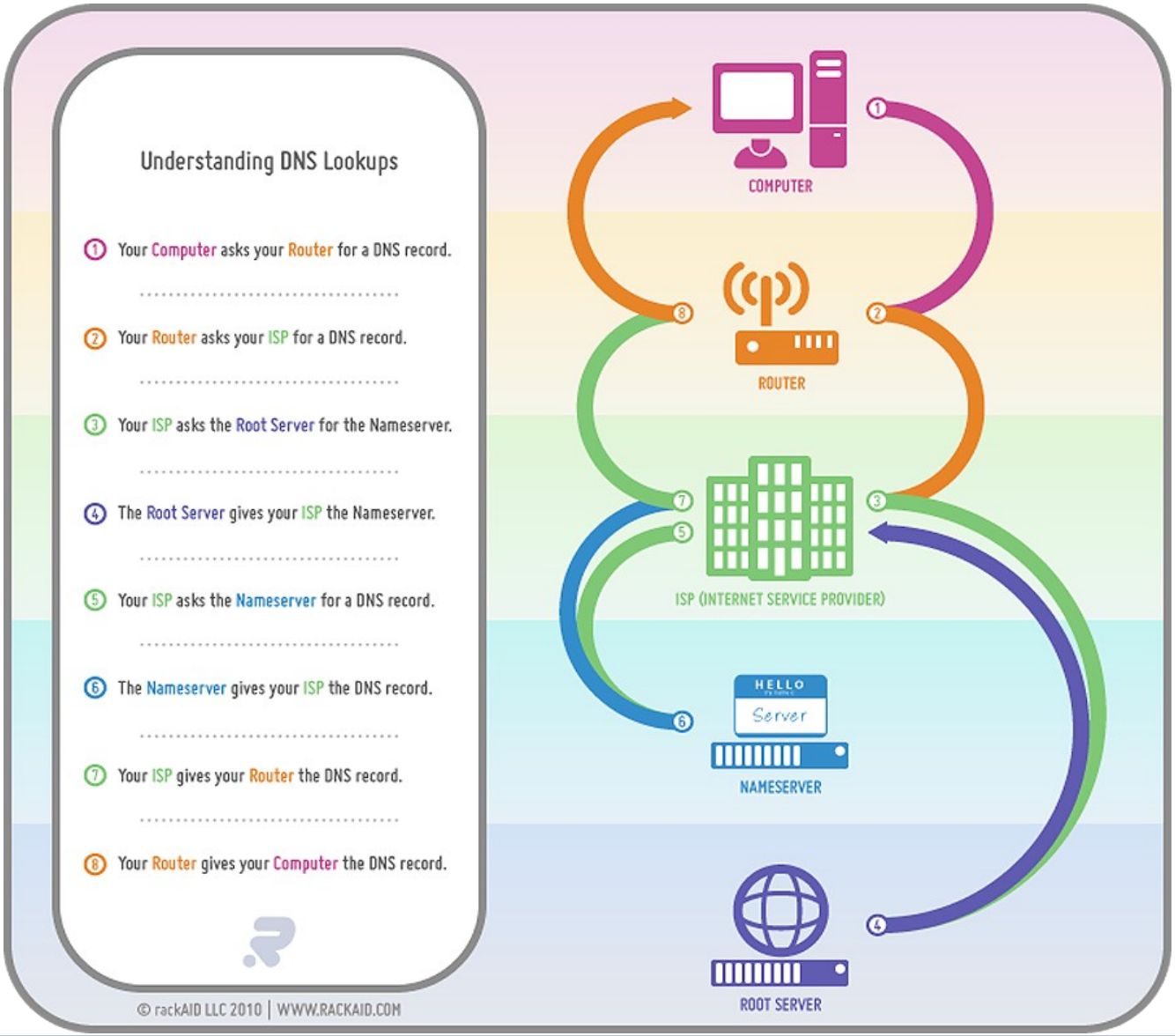
Firma digital



Teoría básica del DNS

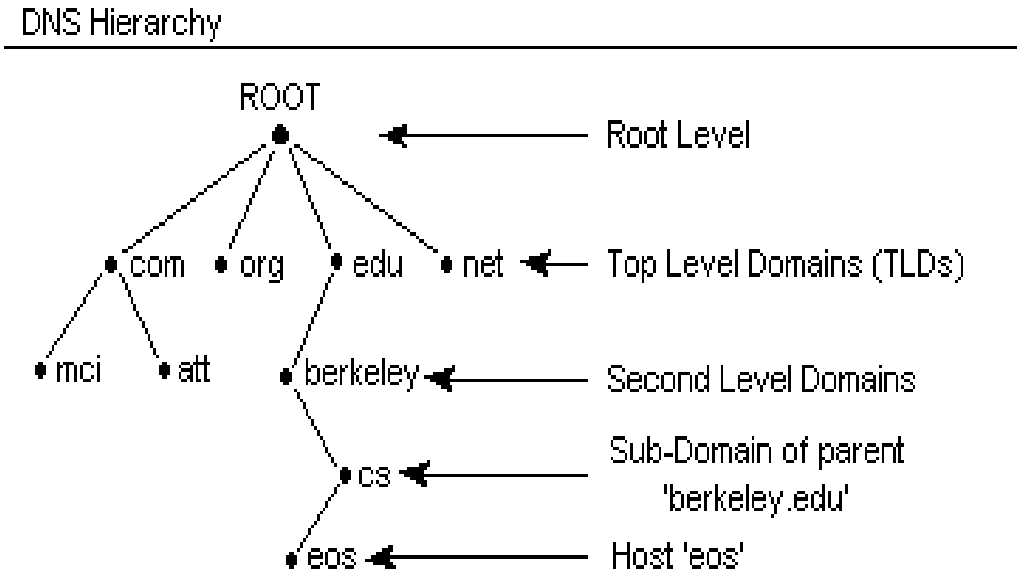
- Dominios, nombres de dominio y subdominios
- El espacio de nombres
- Servidores de nombres
- Resolvers
- Resolución de nombres
- Resolución directa e inversa
- Zonas y delegación
- Consultas recursivas e iterativas
- Caching y redundancia
- Dominios de alto nivel
- Dominios de segundo nivel
- Servidores de nombres raíz
- Registro de nombres de dominio
- Servidores maestros y esclavos
- Resource Records (RRs)
- Tipos de RR

Teoría básica del DNS



Teoría básica del DNS

- El DNS está basado en una estructura jerárquica, dónde la base de datos completa puede verse como un árbol invertido, con la raíz en la parte superior.
- El nombre de la raíz es la cadena nula “”, que generalmente se denota con un punto (.), como puede apreciarse en la siguiente figura:



Teoría básica del DNS

- Cada nodo en el árbol representa a un dominio (una parte de toda la base de datos) y cada dominio puede ser dividido a su vez en subdominios (subconjuntos de un dominio)
- A su vez, cada dominio tiene un nombre de dominio, que identifica su posición en la base de datos. El nombre de dominio completo o FQDN (Fully Qualified Domain Name) es la secuencia de etiquetas desde el dominio hasta la raíz, usando el punto (.) como separador entre etiquetas. En otras palabras, el FQDN es un nombre compuesto tanto por el nombre de la máquina como del nombre de dominio. Por ejemplo, el FQDN www.example.com incluye el nombre de la máquina www, y el nombre de dominio example.com
- De acuerdo al RFC 952, un nombre de dominio es una cadena de hasta 24 letras (de la A-Z y de la a-z), dígitos (del 0-9), o un carácter menos (-). El punto (.) sólo puede usarse para delimitar dominios

Teoría básica del DNS

- El espacio de nombres (name space) es el conjunto de todos los nombres de dominio en el árbol del DNS. Como puede adivinarse, un nombre de dominio es la llave para buscar un dominio en el espacio de nombres de dominio.
- Cada dominio puede ser administrado por una organización diferente, por lo que cada organización puede dividir su dominio en varios subdominios y delegar la responsabilidad de administrar tales subdominios a otras organizaciones. Por ejemplo, el Centro de Información de la Red (Network Information Center o NIC) de México (http://www.nic.mx) administra el dominio mx, pero delega la autoridad sobre el subdominio unam.mx al NIC de la UNAM (http://www.nic.unam.mx). A este proceso se le conoce como delegación de dominio.

Teoría básica del DNS

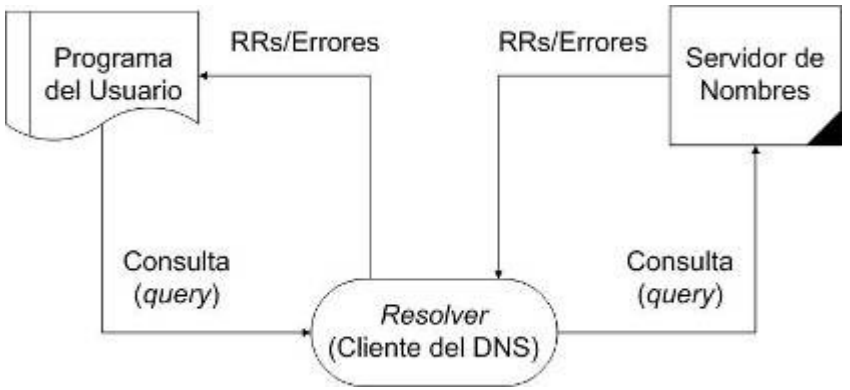
- Una zona es generalmente un subconjunto de un dominio, aunque a veces ambos conceptos pueden referirse a la misma parte del espacio de nombres de dominio
- Generalmente un dominio es segmentado en zonas para facilitar su administración. Para que una computadora pueda determinar qué servidor consultar para resolver el nombre de un servidor, el espacio de nombres del DNS es dividido en zonas
- Dentro de cada zona, uno o más servidores son considerados autoritarios (authoritatives) para resolver nombres a direcciones y viceversa

Teoría básica del DNS

- Al igual que otros servicios en Internet, el DNS está basado en un esquema cliente-servidor. La función del cliente la llevan a cabo unos programas conocidos como resolvers (el resolver esta incluido en todos los sistemas operativos que corren TCP/IP) y la parte del servidor la realizan los servidores de nombres.
- Los servidores de nombres son aquéllos que atienden las peticiones de los resolvers, y pueden clasificarse de la siguiente manera:
  - **Servidor de nombres raíz**. Es aquél que contiene las direcciones IP correspondientes a todos los servidores de nombres de alto nivel o de nivel superior.
  - **Servidor de nombres maestro (master) o primario**. Es aquél que contiene los archivos de datos de una zona.
  - **Servidor de nombres esclavo (slave) o secundario**. Es aquél que depende de un servidor de nombres maestro para mantener actualizados sus archivos de datos de zona. Sirve como respaldo del servidor de nombres maestro.
  - **Servidor de nombres de dominio de alto nivel o TLD (Top Level Domain)**. Es aquél que contiene los datos para los dominios de alto nivel o de nivel superior.
  - **Servidor de nombre autoritario (authoritative)**. Es aquél que contiene la información para sus zonas.
  - **Servidor de caché**. Es aquél que en lugar de publicar datos para una zona únicamente recuerda todas las respuestas a las consultas que ha hecho, con lo cuál se mejora la eficiencia de la red.

Teoría básica del DNS

- Los resolvers son bibliotecas (libraries) de programa que viene codificadas en los programas de aplicación como telnet y que sirven como interfaz entre la aplicación misma y el servidor de nombres, traduciendo las peticiones o consultas de la aplicación al servidor de nombres y las respuestas que éste proporciona.
- El mecanismo de resolución de nombres requiere que los resolvers hagan consultas o queries a los servidores de nombres, quienes a su vez responden con la información solicitada o bien, con mensajes de error, como se muestra en la siguiente figura:



Teoría básica del DNS

- El servidor que contiene las tablas de la base de datos original para la zona se denomina servidor maestro o primario para esa zona. De manera opcional, pueden configurarse uno o más servidores esclavos o que mantengan una copia sincronizada de las tablas de la zona para implementar un sistema redundante y tolerante a fallas, o bien, simplemente para distribuir la carga en el servidor maestro
- Cuando un DNS, ya sea maestro o esclavo, responde a una petición para una computadora que se encuentra dentro de su dominio, la respuesta se considera autoritaria (authoritative), pero si la máquina en cuestión se localiza en otro dominio, es posible que el servidor consultado pueda haber tenido el registro de DNS de la máquina buscada almacenado en su caché, debido a que se había consultado previamente. En ese caso, la respuesta se considera no autoritaria (non-authoritative), ya que es posible, aunque poco probable, que la información almacenada en el caché no esté actualizada

Teoría básica del DNS

- La acción de traducir un nombre en una dirección IP se conoce como “resolver el nombre de dominio”, por lo que el proceso de convertir un nombre de dominio en una dirección IP se conoce como resolución (directa) de nombres
- De igual manera, el proceso de convertir una dirección IP en un nombre de dominio se conoce como resolución inversa de nombres.

Teoría básica del DNS

- Cuando se consulta a un servidor de nombres acerca de una máquina que se encuentra fuera de su zona, el servidor usa uno de dos mecanismos de resolución: recursivo y no recursivo (también conocido como iterativo).
- **Resolución recursiva**. En este tipo de resolución el servidor de nombres recibe una consulta y si no conoce la respuesta, entonces contacta a otro servidor de nombres en espera de la misma; si este otro servidor de nombres tampoco conoce la respuesta, entonces contacta a otro y así sucesivamente, hasta que encuentra la respuesta o un mensaje de error.
- **Resolución iterativa**. En este tipo de resolución el servidor de nombres recibe una consulta y si no conoce la respuesta, entonces simplemente regresa al cliente (que podría ser el resolver u otro servidor de nombres) una referencia a la dirección IP de otro servidor de nombres que posiblemente si conozca la respuesta a la petición

Teoría básica del DNS

- Para acelerar el proceso de resolución, una vez que un servidor de nombres resuelve una consulta (query), almacena en un área de caché todas los nombres de dominio y direcciones IP que recibe. Por ejemplo, una vez que ha realizado una consulta a un servidor de nombres raíz preguntando por el dominio .com, ya conoce la dirección IP para el servidor de nombres que maneja tal TLD, de manera que en lo sucesivo ya no será necesario preguntar nuevamente esa información a un servidor de nombres raíz.
- Obviamente, los servidores de nombres no guardan esta información para siempre, ya que el caché tiene un componente llamado Tiempo de Vida (Time To Live o simplemente TTL) que controla el tiempo que el servidor de nombres guardará la información en su caché. Cuando el servidor de nombres recibe la dirección IP, recibe también el TTL, de manera que guardará la información durante el periodo de tiempo indicado por el TTL (este tiempo varía de minutos a días) y luego la desechará. El TTL permite que se propaguen los cambios en los servidores de nombres

Teoría básica del DNS

- Otra de las características clave del DNS es la redundancia.
- Existen muchos servidores de nombres en cada nivel, de manera que si uno falla, hay otros que pueden manejar las peticiones. Por ejemplo, Google cuenta con 2 servidores de nombres:
    - google-public-dns-a.google.com (8.8.8.8)
    - google-public-dns-b.google.com (8.8.4.4)

Teoría básica del DNS

- Los dominios de alto nivel o de nivel superior, mejor conocidos como TLD (Top-Level Domain) son fijos, y por razones históricas caen dentro de dos categorías:
  - El primer grupo está compuesto por TLDs dentro de los Estados Unidos (USA), tales como .edu, .com o .gov. Estos dominio se conocen como generic TLD o simplemente como gTLDs
  - El segundo grupo de dominios se localiza fuera de USA y se denotan con abreviaciones de dos letras de acuerdo a la ISO (International Standard Organization) para el país de origen (como .mx para México y .us para USA, aunque este último TLD casi no se usa). Este tipo de TLDs se conocen como country-code TLDs o simplemente ccTLDs

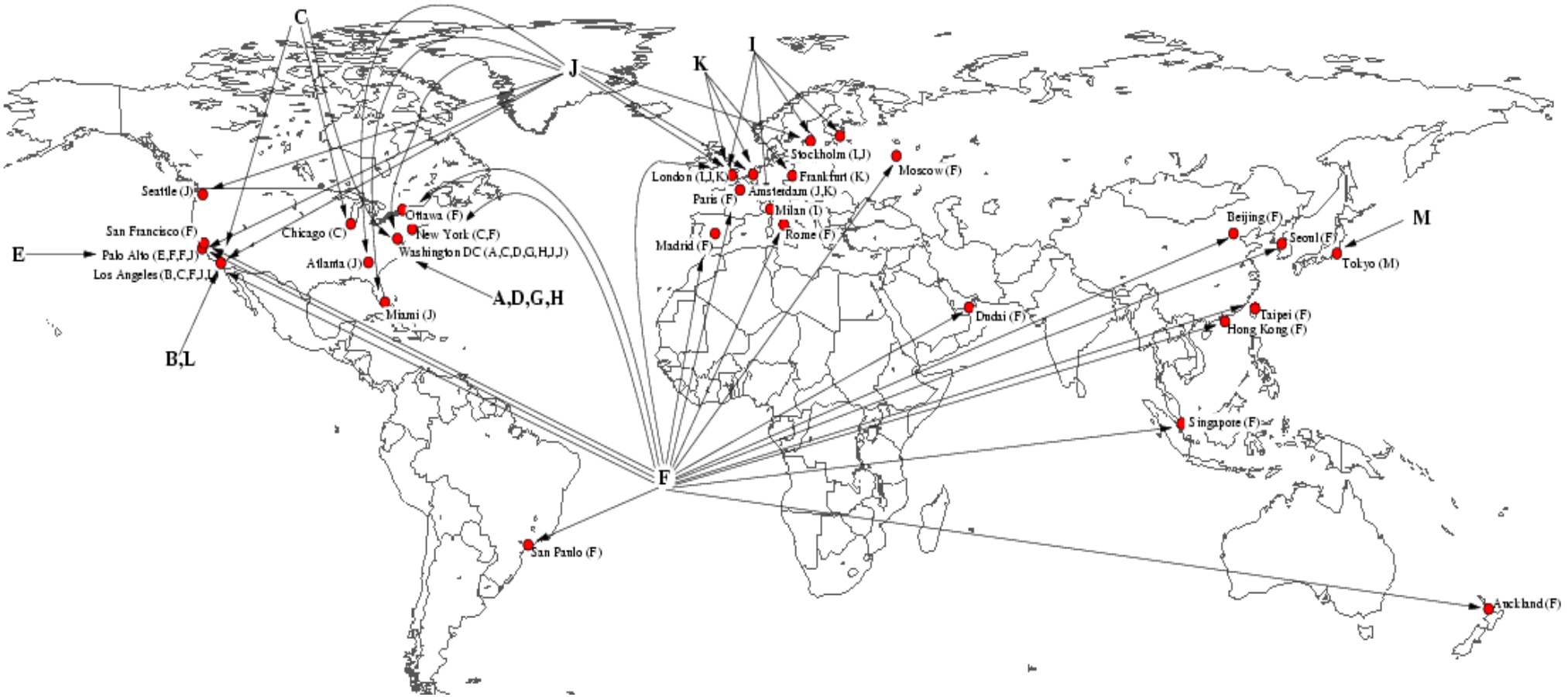
Teoría básica del DNS

- Dentro de cada TLD existen múltiples SLD o dominios de segundo nivel (Second Level Domain).
- Los nombres de dominio son únicos, así que las organizaciones pueden elegir libremente los nombres dentro de sus dominios. Cualquier nombre que elijan no entra en conflicto con el elegido en otros nombres de dominio, puesto que cada nombre de computadora tendrá su propio nombre de dominio único. En otras palabras, cada nombre en un TLD debe ser único, pero si puede haber dominios de segundo nivel duplicados.
- En el caso de www.seguridad.unam.mx, seguridad es un Dominio de Tercer Nivel (Third Level Domain). Es posible usar hasta 127 niveles, pero es raro que se usen más de 4.
- La palabra de más a la izquierda en el URL (como www en el ejemplo anterior) es el nombre del servidor (hostname) de Web, que hace referencia a un nombre dentro de un dominio y a una dirección IP específica.

Teoría básica del DNS

- La organización denominada ICANN (http://www.icann.org) es responsable de administrar y coordinar el DNS para asegurar que la resolución de nombres sea universal.
- ICANN (Internet Corporation for Assigned Names and Numbers) es una organización mundial no lucrativa que asegura que el DNS funcione de forma efectiva supervisando la distribución de nombres de dominio y direcciones IP únicas.
- El ICANN coordina 13 computadoras denominadas servidores raíz (root servers) que se encuentran distribuidos en todo el mundo. Los 13 contienen la misma información con fines de respaldo mutuo y reparto de la carga de trabajo. Estos servidores contienen las direcciones IP de todos los dominios TLD.
- En marzo de 2004 NIC México, Internet Systems Consortium (ISC), Prodigy Data Center, Avantel y Alestra instalaron el primer Servidor Raíz de Nombres de Dominio en México. El Servidor Raíz de Nombres de Dominio instalado en la ciudad de Monterrey, es réplica a nivel mundial del servidor “Root Server F”, que opera Internet Software Consortium (ISC) en Woodside, California.
- Como puede adivinarse, los servidores de nombres raíz son vitales en el proceso de resolución de nombres, por lo que cada servidor de nombres debe tener una lista de todos los servidores de nombres raíz, de manera que cuando el NS necesite resolver un dominio, contacte al primer servidor de nombres raíz de la lista y si éste no responde, entonces contacte al segundo y así sucesivamente

Teoría básica del DNS



Teoría básica del DNS

- Puesto que todos los nombres en un dominio dado deben ser únicos, debe existir una única entidad que controle la lista y se asegure de que no se dupliquen los nombres. Por ejemplo, el dominio .com es administrado por una compañía llamada Verisign (http://www.verisign.com), mientras el dominio .biz es operado por la compañía NeuLevel, Inc. (http://www.neulevel.biz)
- Cuando alguien registra un nuevo nombre de dominio en el TLD.com, la actualización debe hacerse en todos los sitios de registro que trabajan con Verisign, quien a su vez mantiene una base de datos central conocida como whois que contiene información acerca del propietario y los servidores de nombres para cada dominio en el TLD.com

Teoría básica del DNS

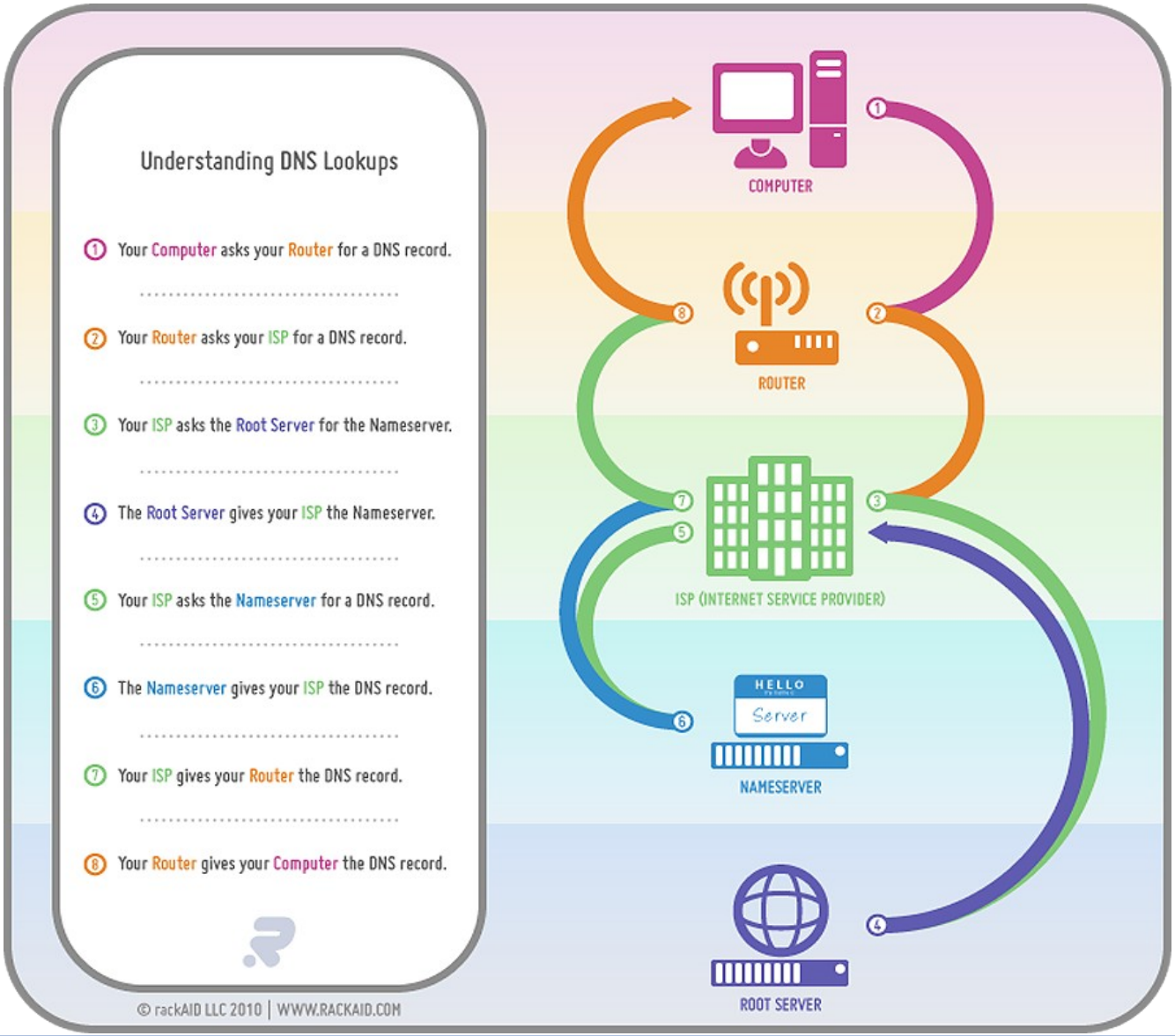
- La base de datos del DNS para un dominio contiene un conjunto de registros de recursos, mejor conocidos como RR (resource records). Estos RR están clasificados en varias categorías, siendo algunas de ellas obligatorias y otras opcionales, como puede apreciarse en la siguiente tabla:

Categoría	Tipo	Nombre	Función
Zona	SOA	Start of Authority	Define la autoridad para una zona del DNS
	NS	Name Server	Especifica el servidor de DNS para la zona
Básico	A	Address	Traslación del nombre de servidor a su correspondiente dirección IP
	PTR	PoinTeR	Traslación de la dirección IP a su correspondiente nombre de servidor
	MX	Mail eXchanger	Usado para enrutar correo electrónico
Opcional	CNAME	Canonical NAME	Alias para el nombre de un servidor
	HINFO	Host INFOrmation	Especifica el hardware y sistema operativo para una máquina
	TXT	TeXT	Texto ASCII no interpretado

Teoría básica del DNS

- Existe un único registro SOA para cada zona, que contiene información tal como el NS primario para el dominio, la dirección electrónica del responsable de dicho dominio, el TTL de la zona, etc.
- El registro NS denota los servidores de nombres que son autoritarios para el dominio en cuestión.
- Los registros A y su contraparte PTR permiten mapear un nombre de servidor a una dirección IP y una dirección IP a un nombre de servidor, respectivamente.
- Los registros MX permiten que los sistemas de correo determinen a que máquina deben enviar el correo para un dominio en particular. Este registro generalmente va acompañado de parámetro denominado valor de preferencia (preference value), que es un número sin signo de 16 bits que indica la prioridad del registro MX, es decir, el orden en que se hará uso de él.
- Los 3 últimos registros (opcionales) pueden o no estar presentes en un dominio dado. El registro CNAME es muy útil para definir alias para un servidor.
- Por su parte, el uso del registro HINFO no se recomienda, ya que puede revelar información a terceros que puede facilitar un ataque.
- Cada dominio tiene un servidor de nombres de dominio (NS) en algún lado, el cuál se encarga de atender sus peticiones y existe también una persona que da mantenimiento a los registros en ese DNS. A esta persona se le conoce como administrador del DNS.

Teoría básica del DNS



Agenda
Día 2

- Seguridad en el DNS
- Instalación de BIND
- Configuración básica de un DNS
- Herramientas de consulta al DNS

Seguridad en el DNS

- Vulnerabilidades
- Ataques
  - Ataques a BIND
  - Cache Poisoning
  - DNS Spoofing
  - DoS/DDoS
  - Amplification Attacks
- Contramedidas
  - Seguridad en la red
  - Seguridad en el DNS
  - Seguridad en los servidores de nombres
  - Seguridad en BIND
  - Seguridad en las zonas
  - Seguridad en las transacciones
  - DNSSEC

Vulnerabilidades

- El viernes 31 de Octubre de 1997, Eugene Kashpureff fue arrestado por la Real Policía Montada de Canadá en Toronto, acusado de “secuestrar” el URL del sitio de registro de dominios de Internet InterNIC (www.internic.net) en julio de 1997. Alegando protesta por el supuesto monopolio que mantiene InterNIC sobre los TLDs, Kashpureff modificó los registros del DNS de manera que los usuarios que ingresaban al sitio http://www.internic.net eran redireccionados a su propio sitio Web http://www.alternic.net, donde manifestaba su inconformidad.

Vulnerabilidades

- En la tarde del lunes 21 de octubre de 2002 los 13 servidores raíz de nombres de dominio sufrieron un ataque de negación de servicio distribuido (DDoS, por sus siglas en inglés) que impactó a Internet entero por espacio de una hora. Este tipo de ataques consiste en concentrar la potencia de muchas computadoras en contra de una única red con el objeto de detener su operación. Este fue el ataque más fuerte y complejo que se había hecho en contra de los servidores de nombres raíz. Por fortuna, aunque sólo 4 o 5 de los 13 servidores resistieron el ataque, no hubo mayor afectación para el usuario común.

Vulnerabilidades

- El 25 de abril de 2005 todos los usuarios que deseaban entrar al sitio Web de servicio de correo electrónico seguro denominado hushmail (www.hushmail.com) eran redireccionados a un sitio falso. La empresa Hush Communications reconoció que los intrusos habían cambiado los registros del DNS de hushmail luego de comprometer la seguridad de su registrar (Network Solutions). Estos cambios fueron revertidos luego de unas horas y el servicio de hushmail volvió a la normalidad. En un comunicado, Hush Communications se disculpó diciendo que no hubo acceso no autorizado a ninguno de sus servidores y que los datos no habían sido comprometidos, pero reconoció que durante el periodo del ataque todo el correo electrónico enviado al dominio hushmail.com no había sido recibido. También sugirieron a sus usuarios ser muy cuidadosos y asegurarse de que se encuentran en su sitio Web seguro antes de ingresar su frase de paso (passphrase) para autenticarse.

Vulnerabilidades

- El 6 de febrero de 2007 fueron atacados 2 de los 13 servidores raíz. Se trató de un ataque de negación de servicio a los servidores G y L.
- El ataque causó que estos servidores no pudieran responder al menos 90% de las consultas.
- Los 2 servidores afectados son administrados por el ICANN y el DoD de USA

Vulnerabilidades

- Como fácilmente puede apreciarse en los ejemplos anteriores, el DNS es susceptible de ser atacado por intrusos precisamente por su naturaleza de sistema distribuido. Los ataques maliciosos en servidores de nombres locales en redes locales pueden resultar en respuestas del DNS falsificadas que desvían o secuestran el tráfico. Como resultado, podría aparecer a los usuarios una página Web falsificada, aún cuando el servidor Web de la víctima no sea tocado por el intruso.
- Una amenaza aún más seria, la corrupción de datos del DNS, puede llevar al mal direccionamiento o al mal envío de correo electrónico. El hecho de que existan millones de servidores de nombres en el mundo, aunado al hecho de que muchos no ejecutan la última versión del software de BIND, o peor aún, corren software mal configurado, los convierte en objetivos para los intrusos.

Ataques a BIND

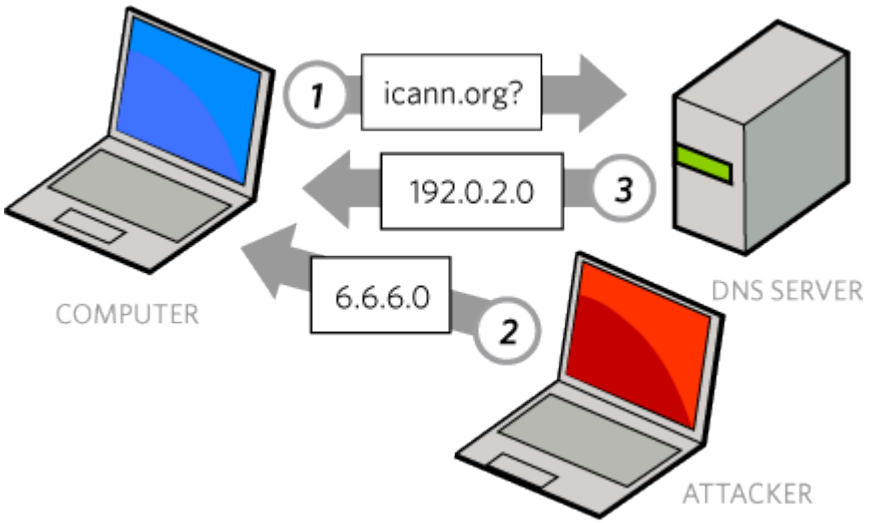
- Una consecuencia de una vulnerabilidad en el DNS es la posibilidad de que un intruso obtenga acceso de superusuario en el servidor de nombres. Este ataque generalmente es llevado a cabo aprovechando una vulnerabilidad de desbordamiento de búfer (buffer overflow).

Ataques a BIND

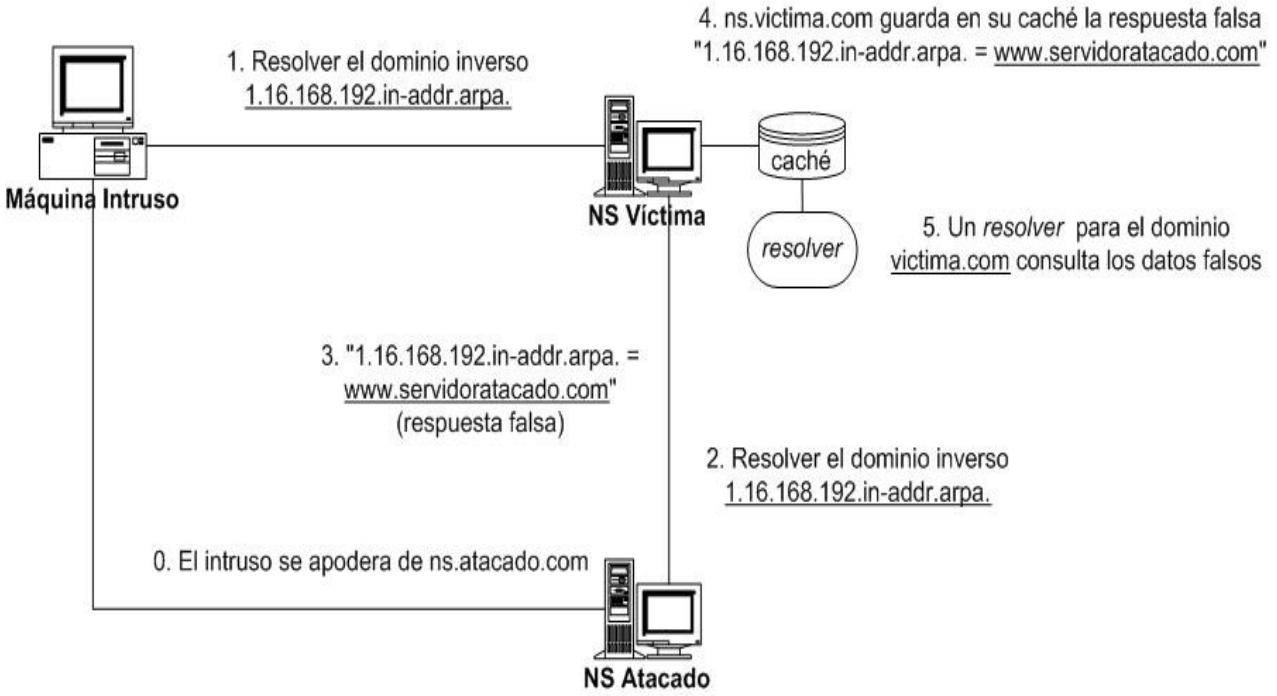
- Este ataque sucede cuando un programa acepta de una fuente externa más datos de los que puede almacenar en la memoria que tiene asignada para ello. Los datos extra son desbordados en una región de la memoria dónde se encuentran alojadas las instrucciones, y entonces se ejecutan como si fueran parte del programa original. Una vez que esto ocurre, un intruso puede perpetrar muchos ataques, como la intercepción de datos, la inserción de datos falsos en el sistema, la inserción de agentes de DDoS y el uso de la máquina para atacar otros sitios.
- La manera de prevenir este ataque es muy sencilla: usar siempre la última versión del software de DNS y ejecutar el programa servidor de nombres en un ambiente controlado y sin privilegios especiales.

Envenenamiento de cache

- Un ataque de envenenamiento de caché básicamente consiste en que se introduzcan registros falsos en el caché de un servidor de nombres



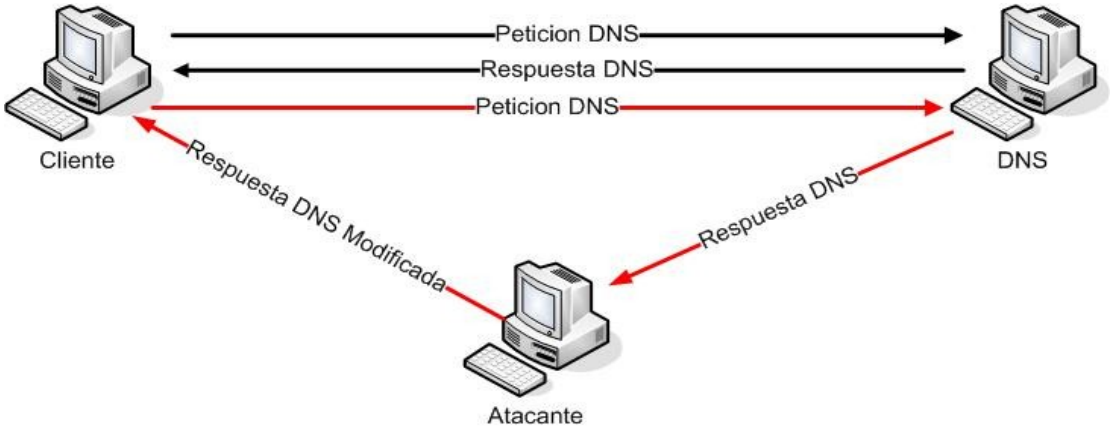
Envenenamiento de cache



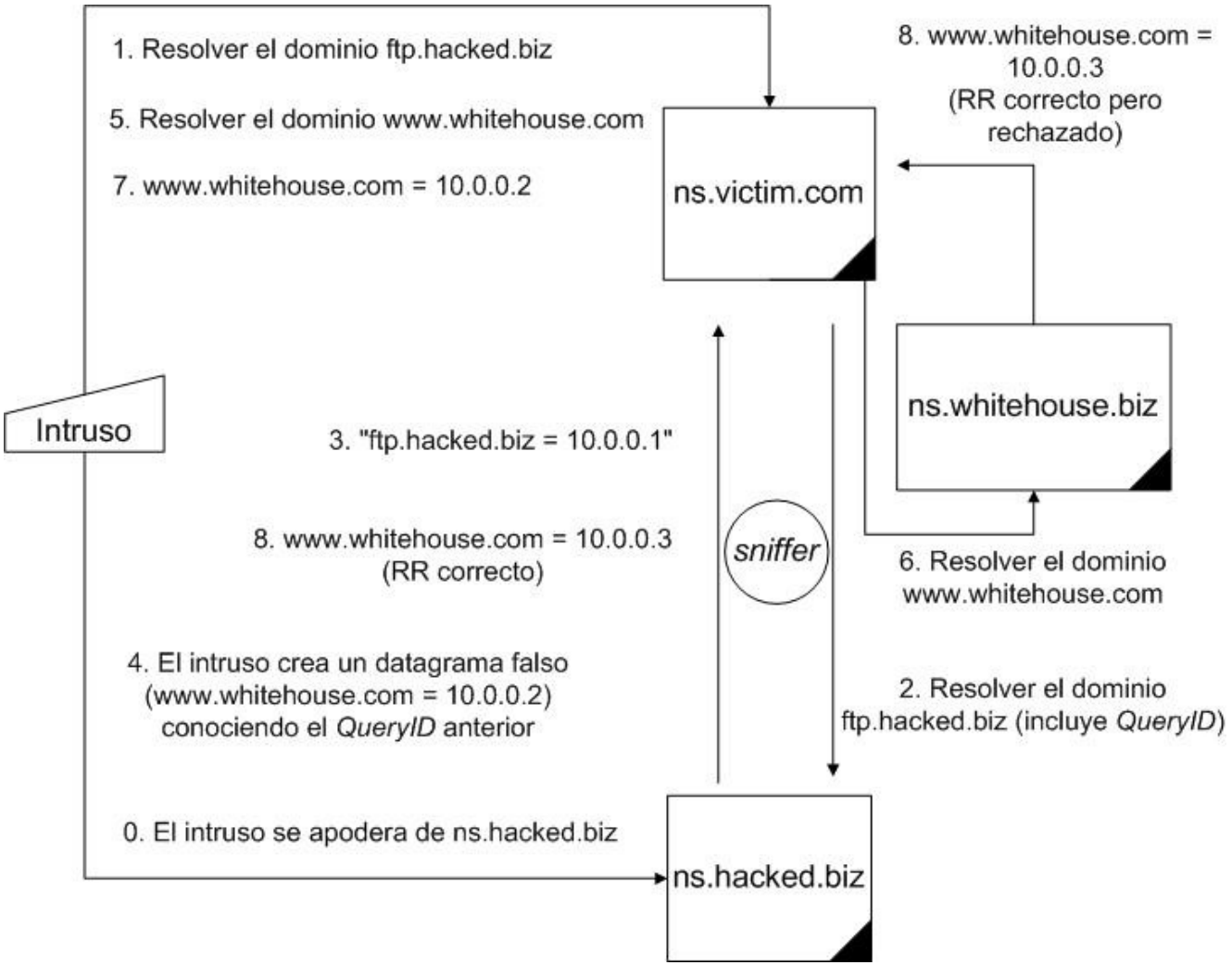
- Una forma simple de hacer más difícil este tipo de ataque es desactivando la recursión en un servidor de nombres y hacer uso del protocolo TSIG para autenticar las transacciones entre servidores de nombres

Suplantación de un DNS

- El ataque de DNS Spoofing consiste en que un servidor de nombres acepta y utiliza información incorrecta de una máquina que no posee autoridad para proporcionar tal información, la cuál puede luego ser usada para cambiar y reenrutar las direcciones IP para acceso al Web, correo electrónico y transferencias de archivos a sitios arbitrarios elegidos por un intruso. Por ejemplo, si un sitio de comercio electrónico es suplantado mediante una vulnerabilidad en el DNS, los clientes de una empresa podrían ser direccionados al sitio Web de un intruso para ingresar sus números de tarjeta de crédito.
- Los ataques de DNS Spoofing son sumamente difíciles de prevenir, pero las extensiones de seguridad del DNS (DNSSEC) podrían ser de gran ayuda.



DNS Spoofing



Negación de Servicio

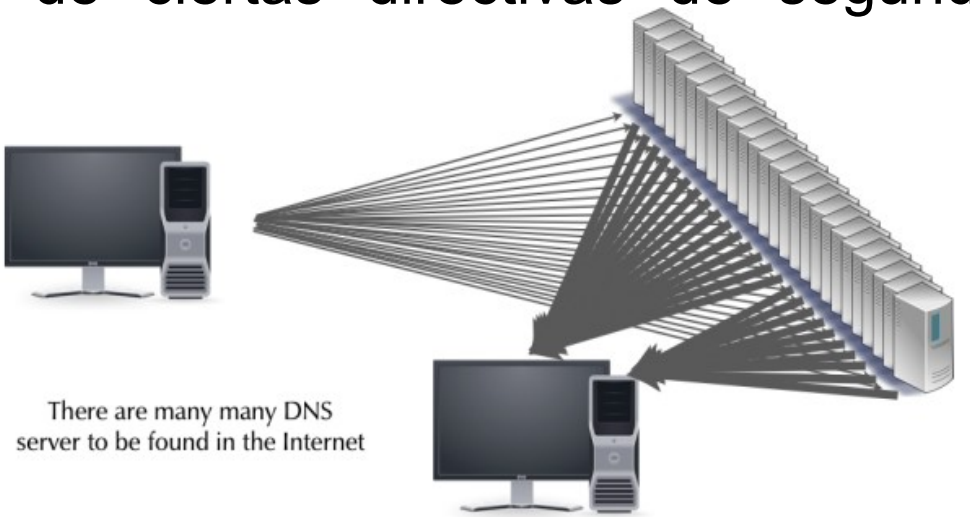
- Un ataque de negación de servicio, mejor conocido como DoS por sus siglas en inglés (Denial of Service) ocurre cuando una computadora, un servicio de red, o una red misma es atacada con el único fin de agotar sus recursos y dejarlo indispuesto para atender los requerimientos de usuarios legítimos.
- En un ataque DoS, un sistema típicamente utiliza las facilidades locales para atacar a un único sistema objetivo. Lo único que debe hacer la fuente de ataque es atacar algún componente, lo cuál provocará que los intentos legítimos de conexión sean rechazados. Los efectos de este tipo de ataque pueden ser devastadores para una empresa que depende del acceso a la red.
- Los ataques de negación de servicio también son muy difíciles de prevenir, pero una buena contramedida es colocar el servidor de nombres en un arreglo de red que incluya una zona desmilitarizada (DMZ) protegida por un firewall, que permita filtrar el tráfico hacia y desde el servidor de nombres.

Negación de Servicio Distribuida

- Cuando un ataque DoS es coordinado y proviene de diversas fuentes al mismo tiempo, se conoce como negación de servicio distribuido o DDoS (Distributed Denial of Service).
- Un ataque DDoS es más intenso y dañino que un ataque DoS porque está diseñado para atacar a un sistema desde diversas fuentes al mismo tiempo. Muchas empresas han sufrido ataques de este tipo, incluyendo Microsoft (http://www.microsoft.com), GRC (http://grc.com) y Blue Security.
- Los ataques de tipo DDoS son tan certeros, que incluso ha habido empresas que no logran recuperarse del mismo, como fue el caso del ISP británico CloudNine Communications, quién fue atacado a finales de enero de 2002 y se vio forzado a venderse a la competencia (Zetnet) porque no pudo recuperarse con la rapidez que el mercado exigía.
- Quizá el ataque de negación de servicio al DNS más memorable (hasta la fecha) sea el que ocurrió el 21 de octubre de 2002, cuando se realizó un DDoS en contra de los servidores raíz del DNS.

Ataques de Amplificación

- Los ataques de amplificación consisten en generar respuestas mucho mayores a las esperadas, saturando por tanto el ancho de banda de la víctima. Por ejemplo, si un sistema generalmente recibe paquetes de un tamaño de 50 bytes y de repente se envían paquetes de 1500 bytes, el tráfico se incrementa 30 veces.
- Para el caso de BIND, este tipo de ataques puede evitarse haciendo uso de ciertas directivas de seguridad incluidas en BINDv9.



Contramedidas: Seguridad en la red

- No colocar todos los servidores de DNS detrás de un solo router
- No colocar todos los servidores de DNS en el mismo segmento de red
- Distribuir geográficamente los servidores de DNS
- Implementar software que resista los ataques DoS
- Dividir los servidores de DNS (DNS-split) en internos (privados) y externos (públicos) y configurar un esquema apropiado de seguridad a nivel red
- En cuanto a protección contra ataques DoS/DDoS, los ISPs necesitan ser más receptivos a las peticiones para reaccionar ante este tipo de ataques

Contramedidas: Seguridad en el DNS

- Configurar correctamente los servidores de DNS, apoyándose con herramientas, consultores, y diversas soluciones
- Restringir las consultas y las transferencias de zona a servidores autorizados
- Diversificar en lugar de homogenizar (correr los servidores de nombres en plataformas distintas)
- Evitar lame-delegation (delegar una zona a un servidor de nombres que no sea autoritario para esa zona)

Contramedidas: Seguridad en el DNS

- No instalar, desinstalar o al menos desactivar el software del DNS en las máquinas que no lo requieran
- Asegurar el sistema operativo de la máquina que ejecutará el software del DNS
- Correr los servidores de DNS en máquinas dedicadas
- Asegurar el propio software del DNS
- Revisar periódicamente la integridad de los archivos de configuración del software del DNS
- Ejecutar el programa del software del DNS como un usuario sin privilegios especiales
- “Enjaular” el programa del software del DNS en una estructura de directorios chroot
- Mantener siempre actualizado el software del DNS y/o aplicar los parches o workarounds que resuelvan las vulnerabilidades
- Estar al pendiente de los avisos de seguridad de los proveedores del hardware, sistema operativo y software del DNS
- Si se cree conveniente, considerar el uso de software del DNS alterno a BIND
- Verificar las delegaciones dentro de los servidores de nombres, de manera que las peticiones que no puedan responder sean delegadas a servidores de nombres que sí puedan hacerlo (las “lame delegations” pueden permitir el envenenamiento de caché)

Contramedidas: Seguridad en BIND

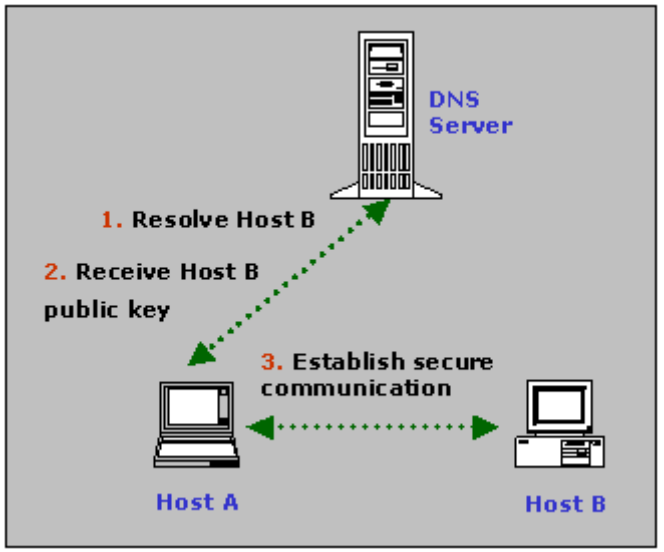
- Usar siempre la versión estable más reciente de BIND
- Ocultar la versión de BIND (sentencia version)
- Restringir las consultas (sentencia allow-query)
- Restringir las transferencias de zona (sentencia allow-transfer)
- Ejecutar BIND con un usuario sin privilegios (named) para mayor seguridad en el caso de futuros ataques remotos. Sin embargo, hay que tener en cuenta que sólo los procesos que se ejecutan como root pueden ser configurados para que usen los puertos inferiores al 1024, lo cuál es un requisito del DNS, por lo que debe configurarse BIND para que cambie de identidad una vez que se ha asociado al puerto.
- Evitar en lo posible glue records (sentencia fetch-glue, yes por default en BINDv8) para defenderse de ataques de envenenamiento de caché. Los glue records son RRs tipo A que permiten resolver el clásico problema del “huevo y la gallina”: para que un resolver o cliente sea capaz de resolver el nombre de dominio de un servidor de nombres, dicho servidor debe contener un registro tipo A que apunte a su propia dirección IP.
- Evitar en lo posible la recursión (sentencia recursion, yes por default; sentencia allow-recursion)
- Evitar la notificación automática de actualización de la zona (sentencia notify, no por default)
- Evitar en lo posible las actualizaciones dinámicas (sentencia allow-update)
- Ignorar las consultas de ciertas máquinas (sentencia blackhole)

Contramedidas: Seguridad en BIND

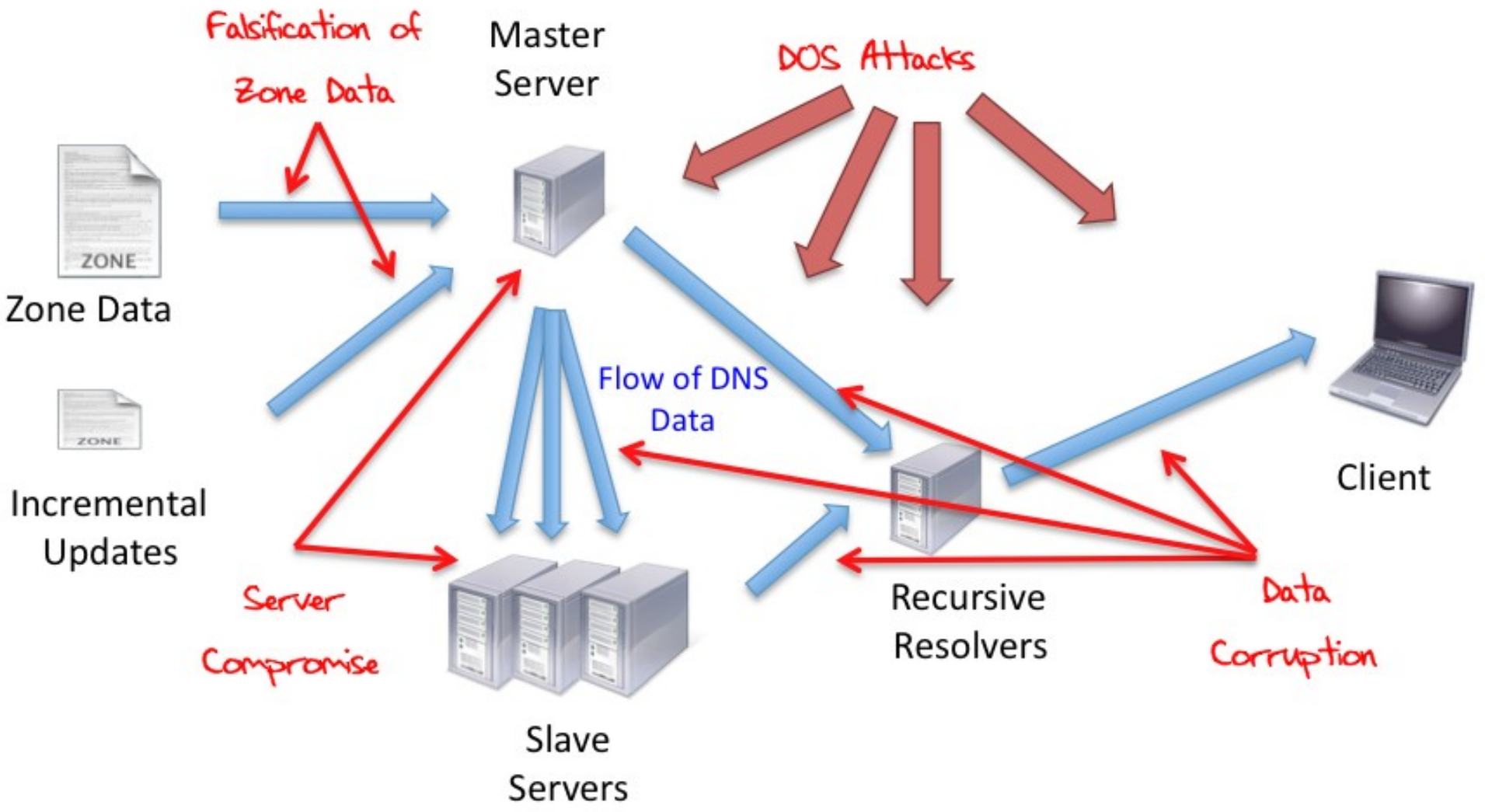
- Limitar el número de transferencias de zona por servidor de nombres (sentencia transfers-per-ns)
- Limitar el número de transferencias concurrentes (sentencia transfers-in)
- Limitar la duración de las transferencias (sentencia max_transfer-time-in)
- Evitar el uso de los registros WKS y HINFO
- Ejecutar BIND “enjaulado” (chrooted) en una estructura de directorios como medida de protección frente a posibles ataques remotos
- En BINDv8, usar la sentencia use-id-pool yes; para dificultar el DNS Spoofing (BINDv9 lo usa por default)
- En BINDv8 los canales de control inet son inseguros porque el servidor de nombres usa las direcciones IP para autenticar los comandos, por lo que se recomienda evitar el uso de dichos canales y en su lugar usar simplemente Unix
- En BINDv9 los canales de control inet son más seguros porque el servidor de nombres usa criptografía para autenticar los comandos, aunque de cualquier manera se recomienda restringir las direcciones IP que pueden enviar comandos, o mejor aún, restringir a algunas llaves TSIG
- Restringir las actualizaciones dinámicas porque toman casi todo el control de la zona, ya que es posible borrar cada registro en la zona, excepto el RR SOA y un RR NS y agregar prácticamente cualquier otro RR. Es recomendable restringir las actualizaciones dinámicas a sólo algunas direcciones IP o mejor aún, a algunas llaves TSIG
- Enviar los registros de syslog del servidor de DNS a un servidor de logs, de manera que pueda usarse una herramienta como LogSentry para revisar dichas bitácoras en busca de actividad sospechosa

Contramedidas: Seguridad en las zonas y en las transacciones

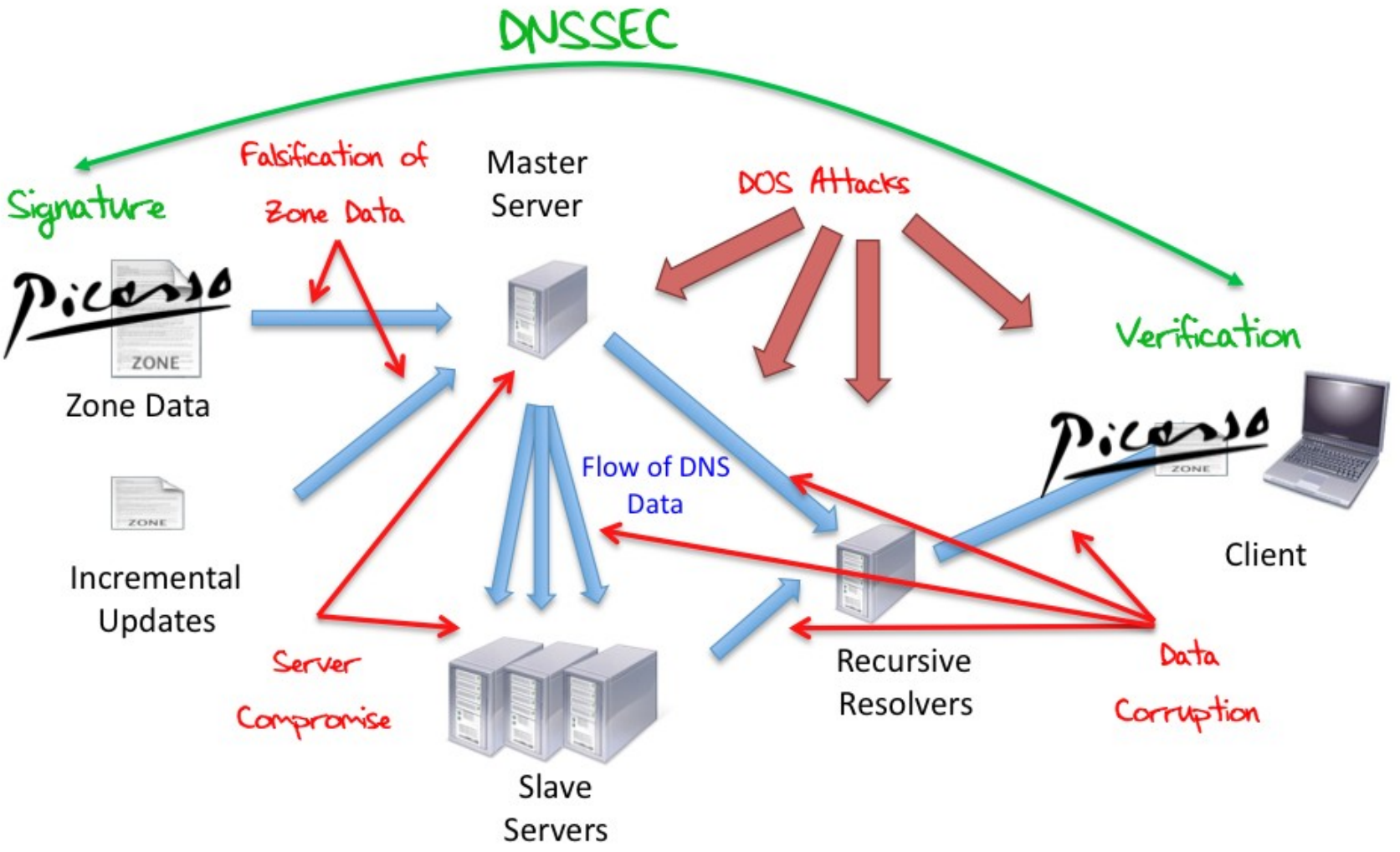
- Las contramedidas para implementar seguridad en las transacciones entre servidores de nombres se implementan mediante lo que se conoce como Transaction Signatures o TSIG
- Las contramedidas para implementar zonas de forma segura consisten en la implantación de las extensiones de seguridad DNSSEC



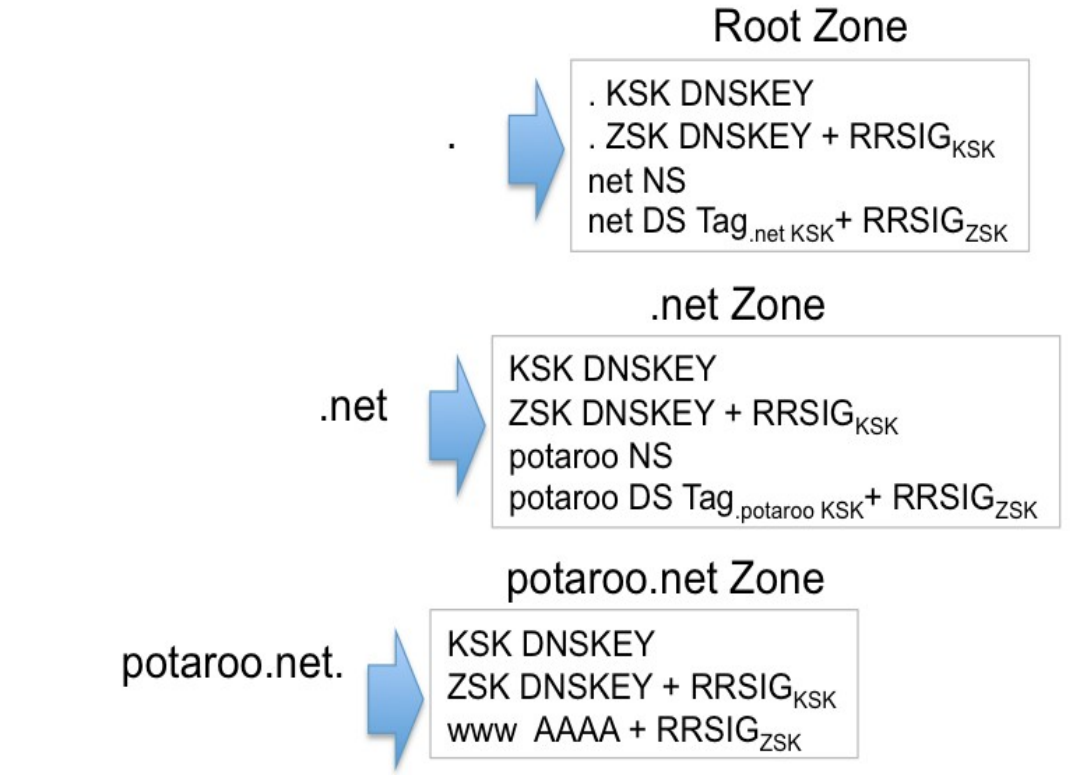
DNSSEC


Falsification of
Zone Data
Master
Server
DOS Attacks
ZONE
Zone Data
ZONE
Incremental
Updates
Flow of DNS
Data
Client
Server
Compromise
Recursive
Resolvers
Data
Corruption
Slave
Servers

DNSSEC

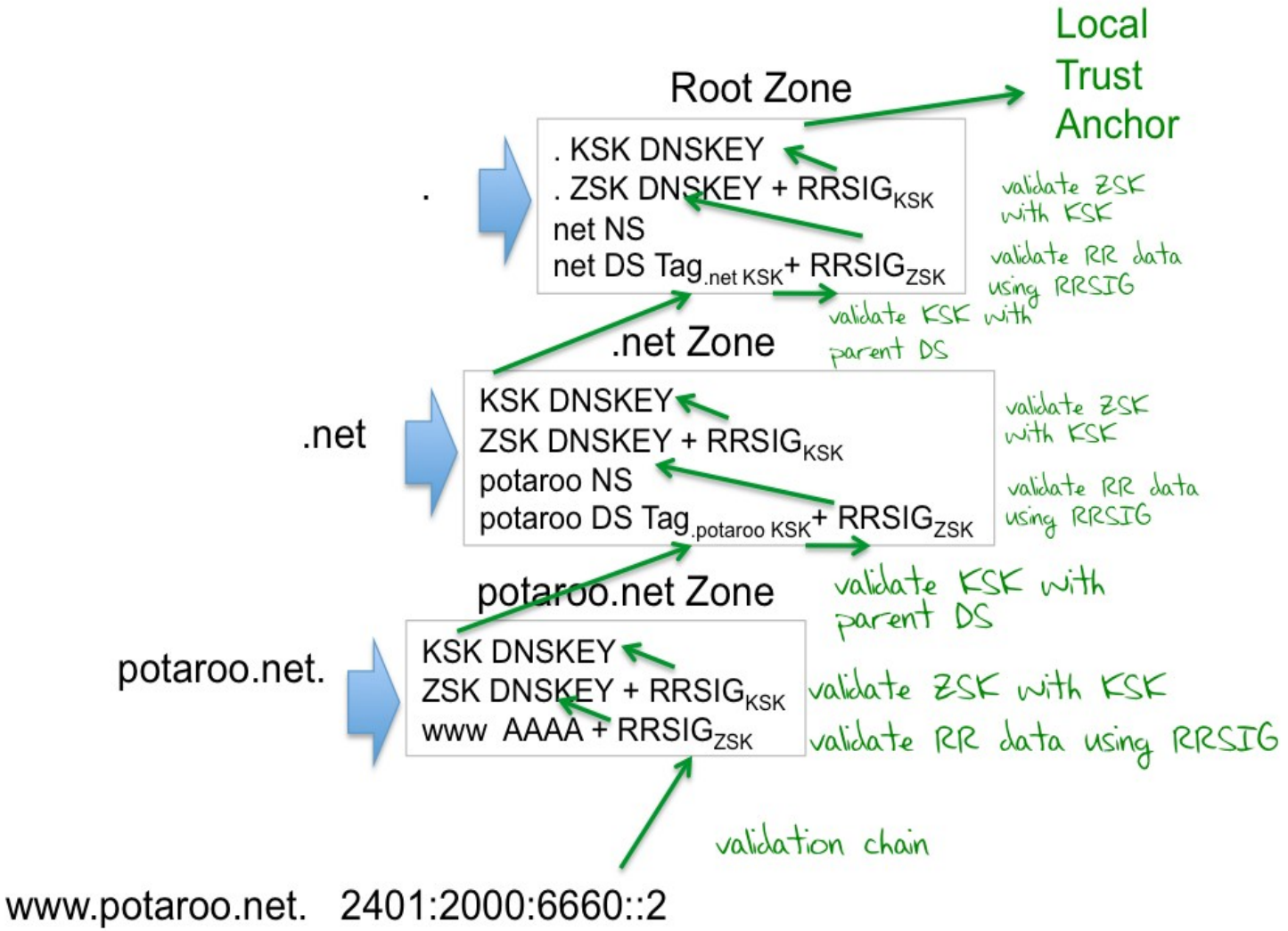

DNSSEC
Signature
Falsification of Zone Data
Master Server
DOS Attacks
Verification
Picasso
ZONE
Zone Data
Incremental Updates
Flow of DNS Data
Client
Server Compromise
Slave Servers
Recursive Resolvers
Data Corruption

DNSSEC



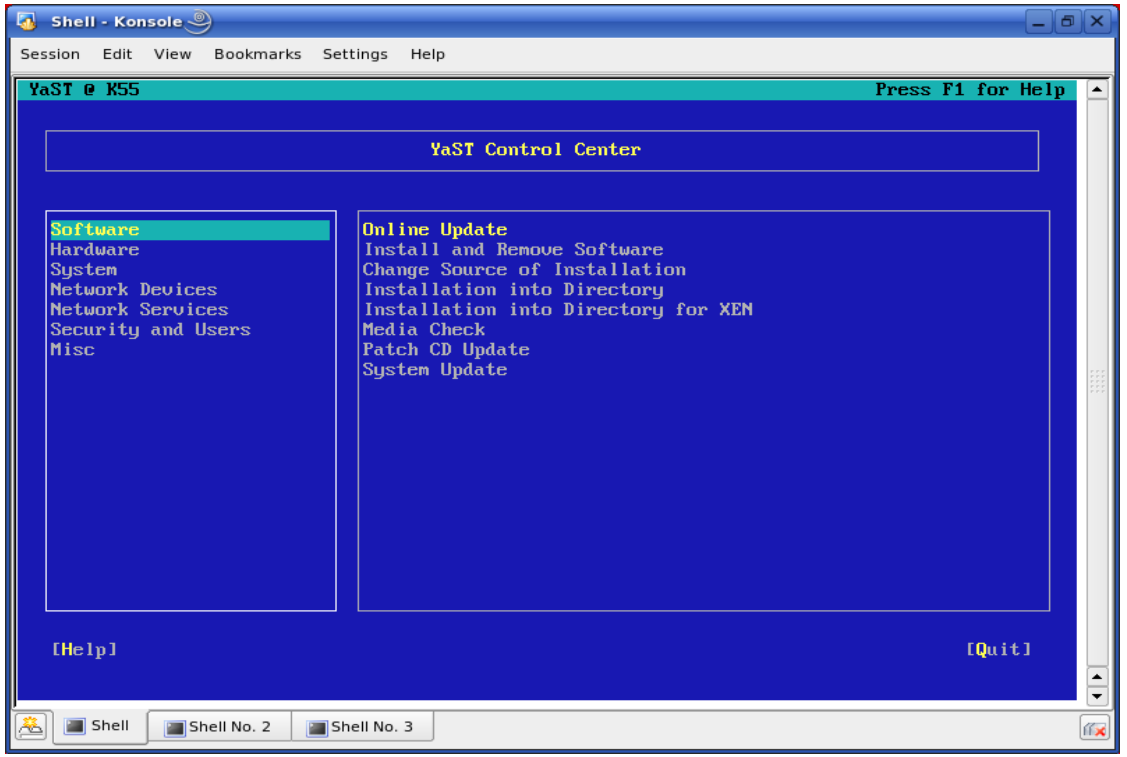
www.potaroo.net. 2401:2000:6660::2

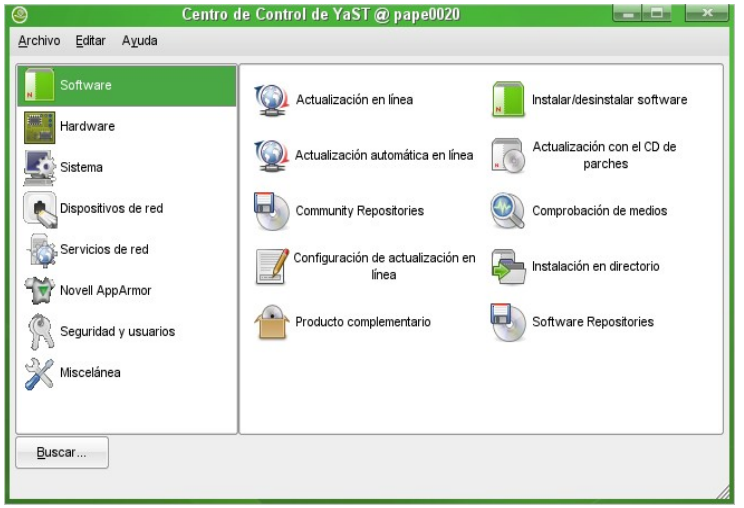
DNSSEC


Local
Trust
Anchor
Root Zone
.
. KSK DNSKEY
. ZSK DNSKEY + RRSIG$_{KSK}$
net NS
net DS Tag$_{.net\ KSK}$+ RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validate KSK with parent DS
.net Zone
.net
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
potaroo NS
potaroo DS Tag$_{.potaroo\ KSK}$+ RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validate KSK with parent DS
potaroo.net Zone
potaroo.net.
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
www AAAA + RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validation chain
www.potaroo.net. 2401:2000:6660::2

Instalación de BIND

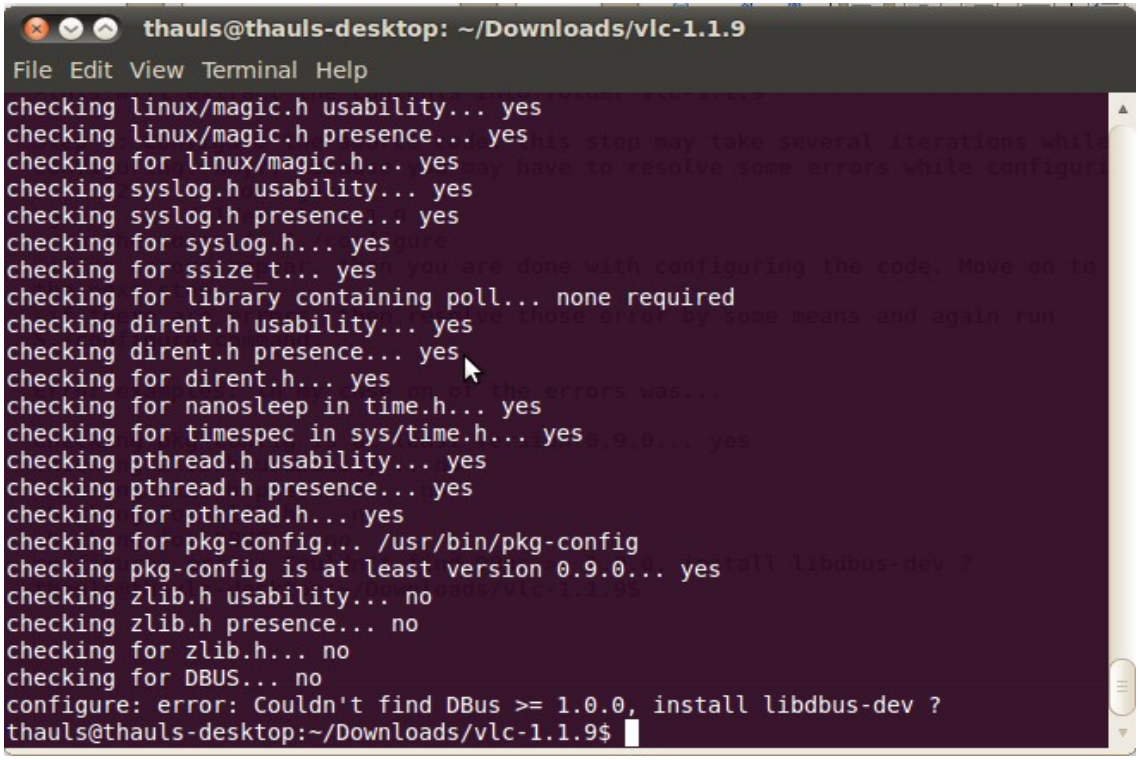
- Repositorio





Instalación de BIND

- Compilar Código fuente



Instalación de BIND

- Código Fuente

  http://www.isc.org/downloads/
- Dependencias
  - gcc
  - make
  - libopenssl0_9_8
  - libopenssl-devel

Instalación de BIND

- Opciones de compilación

  # ./configure --with-openssl=yes --enable-openssl-hash --enable-newstats

- Comparación de versiones

  # named -v

  BIND 9.9.4-rpz2.13269.14-P2 (Extended Support Version)

  # /usr/local/sbin/named -v

  BIND 9.9.5-W1 (Extended Support Version)

Configuración básica de un DNS

- Configuración de un servidor de nombres maestro
- Configuración del resolver
- Configuración de un servidor de nombres esclavo

Configuración básica de un DNS

Campo SOA	Descripción	Valor Sugerido por el RFC 1912/2308
SERIAL NUMBER	Este número debe ser incrementado cada vez que se hace un cambio a la zona, en el formato AAAAMMDDnn.	N/A
REFRESH	Este valor determina con que frecuencia el servidor de nombres esclavo pregunta al maestro si hay actualizaciones.	1200 a 43200 segundos (20 minutos a 12 horas)
RETRY	Este valor determina cuanto tiempo debe esperar el servidor de nombres esclavo para intentar contactar nuevamente al maestro en caso de alguna falla de comunicación.	120 a 7200 segundos (2 minutos a 2 horas)
EXPIRE	Este valor determina cuanto tiempo debe esperar el servidor de nombres esclavo para considerar inválidos sus datos (expirar la zona) si no puede contactar al maestro.	1209600 a 2419200 segundos (2 a 4 semanas)
MINIMUM	Este valor determina el caché de respuestas negativas (registros que no existen).	3600 a 86400 segundos (1 a 24 horas)

Configuración básica de un DNS

- Sentencias de control de BIND

$ORIGIN – Cambia el origen de un archivo de zona (el origen es el nombre de dominio que se agrega a todos los nombres que no terminan con .)

$INCLUDE – Inserta un nuevo archivo en el archivo de zona actual

$TTL – Define el TTL (Time-To-Live) para la zona (el tiempo que cualquier servidor de nombres debe mantener los registros definidos en el archivo de zona)

Ejemplo:

$ORIGIN my.example.com.

$INCLUDE db.my.example.com

Configuración básica de un DNS

- Archivo de zona ejemplo (example.com.zone)

```
$ORIGIN .
$TTL 1d
example.com  IN SOA  ns.example.com. root.ns.example.com. (
                        2014020401   ; numero de serie
                        1200         ; refresh (20 minutos)
                        3600         ; retry (1 hora)
                       1209600       ; expire (2 semanas)
                        3600         ; minimum (1 hora)
                        )
                    A     192.168.0.1
                    NS     ns.example.com.
                    MX     10     mail.example.com.
$ORIGIN example.com.
ns                  A     192.168.0.2
mail                A     192.168.0.3
www             CNAME example.com.
```

Configuración básica de un DNS

- Archivo de zona inversa ejemplo (192.168.0.zone)

```
$ORIGIN .
$TTL 1d
0.168.192.in-addr.arpa  IN SOA  ns.example.com. root.ns.example.com. (
                        2014020401   ; numero de serie
                        1200         ; refresh (20 minutos)
                        3600         ; retry (1 hora)
                        1209600      ; expire (2 semanas)
                        3600         ; minimum (1 hora)
                        )
                    NS      ns.example.com.
$ORIGIN 0.168.192.in-addr.arpa.
8               PTR     security.example.com.
```

Configuración básica de un DNS

- Archivo named.conf ejemplo:

```
options {
        directory "/var/lib/named";
        pid-file "/var/run/named.pid";
};

zone "." {
    type hint;
    file "root.hint";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "127.0.0.zone";
};

zone "example.com" {
        type master;
        file "example.com.zone";
};

zone "0.168.192.in-addr.arpa" {
    type master;
    file "192.168.0.zone";
};
```

Configuración básica de un DNS

- Configuración del resolver

  El archivo de configuración del resolver es */etc/resolv.conf*

  Las directivas más comunes son:

  nameserver - Especifica la dirección IP del servidor de nombres que va a consultar el resolver
  domain - Especifica el nombre de dominio local
  search - Define la lista de búsqueda de dominios para resolver nombres (limitado a 6 dominios)

  Ejemplo:

  ```
  # cat /etc/resolv.conf
  domain example.com
  search example.com my.example.com your.example.com
  nameserver 192.168.0.1
  ```

Configuración básica de un DNS

- Archivo named.conf ejemplo (slave)

```
options {
        directory "/var/lib/named/slave";
        pid-file "/var/run/named.slave.pid";
};

zone "." {
    type hint;
    file "root.hint";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "127.0.0.zone";
};

zone "example.com" {
       type slave;
       file "example.com.zone.slave";
       masters { 192.168.122.65; };
};

zone "0.168.192.in-addr.arpa" {
    type slave;
    file "192.168.0.zone.slave";
    masters { 192.168.122.65; };
};
```

Herramientas de Consultas al DNS

- nslookup
- host
- dig

Herramientas de Consultas al DNS

- nslookup
  - La utilería nslookup permite consultar servidores de nombres y tiene dos modos de uso: modo interactivo y modo no interactivo.
    - El **modo interactivo** permite al usuario solicitar a servidores de nombres información detallada sobre hosts o dominios.
    - El **modo no interactivo** se usa para solicitar información sobre un solo host o dominio.
  - *Esta utilería se considera obsoleta, por lo que se recomienda usar en su lugar las utilería host o dig*.

    La sintaxis de esta utilería es:

    nslookup –opciones host_a_resolver –servidor_nombres

Herramientas de Consultas al DNS

- Se ingresa al modo interactivo en los siguientes casos:
  - Cuando el comando es ejecutado sin argumentos (se usa el servidor de nombres configurado por omisión)
  - Cuando el primer argumento es un guión (-) y el segundo argumento especifica al servidor de nombres
- El modo no interactivo se usa cuando el nombre de dominio o dirección que desea resolverse se especifica como el primer argumento. El segundo argumento, opcional, especifica el servidor de nombres a consultar.
- Algunos de los comandos más útiles del modo interactivo de nslookup son los que aparecen en la siguiente página. Estos comandos pueden cancelarse tecleando CTRL+C, y para salir de este modo debe teclearse CTRL+D o el comando exit.

Herramientas de Consultas al DNS

Comando	Descripción	Sintáxis	Opciones
ls	Lista la información disponible para el dominio y de forma opcional guarda la salida en un nombre de archivo	ls [opción] dominio [>> nombre_archivo]	-t tipo_consulta Lista todos los registros que coincidan con el tipo especificado
			-d Lista todos los registros del dominio
			-h Lista información sobre los procesadores y sistema operativo de los hosts en el dominio
set	Cambiar el estado de la información que se solicita	set palabra_clave [=valor]	class=valor
			Cambia la clase de consulta a in, chaos, hesiod o any
			(in)
			[no]debug
			Habilita / deshabilita el modo de depuración (nodebug)
			[no]d2
			Habilita / deshabilita el modo exhaustivo de depuración (nod2)
			port=valor
			Cambia el puerto TCP/UDP del servidor de nombres (53)
			type=valor
			Cambia el tipo de consulta de información a ns, mx, a, cname, ptr, soa, etc. (a)
			[no]recurse
			Indica al servidor de nombres que consulte a otros servidores si no conoce la respuesta (recurse)

Herramientas de Consultas al DNS

- Host
  - Esta utilería básicamente sirve para resolver nombres de dominio en direcciones IP o viceversa, pero las opciones que se describen abajo permiten realizar consultas más elaboradas.
  - Su sintaxis es:

    host [-l] [-v] [-w] [-r] [-d] [-t tipo_de_consulta] [-a] host_a_resolver [servidor_de_nombres]
  - El argumento host_a_resolver es el host que desea resolverse y puede ser un nombre de dominio o una dirección IP; si es una dirección IP el programa realiza una consulta inversa. Por omisión, el programa interpreta este argumento como una dirección IP, pero si falla lo interpreta como un nombre de dominio.
  - El segundo argumento (servidor_de_nombres) es opcional y permite especificar el servidor de nombres al que debe consultarse. Si no se especifica, se usa el servidor de nombre configurado por omisión

Herramientas de Consultas al DNS

Las opciones más importantes de esta utilería son las siguientes:

- **-v**

  Usar el formato detallado (verbose) en la impresión de salida

- **-r**

  Deshabilitar la recursión en la consulta, lo cuál significa que el servidor de nombres regresará únicamente los datos que tenga en su base de datos (caché), sin preguntar a otros servidores de nombres

- **-d**

  Habilitar el modo de depuración, muestra el detalle de las transacciones de red

- **-t** tipo_de_consulta

  Especificar un tipo de consulta en particular. Entre los tipos de consulta se encuentran “a”, “ns” “cname”, “soa”, “mx” y “any”. Este último tipo es un comodín para todos los tipos de consulta. Los tipos deben ser escritos en minúsculas. El comportamiento por omisión es buscar primero registros “a” y luego “mx”

- **-l**

  Listar un dominio completo; esta opción no devuelve nada si el servidor de nombres autoritario restringe las transferencias de zona

Herramientas de Consultas al DNS

- Ejemplos de uso:

  $ host www.example.com

  $ host -v www.example.com

  $ host -r www.example.com

  $ host -d www.example.com

  $ host -l example.com

  $ host -t mx example.com

  $ host -t soa example.com

Herramientas de Consultas al DNS

- **dig**
  - La utilería dig (*domain information groper*) es un comando muy flexible que puede ser usado para consultar información de los servidores de nombres. Esta utilería tiene dos modos de operación: el modo simple interactivo útil para una única consulta y el modo batch o no-interactivo que ejecuta todas las consultas definidas en una lista.
  - La sintaxis general de esta utilería es:

    dig [@servidor_de_nombres] nombre_de_dominio [<tipo_de_consulta>] [<clase_de_la_consulta>] [+<opcion_consulta>] [-<opcion_dig>]

dónde:

- servidor_de_nombres es un nombre de dominio o una dirección IP; si no se especifica, se usa el servidor de nombres configurado por omisión. Si se especifica un nombre de dominio, es necesario contar con un resolver de DNS, tal como BIND o un servidor de nombres definido en el archivo /etc/resolv.conf
- nombre_de_dominio es el nombre de dominio o la dirección IP para la cuál se está solicitando información
- tipo_de_consulta es el tipo de información que se está solicitando; por omisión es del tipo “a” (RR A). Otros tipos reconocidos son: “any” (toda la información del dominio), “mx” (servidores de correo para el dominio), “ns” (servidores de nombres), “soa” (zona de autoridad) y “axfr” (transferencias de zona)
- clase_de_la_consulta es la clase de red solicitada en la consulta; por omisión es “in” (Internet)

Herramientas de Consultas al DNS

Palabra Clave	**Significado**
[no]debug	Habilitar o deshabilitar el modo de depuración
[no]d2	Habilitar o deshabilitar el modo extendido de depuración [nod2]
[no]recursive	Usar o no el modo recursivo [rec]
[no]trace	Rastrear la delegación desde el dominio raíz hacia abajo
[no]dnssec	Solicitar registros DNSSEC

Herramientas de Consultas al DNS

- Herramientas de consultas de DNS desde Internet
  - www.dnsstuff.com
  - www.kloth.net
  - network-tools.com
  - http://www.domaintools.com/research/dns/
  - mxtoolbox.com

Agenda
Día 3

- Configuración avanzada de un DNS
- Diseño
- Transaction Signatures (TSIG)
- DNS Dinámico

Configuración avanzada de un DNS

- Listas de Control de Acceso (ACLs)
- Seguridad
  - Ocultamiento de la versión de BIND
  - Restricción de las transferencias de zonas
  - Restricción de las consultas iterativas
  - Restricción de las consultas recursivas
  - Otras directivas interesantes
- Logging
- BIND chrooted
- Control de BIND con RNDC

Configuración avanzada de un DNS

- Listas de Control de Acceso (ACL)
  - Son listas de elementos que especifican direcciones IP
  - Cada elemento puede ser una dirección IP, un prefijo de IP o una lista de nombres
  - Un prefijo de IP tiene el formato red_en_formato_octal/bits_en_la_mascara

- Ejemplos de prefijos de IP:
  - 172.18/16 (red 172.18.0.0 con máscara de red 255.255.0.0)
  - 150.23.15/24 (red 150.23.15.0 con máscara de red 255.255.255.0)
  - 15/8 (red 15.0.0.0 con máscara de red 255.0.0.0)

- Formato de una sentencia de ACL:

  acl nombre_acl { lista_ips; };

- Ejemplo de ACL:

  acl “red_cliente” { 172.18/16; };

Configuración avanzada de un DNS

- Existen 4 palabras reservadas para las ACLs
  - none (“ninguna dirección IP”)
  - any (“cualquier dirección IP”)
  - localhost (“cualquiera de las direcciones IP de localhost”)
  - localnets (“cualquiera de las redes configuradas en localhost”)

Configuración avanzada de un DNS

- Ocultar la versión de BIND

  Se agrega la sentencia version a la sección options del archivo de configuración named.conf:

```
options {
        directory "/var/named/master";
        pid-file "/var/named/master/named.pid";
        version "Oculta deliberadamente";
};
```

Configuración avanzada de un DNS

- Restringir las transferencias de zona

Se agrega la sentencia allow-transfer a la sección options del archivo de configuración named.conf:

```
options {
    directory "/var/named/master";
    pid-file "/var/named/master/named.pid";
    version "Oculta deliberadamente";
    allow-transfer { none; };
};
```

Configuración avanzada de un DNS

- Restringir las consultas iterativas (no recursivas)

Se agrega la sentencia allow-query a la sección options del archivo de configuración named.conf:

```
options {
    directory "/var/named/master";
    pid-file "/var/named/master/named.pid";
    version "Oculta deliberadamente";
    allow-transfer { none; };
    allow-query { localnets; };
};
```

Configuración avanzada de un DNS

- Restringir las consultas recursivas

Se agrega la sentencia allow-recursion a la sección options del archivo de configuración named.conf:

acl “red_cliente” { 172.18/16; };

```
options {
    directory "/var/named/master";
    pid-file "/var/named/master/named.pid";
    version "Oculta deliberadamente";
    allow-transfer { none; };
    allow-query { localnets; };
    allow-recursion { “red_cliente”; };
};
```

Configuración avanzada de un DNS

- Restringir el número de clientes de consultas recursivas

Se agrega la sentencia recursive-clients a la sección options del archivo de configuración named.conf:

```
acl “red_cliente” { 172.18/16; };
options {
    directory "/var/named/master";
    pid-file "/var/named/master/named.pid";
    version "Oculta deliberadamente";
    allow-transfer { none; };
    allow-query { localnets; };
    allow-recursion { “red_cliente”; };
    recursive-clients 2000;
};
```

Configuración avanzada de un DNS

- Ignorar las consultas de ciertos clientes

  - Se agrega la sentencia blackhole a la sección options del archivo de configuración named.conf:

    /* No consultar y no responder a las consultas de las redes privadas */

    acl “redes_no_ruteables“ { 10/8; 172.16/12; 192.168/16; };

    options {

    directory "/var/named/master";

    pid-file "/var/named/master/named.pid";

    blackhole { “redes_no_ruteables”; };

    };

Configuración avanzada de un DNS

Logging

- Existen 2 conceptos importantes en el logging de BIND: canales (channels) y categorías (categories)
  - El canal especifica a donde van los datos de log (a syslog, a un archivo, a la salida de error estándar de named, etc.)
  - La categoría especifica qué datos son los que se registran
- Cada categoría puede ser enviada a uno o a varios canales
- Los canales permiten filtrar la severidad del mensaje (critical, error, warning, notice, info (default), debug y dynamic)

Configuración avanzada de un DNS

Logging

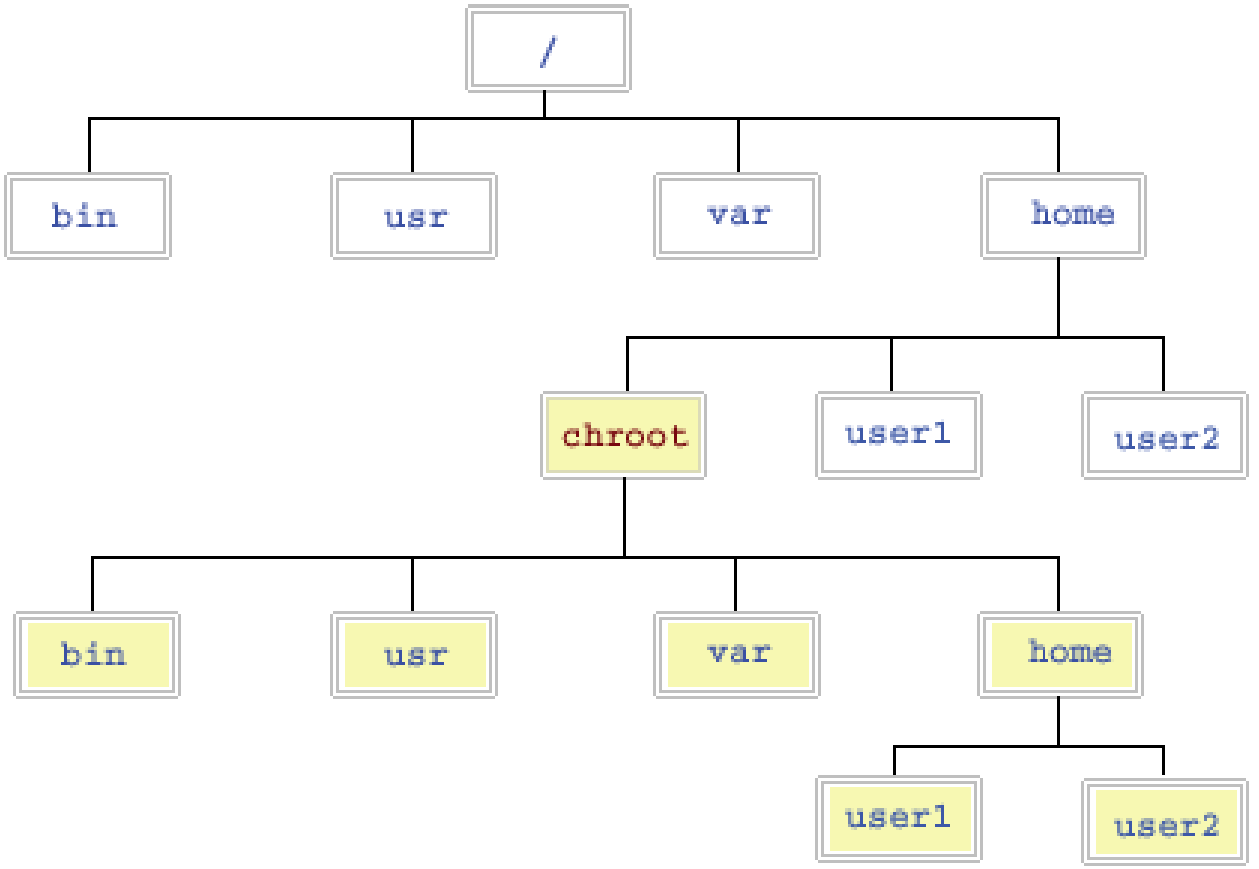
```
logging {
        channel archivo {
                file "named.log" versions 9 size 10M;
                print-time yes;
                print-category yes;
                print-severity yes;
        };
        category default { archivo; default_syslog; };
};
```

Configuración avanzada de un DNS

Enjaular el servicio de DNS (/chroot)

- *chroot* es una operación que invoca un proceso, cambiando para este y sus hijos el directorio raíz del sistema.
- Comúnmente, el entorno virtual creado por chroot a partir de la nueva raíz del sistema se conoce como "jaula chroot".
- Al usar "chroot" para invocar un proceso, se impedirá al mismo y a sus procesos hijos acceder por su nombre a ningún fichero que esté por encima del nuevo directorio raíz.
- Debido a que los programas esperan encontrar en lugares determinados su espacio de almacenamiento, los archivos de configuración o sus bibliotecas de enlace dinámico, entre otros, preparar una jaula chroot implica también incluir dichos recursos dentro de ella.
- Los programas tienen permitido llevarse descriptores de archivos abiertos (sean archivos físicos, tuberías, o conexiones de red) dentro de la jaula, lo cual puede simplificar el diseño haciendo innecesario dejar archivos funcionales dentro del directorio chroot. Esto también funciona como un sistema de capacidades simple, en el cual, al programa se le otorga acceso explícito a los recursos externos del chroot basado en los descriptores que puede llevar a su interior.

Configuración avanzada de un DNS



Configuración avanzada de un DNS

- **Ejecutar BIND en un ambiente chrooted (“enjaulado”)**

- Crear un grupo llamado bind con GID 53:

  # groupadd –g 53 bind

- Crear un usuario llamado bind con UID 53, grupo bind, descripción “DNS BIND”, directorio HOME /chroot, un shell falso y con su contraseña bloqueada:

  # useradd –u 53 –g bind –d /chroot –s /bin/false –c “DNS BIND” bind

  # passwd –l bind

- Crear la estructura de directorios siguiente:

  /chroot/etc

  /chroot/var/named

  /chroot/var/run

  /chroot/lib

  /chroot/lib64

  # mkdir -p /chroot/etc /chroot/var/named /chroot/var/run /chroot/lib /chroot/lib64

Configuración avanzada de un DNS

- **Copiar los archivos /etc/localtime /etc/profile /etc/named.conf al nuevo directorio /chroot/etc:**

  # cp /etc/localtime /etc/profile /etc/named.conf /chroot/etc

- **Copiar los engines de ssl al directorio enjaulado**

  # cp /lib64/engines/libgost.so /chroot/lib64/engines/

  # cp /lib/engines/libgost.so /chroot/lib/engines/

- **Copiar los archivos de zona al directorio enjaulado**

  # cp /var/lib/named/* /chroot/var/named/

- **Asignar los permisos 700 de forma recursiva al sistema de archivos /chroot, 600 para los archivos y asignar como propietario y grupo a bind:**

  # chmod -R 700 /chroot

  # find /chroot –type f | xargs chmod 600

  # chown –R bind:bind /chroot

- **Iniciar named en la “jaula”:**

  # named –u bind –t /chroot -4 -d 2

Configuración avanzada de un DNS

Control de BIND con RNDC

- El programa named, al igual que cualquier programa en Unix, puede ser controlado mediante señales a nivel sistema operativo que le indican acciones a tomar (como por ejemplo, hacer que vuelva a leer su archivo de configuración).
- Sin embargo, existe un número limitado de señales disponibles y la función de éstas también se encuentra limitada. En BINDv9, es posible enviar mensajes al servidor de nombres mediante un canal de control especial, que es un puerto TCP en que el servidor “escucha” en espera de mensajes.
- Esto puede lograrse usando un programa llamado rndc (remote name daemon control). BINDv9 usa la sentencia controls para determinar de qué manera el servidor de nombres va a escuchar los mensajes dirigidos a él. El puerto por omisión que usa rndc es el 953/TCP.

Configuración avanzada de un DNS

Configuración de RNDC

- Crear una llave compartida codificada en formato base 64 usando el comando rndc-confgen (este comando genera el archivo /chroot/etc/rndc.key, que contiene una llave secreta de 512 bits de longitud):

  # /usr/local/sbin/rndc-confgen -a -b 512 -c /chroot/etc/rndc.key

  wrote key file "/chroot/etc/rndc.key"

- Esta llave también puede ser elegida usando el comando mmencode y creando manualmente el archivo /chroot/etc/rndc.key, por ejemplo:

  $ mmencode

  mi_llave_secreta_para_rndc

  bWlfbGxhdmVfc2VjcmV0YV9wYXJhX3JuZGMK

- El archivo /chroot/etc/rndc.key generado contiene las siguientes líneas:

```
key "rndc-key" {
        algorithm hmac-md5;
        secret
"EDwdjtBO3qqK5Mo91+TCOz3YSz9709Y5ba1JbDIFFLeq/4hKcS7dAj/2s2wB2ANVv9HNzVNSO3E9aZr38q3CJQ==";
};
```

Configuración avanzada de un DNS

- Crear el archivo /chroot/etc/rndc.conf, conteniendo las siguientes líneas:

```
options {
        default-server localhost;
        default-key "llave_para_rndc";
};
key "llave_para_rndc" {
    algorithm hmac-md5;
    secret
"EDwdjtBO3qqK5Mo91+TCOz3YSz9709Y5ba1JbDIFFLeq/4hKcS7dAj/2s2wB2
ANVv9HNzVNSO3E9aZr38q3CJQ==";
};
```

- Eliminar el archivo /chroot/etc/rndc.key generado

Configuración avanzada de un DNS

- Agregar las siguientes líneas al archivo /chroot/etc/named.conf:

```
controls {
    inet 127.0.0.1 port 953 allow { localhost; } keys
{ "llave_para_rndc"; };
    };
    key "llave_para_rndc" {
    algorithm hmac-md5;
    secret "EDwdjtBO3qqK5Mo91+TCOz3YSz9709Y5ba1JbDIFFLeq/4hKcS7dAj/2s2wB2ANVv9HNzVNSO3E9aZr38q3CJQ==";
};
```

- Tanto los nombres de las llaves como las llaves mismas deben coincidir en ambos archivos (rndc.conf y named.conf)

Configuración avanzada de un DNS

- Reiniciar el programa named:

```
# ps auxf | grep named
    bind 25848     1  0 21:02:21 ?        0:00 /usr/local/sbin/named -u bind -t /chroot
# kill -9 25848
# /usr/local/sbin/named -u bind -t /chroot
# ps auxf | grep named
    bind 25874     1  0 22:10:39 ?        0:00 /usr/local/sbin/named -u bind -t /chroot
```

- Crear una liga simbólica en /etc/rndc.conf que apunte al archivo /chroot/etc/rndc.conf:

```
# ln -s /chroot/etc/rndc.conf /etc/rndc.conf
# ls -l /etc/rndc.conf
lrwxrwxrwx   1 root     other         21 Feb 5 22:16 /etc/rndc.conf -> /chroot/etc/rndc.conf
# chown bind:bind /chroot/etc/rndc.conf
# chmod 600 /chroot/etc/rndc.conf
# ps –fea | grep named
    bind 25874     1  0 22:10:39 ?        0:00 /usr/local/sbin/named -u bind -t /chroot
```

Configuración avanzada de un DNS

- Revisar el estado el servidor de nombres:

  # /usr/local/sbin/rndc status

  number of zones: 3

  debug level: 0

  xfers running: 0

  xfers deferred: 0

  soa queries in progress: 0

  query logging is OFF

  recursive clients: 0/1000

  tcp clients: 0/100

  server is up and running

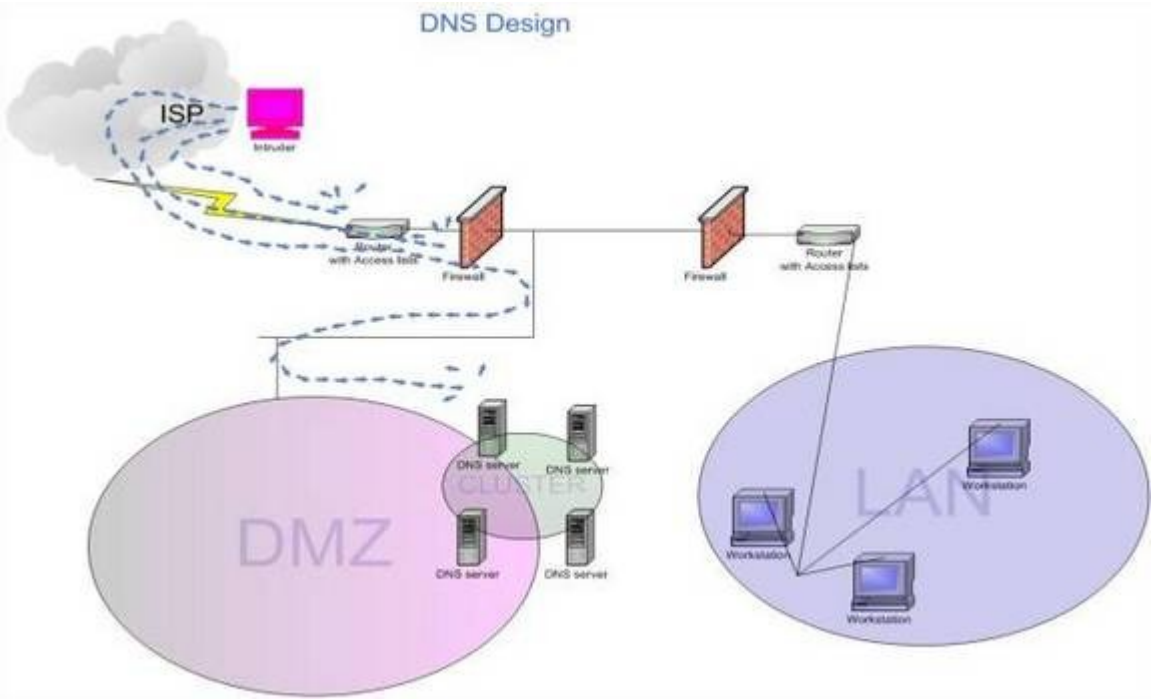
Configuración avanzada de un DNS

- Opciones de RNDC

Opción	**Efecto**
halt	Inmediatamente detiene el programa **named**
querylog	Activa el registro de las consultas hechas al servidor de nombres
reload	Indica al servidor de nombres que cargue nuevamente los archivos de zona
stats	Guarda las estadísticas del programa **named** en el archivo **/var/named/named.stats**
dumpdb	Guarda el caché del *DNS* en el archivo **/var/named/named_dump.db**
flush	Limpia la memoria caché de **named**
status	Muestra el estado del programa **named**
stop	Detiene limpiamente el programa **named**

Diseño

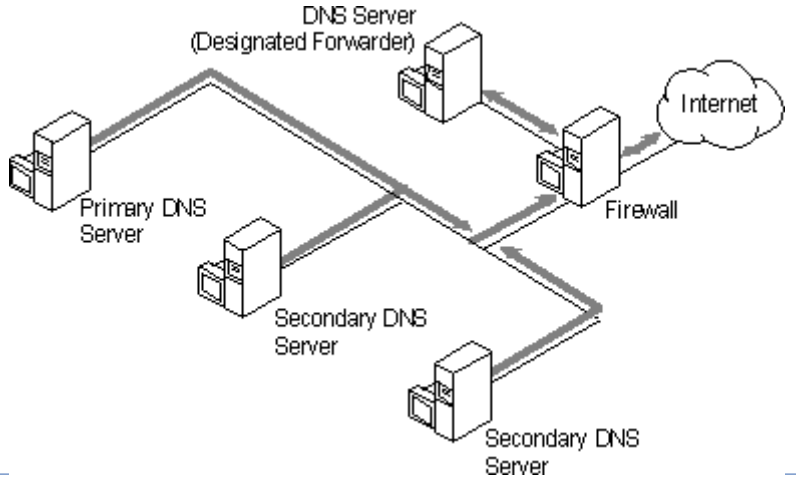
- Forwarding
- DNS Split



Diseño

Forwarding

- Se usa para limitar el tráfico de DNS, ya sea porque la conexión a la red se factura por volumen de tráfico, porque se cuenta con un enlace de red muy lento, o por alguna otra razón.
- También puede usarse para redireccionar las consultas de varios servidores de nombres sin acceso a Internet a un servidor de nombres que si tenga acceso (a este último servidor se le conoce como forwarder).
- Debe evitarse encadenar los forwarders (es decir, cuando un NS A hace forward a B y B a C)



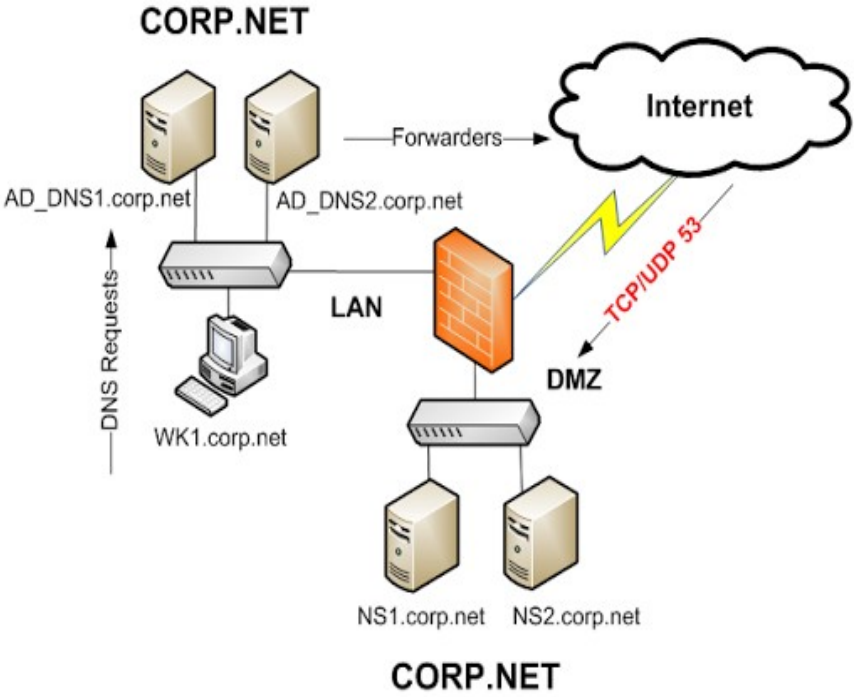
Diseño

Se agrega la sentencia forwarders a la sección options del archivo de configuración named.conf:

```
options {
    directory "/var/named/master";
    pid-file "/var/named/master/named.pid";
    forwarders { 10.16.143.248 port 53; };
};
```

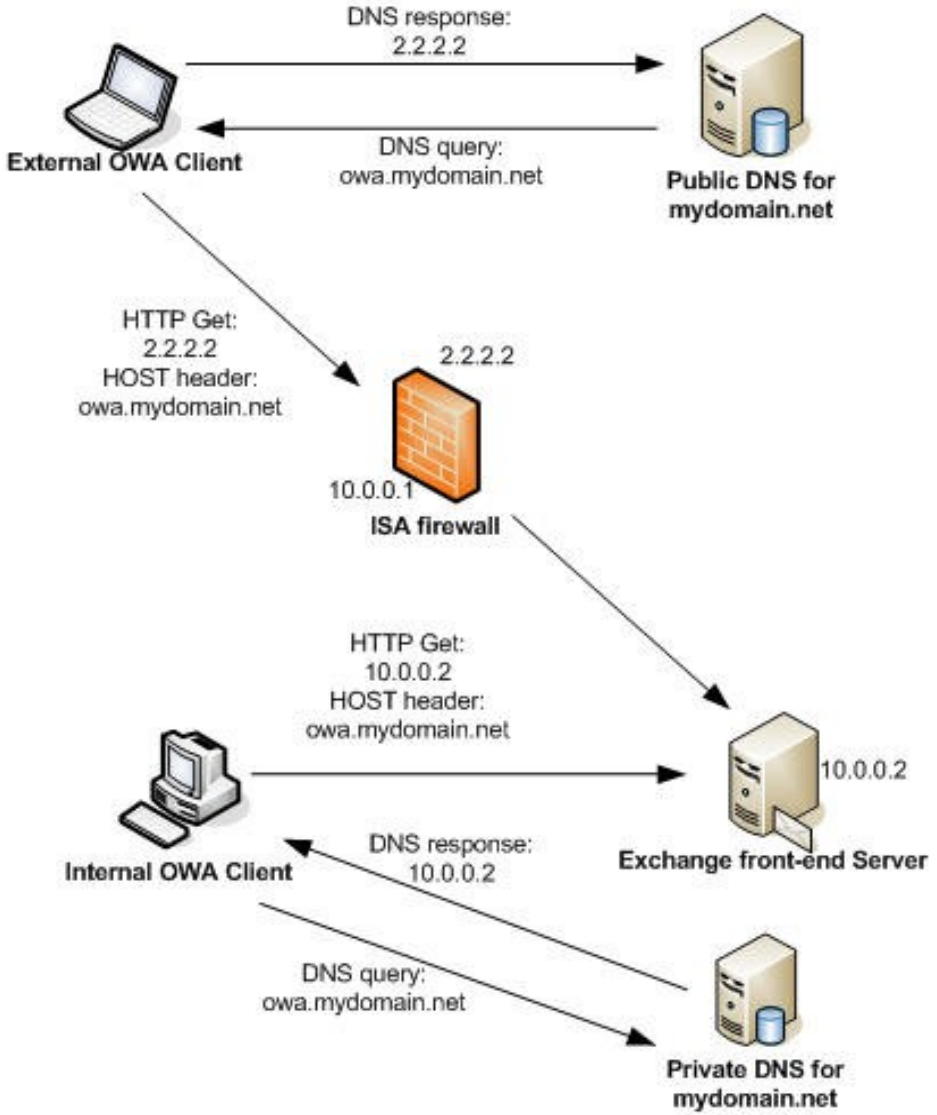
Diseño

- El diseño “Split DNS” consiste en la separación de los servidores de nombres internos de los externos.
- Los servidores de nombres internos únicamente contienen registros de la red interna y los servidores de nombres externos sólo contienen registros ajenos a la red interna.
- En una infraestructura Split DNS se crean 2 zonas para el mismo dominio: una para ser usada por la red interna y la otra para ser usada por la red externa.



- El Split DNS dirige las consultas de los hosts internos al servidor de nombres interno y dirige las consultas de los hosts externos a un servidor de nombres externo.
- Los intrusos buscan servidores de nombres que no usan este diseño y publican los hosts internos en Internet.

Diseño


DNS response:
2.2.2.2
External OWA Client
DNS query:
owa.mydomain.net
Public DNS for
mydomain.net
HTTP Get:
2.2.2.2
HOST header:
owa.mydomain.net
2.2.2.2
10.0.0.1
ISA firewall
HTTP Get:
10.0.0.2
HOST header:
owa.mydomain.net
10.0.0.2
Internal OWA Client
DNS response:
10.0.0.2
Exchange front-end Server
DNS query:
owa.mydomain.net
Private DNS for
mydomain.net

Transaction Signatures (TSIG)

Transaction Signatures (TSIG)

- Descripción
- Consideraciones de seguridad
- Configuración de TSIG con BINDv9
- Configuración del maestro
- Configuración del esclavo
- Ventajas y desventajas

Transaction Signatures (TSIG)

- El “nombre formal” para TSIG (Transaction Signatures), de acuerdo al RFC 2845 que describe este protocolo es “autenticación de transacciones con llave secreta para el DNS” (Secret Key Transaction Authentication for DNS).
- El protocolo TSIG permite la autenticación a nivel transacción usando secretos compartidos y una función hash de una sola vía (one way hashing). TSIG puede ser usado para autenticar actualizaciones dinámicas (dynamic updates) que provengan de un cliente autorizado o para autenticar las respuestas que vienen de un servidor de nombres recursivo autorizado.
- TSIG hace uso de un código de autenticación de mensaje (Message Authentication Code, MAC), específicamente HMAC-MD5 (una función hash con llave o keyed hash function) para permitir un método eficiente de autenticación y verificación de integridad para las transacciones.
- En otras palabras, TSIG es un mecanismo de autenticación de transacciones que usa llaves secretos compartidas y funciones hash para establecer una relación de confianza entre dos entidades.

Transaction Signatures (TSIG)

- TSIG fue introducido en BINDv8.2 debido a la necesidad que había de un mecanismo de autenticación simple y eficiente entre los clientes y los servidores locales.
- Sin embargo, TSIG no es la solución adecuada para autenticar la comunicación entre varios servidores, ya que la administración de llaves se dificultaría cada vez más, pues el número de llaves compartidas se incrementaría de forma cuadrática por cada servidor adicional.

Transaction Signatures (TSIG)

- Antes de que TSIG pueda ser utilizado, es necesario que se configure una llave en cada entidad.
- Por ejemplo, si se desea asegurar las transferencias de zona entre un servidor de nombres maestro (primario) y un servidor de nombres esclavo (secundario), debe indicarse en ambos el uso de una llave común para firmar la comunicación entre ellos.
- Para el caso de BIND, esta llave debe ser definida en ambos servidores con la sentencia key en el archivo de configuración del programa named, que usualmente es /etc/named.conf:

```
key nsmaster-nsslave.example.com. {
    algorithm hmac-md5;
    secret "GnTuTd0/xHoh6GgxOR4iOw==";
};
```

- El argumento de la sentencia key (nsmaster-nsslave.example.com.) es el nombre de la llave. El RFC para TSIG sugiere usar los nombres de las máquinas que comparten la llave como el nombre de la misma. Es importante recordar que el nombre de la llave debe ser exactamente el mismo en ambos servidores de nombres.

Transaction Signatures (TSIG)

- La sintaxis para crear la llave en BINDv9 usando el comando dnssec-keygen es la siguiente:

  # dnssec-keygen –a HMAC-MD5 –b 512 –n HOST nsmaster-nsslave.example.com.

  Knsmaster-nsslave.example.com.+157+08660

  dónde:

  -a Nombre del algoritmo con que será usada la llave

  -b Longitud de la llave en bits

  -n Tipo de llave a generar

- El comando anterior crea un par de archivos que contienen las llaves generadas:

  Knsmaster-nsslave.example.com.+157+08660.key

  Knsmaster-nsslave.example.com.+157+08660.private

Transaction Signatures (TSIG)

- **Configuración de TSIG en un servidor de nombres maestro**

- Crear la llave compartida:

  # dnssec-keygen –a HMAC-MD5 –b 512 –n HOST nsmaster-nsslave.example.com.

- Agregar las siguientes líneas al archivo named.conf del servidor de nombres maestro:

```
options {
        directory "/var/named";
        pid-file "/var/run/named.pid";
…
        allow-transfer { key nsmaster-nsslave.example.com.; };
};

key nsmaster-nsslave.example.com. {
    algorithm hmac-md5;
    secret " rsa6MSmfDRUc+x6x5CdwgVfT4cDbFKoGM7fHoX1KfTmxuQqGHrN7g3/H BLwvmFk6dlPP0zyRIHmfxKizd4Zwqw== ";
};
```

- Reiniciar el demonio named en el servidor de nombres maestro

Transaction Signatures (TSIG)

- **Configuración de TSIG en un servidor de nombres esclavo**

- Transferir cualquiera de los archivos de llaves generados en el maestro (o hacer copy&paste de la llave) al servidor de nombres esclavo

- Agregar las siguientes líneas al archivo named.conf del servidor de nombres esclavo:

```
options {
        directory "/var/named";
        pid-file "/var/run/named.pid";
       …
        allow-transfer { none; };
};


key nsmaster-nsslave.example.com. {
algorithm hmac-md5;
secret " rsa6MSmfDRUc+x6x5CdwgVfT4cDbFKoGM7fHoX1
KfTmxuQqGHrN7g3/HBLwvmFk6dlPP0zyRIHmfxKizd4Zwqw==";
};
server 10.0.0.1 {
keys { nsmaster-nsslave.example.com.; };
};
```

- Reiniciar el demonio named en el servidor de nombres esclavo

Transaction Signatures (TSIG)

- Tareas posteriores a la configuración de TSIG

- Eliminar los archivos creados en el servidor de nombres maestro:

  # rm Knsmaster-nsslave.example.com.+157+08660.key

  # rm Knsmaster-nsslave.example.com.+157+08660.private

- Validar que las horas de los servidores de nombres estén sincronizadas (es despreciable una diferencia de hasta 3 minutos)

- Hacer pruebas de transferencia de zonas entre el servidor maestro y el esclavo usando TSIG

Transaction Signatures (TSIG)

- **Ventajas de TSIG**
  - Su configuración es fácil y rápida
  - Es un mecanismo ligero para servidores de nombres y resolvers
  - Es computacionalmente mucho menos caro que DNSSEC. Mientras la llave secreta compartida no se vea comprometida, se dispone de autenticación fuerte entre el servidor de nombres y el resolver del usuario
  - Es flexible para asegurar mensajes del DNS y actualizaciones dinámicas
  - Constituye una solución adecuada para esquemas simples de DNS

Transaction Signatures (TSIG)

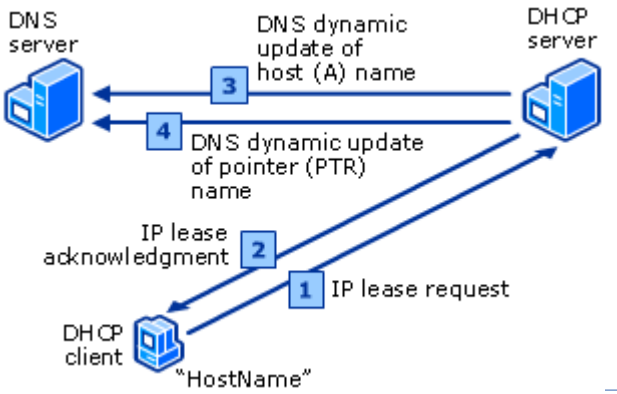
- **Desventajas de TSIG**
  - Como TSIG usa secretos compartidos, no es práctico configurarlo para muchos servidores de nombres, debido al problema de la distribución y administración de las llaves.
  - Como la mayoría de los protocolos de cifrado de llave secreta, TSIG no contempla la distribución de los secretos compartidos o llaves, así que esta tarea debe ser realizada por el administrador del DNS.
  - TSIG es adecuado para asegurar las transacciones entre dos servidores de nombres, pero no puede proteger un servidor de nombres cuya seguridad ha sido comprometida.
  - TSIG no autentica la fuente de los datos, únicamente su transmisión entre dos entidades que comparten un secreto.

DNS Dinámico

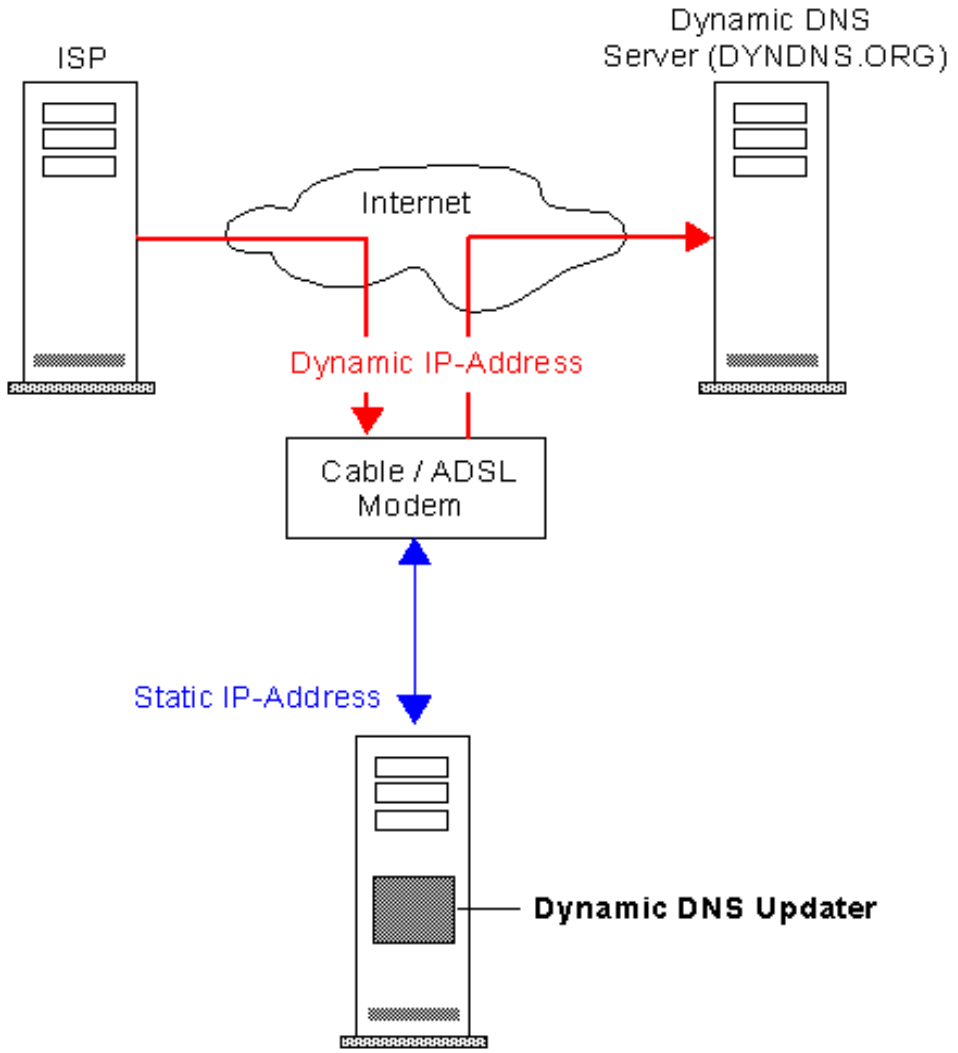
- DNS Dinámico
- Configuración de un DNS Dinámico
- Actualización dinámica de registros

DNS Dinámico

- El DNS dinámico (DDNS) es un servicio que permite la actualización en tiempo real de la información sobre nombres de dominio situada en un servidor de nombres. El uso más común que se le da es permitir la asignación de un nombre de dominio de Internet a un dispositivo con dirección IP variable (dinámica). Esto permite conectarse con la máquina en cuestión sin necesidad de tener conocimiento de que dirección IP posee en ese momento.
- El DNS dinámico hace posible utilizar un software de servidor en un dispositivo con dirección IP dinámica (como la suelen facilitar muchos ISP) para, por ejemplo, alojar un sitio web en la PC de nuestra casa, sin necesidad de contratar un hosting de terceros; pero hay que tener en cuenta que las PC caseras posiblemente no estén tan bien dotadas como los servidores de un Datacenter , ni tengan toda la infraestructura que poseen estos lugares.



DNS Dinámico



DynDNS

DNS Dinámico

Configuración de un DNS Dinámico

- Crear el archivo */etc/named.conf* con el siguiente contenido:

```
options {
        listen-on port 53 { any; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { any; };
};
```

DNS Dinámico

```
        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";

        recursion no;
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
```

DNS Dinámico

```
// use the default rndc key
include "/etc/rndc.key";

controls {
        inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "rndc-key"; };
};

include "/etc/named.rfc1912.zones";
include "example.com.key";

zone "example.com" IN {
        type master;
        file "dynamic/example.com.db";
        allow-update { key example.com ; } ;
};
```


DNS Dinámico

- El servicio named también está configurado para permitir a los sistemas que cuenten con una llave de seguridad de DNS puedan realizar actualizaciones de resolución. Esta llave debe ser incluida en el archivo de configuración.
- Generamos la llave en el directorio de trabajo de nuestro DNS:

  # cd /var/named

  # dnssec-keygen -a hmac-md5 -b 256 -n USER example.com

DNS Dinámico

- Este comando nos genera la llave con la cual, se debe configurar el archivo */var/named/example.com.key*

```
key example.com {
  algorithm HMAC-MD5;
  secret "qszIouTm5HejMWtaGgGrxEOyX9G2Vk+wpQCs+jtBwGM=";
};
```

DNS Dinámico

- Se debe crear el archivo de zona /var/named/dynamic/example.com.db

```
$ORIGIN .
$TTL 1 ; 1 second
example.com   IN SOA    ns.example.com. root.example.com. (
                                2014020601 ; serial
                                60         ; refresh (1 minute)
                                15         ; retry (15 seconds)
                                1800       ; expire (30 minutes)
                                 10        ; minimum (10 seconds)
                                 )
                               NS  ns.example.com.
                               MX  10 mail.example.com.
$ORIGIN example.com.
```

DNS Dinámico

- Con la configuración terminada se deben asignar los permisos correspondientes:

  # chown -v root.named /var/named/*

- Iniciamos el servicio

  # named -4 -d 2 -c /etc/named.conf.dyn

DNS Dinámico

Actualización dinámica de registros

- El comando **nsupdate** se utiliza para agregar registros en los archivos de zona de forma dinámica.
- Este comando requiere acceso al par de llaves, privada y pública, para ingresar las actualizaciones. La opción **-k** se utiliza para especificar la llave.

DNS Dinámico

- El comando **nsupdate** provee un intérprete de comandos para ejecutar diferentes acciones:
  - **server** Especifica la dirección IP del servidor de resolución de nombres
  - **update** Actualiza los registros en los archivos de zonas.

    Utiliza *add* ó *delete* como primer opción
  - **show** Muestra la actualización del registro ingresada, pero no la envía al servidor de resolución
  - **send** Envía la actualización de los registros ingresados
  - **quit** termina la sesión del intérprete de nsupdate

DNS Dinámico

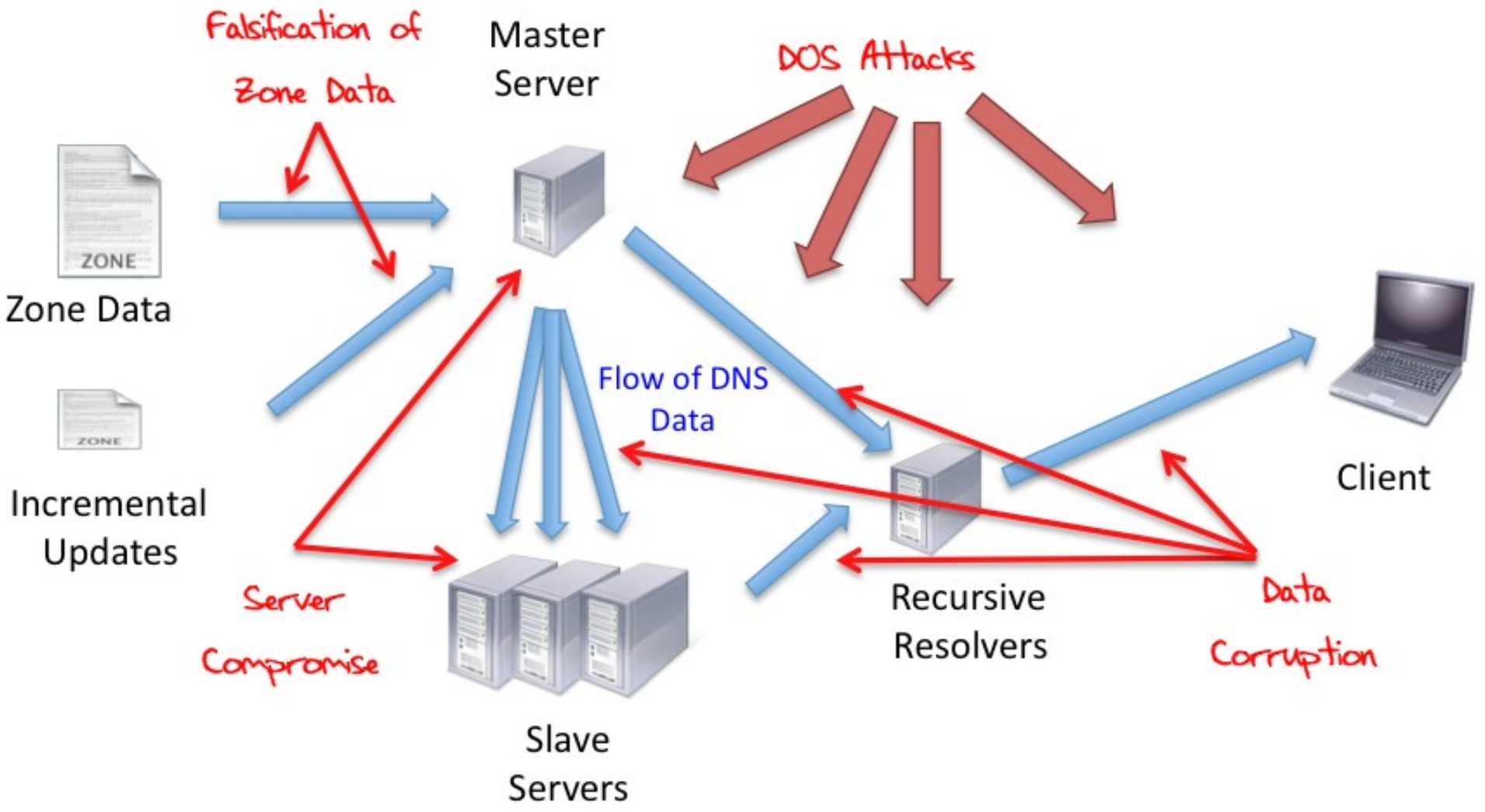
```
# nsupdate -k Kexample.com.+157+47071.key
> server 192.168.122.65
> update add mail.example.com 38400 A 192.168.0.3
> show
Outgoing update query:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
;; flags:; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
mail.example.com.  38400  IN A  192.168.0.3

> send
> quit
#
```

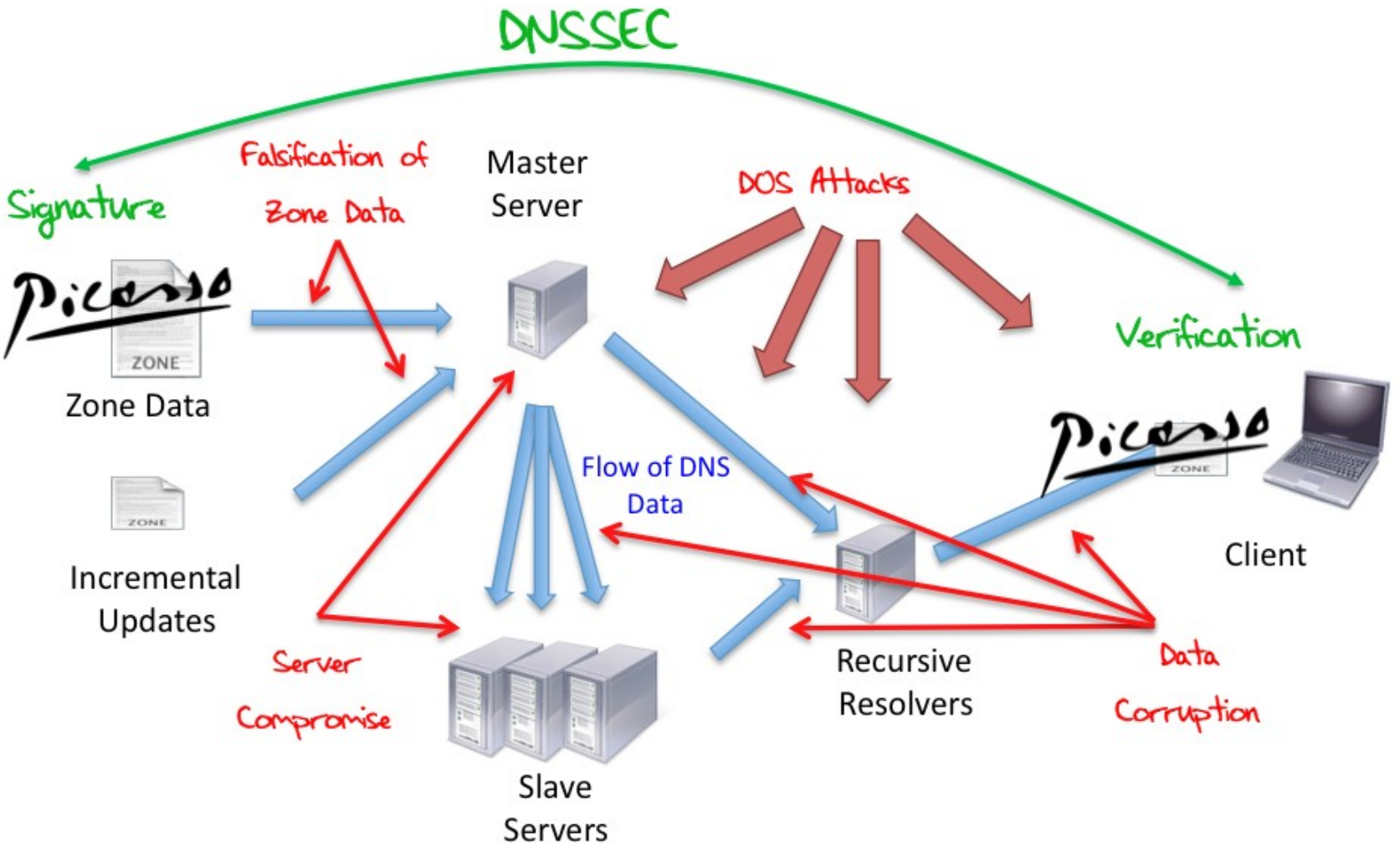
Agenda
Día 4

- DNSSEC
- IPA Server
- Troubleshooting

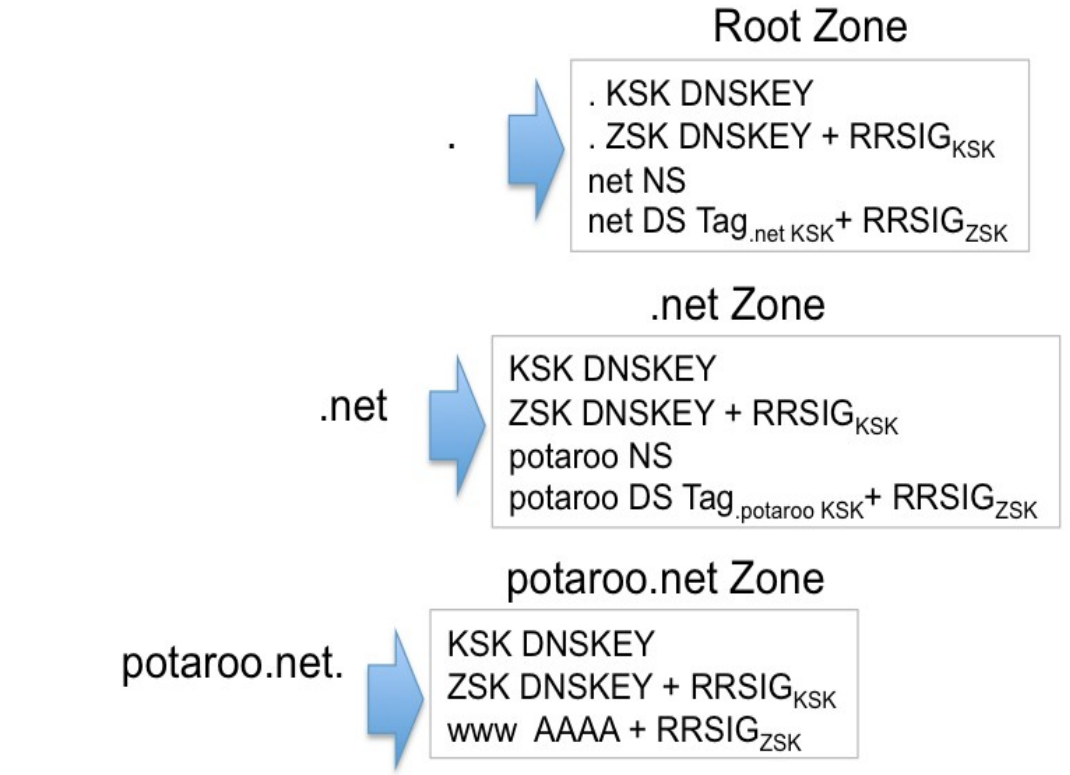
DNSSEC


Falsification of Zone Data
Master Server
DOS Attacks
ZONE
Zone Data
ZONE
Incremental Updates
Flow of DNS Data
Client
Server Compromise
Recursive Resolvers
Data Corruption
Slave Servers

DNSSEC

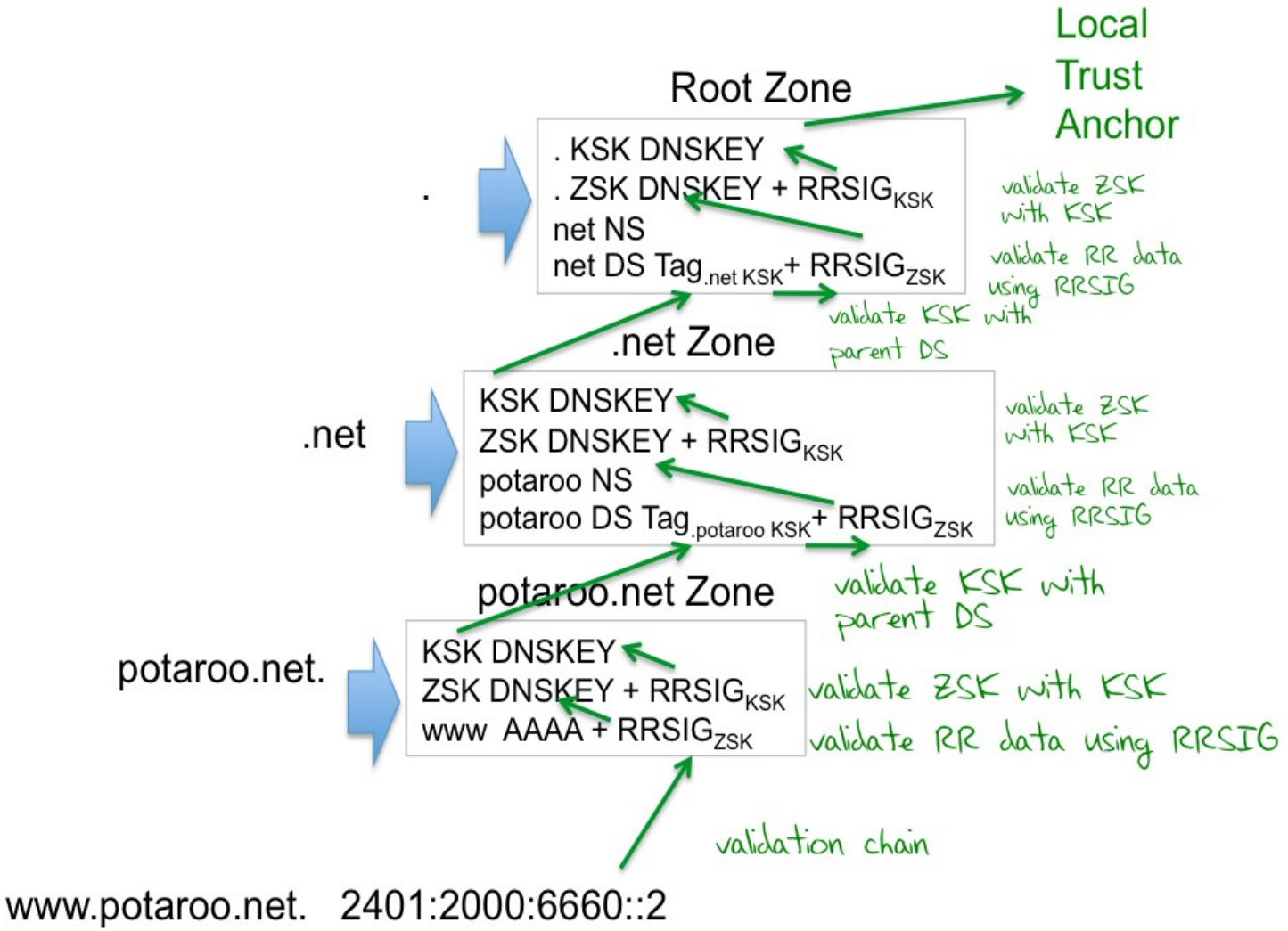

DNSSEC
Signature
Falsification of Zone Data
Master Server
DOS Attacks
Verification
Picasso
ZONE
Zone Data
Incremental Updates
Flow of DNS Data
Client
Server Compromise
Recursive Resolvers
Data Corruption
Slave Servers

DNSSEC



www.potaroo.net. 2401:2000:6660::2

DNSSEC


Local
Trust
Anchor
Root Zone
.
. KSK DNSKEY
. ZSK DNSKEY + RRSIG$_{KSK}$
net NS
net DS Tag$_{.net\ KSK}$+ RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validate KSK with parent DS
.net Zone
.net
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
potaroo NS
potaroo DS Tag$_{.potaroo\ KSK}$+ RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validate KSK with parent DS
potaroo.net Zone
potaroo.net.
KSK DNSKEY
ZSK DNSKEY + RRSIG$_{KSK}$
www AAAA + RRSIG$_{ZSK}$
validate ZSK with KSK
validate RR data using RRSIG
validation chain
www.potaroo.net. 2401:2000:6660::2

DNSSEC

Un concepto importante a entender es que BIND toma dos roles diferentes relacionados con DNSSEC. Uno de ellos es el de proporcionar datos firmados para una zona para la que está autorizado (donde es autoritario). El otro es el de la validación de un dispositivo de resolución para las zonas externas.

DNSSEC

Requisitos

- Última versión de BIND v9
- Librerías de SSL

Algo importante a tener en cuenta es que un servidor con DNSSEC que actúa como un dispositivo de resolución no validará los datos de una zona para la que está autorizado. Los datos vienen de su propio disco local, por lo que se consideran válidos. Esto significa que se necesitarán por lo menos dos instalaciones de BIND, un servidor autoritario para las zonas, y uno de resolución para confirmar la operación.

DNSSEC

- **Pre-requisitos**
  - Habilitar DNSSEC en el archivo de configuración named.conf

    dnssec-enable yes;
  - Reiniciar el servicio named

  - Generar las llaves para firmar las zonas

DNSSEC

- Antes de firmar una zona, se necesita tomar la decisión respecto al algoritmo y fortaleza de la llave que la firma va a utilizar.
- Los algoritmos soportados actualmente son:
  - RSA/MD5
  - RSA/SHA1*
  - DSA
- Las mejores prácticas en cuanto a la fortaleza son:
  - 1024 bits para la llave de firma de zona (ZSK)
  - 2048 bits para la llave de firma de llave (KSK)

DNSSEC

- Generar llave ZSK

  # dnssec-keygen -a RSASHA1 -b 1024 -n ZONE example.com

- Generar llave KSK

  # dnssec-keygen -a RSASHA1 -b 2048 -n ZONE -f KSK example.com

- Esto crea dos pares de archivos con el formato:

  Kexample.com+<alg>+<id>.key (the public key)

  Kexample.com+<alg>+<id>.private (the private key)

DNSSEC

- La llave pública de la ZSK y KSK necesitan ser añadidas al archivo de zona.

  Esto habilita validadores para recuperar las llaves.

  Utilizamos el comando:

  ```
  # cat Kexample.com+*.key >> example.com.zone
  ```

- Reiniciamos el proceso de BIND

DNSSEC

- Ahora existen dos nuevos registros en el fichero de zona, de tipo DNSKEY. Realizamos consultas para estos nuevos registros:

```
# dig @localhost example.com DNSKEY
```

DNSSEC

Firmar la zona

- Para firmar la zona, usamos el comando:

```
# dnssec-signzone -o example.com -N INCREMENT example.com
```

  - La opción **-N** actualiza el número de serie de SOA

- Editamos el archivo named.conf, modificando el nombre del archivo de zona

```
zone "example.com" in {
    file "example.com.zone.signed";
};
```

DNSSEC

Validación del resolver

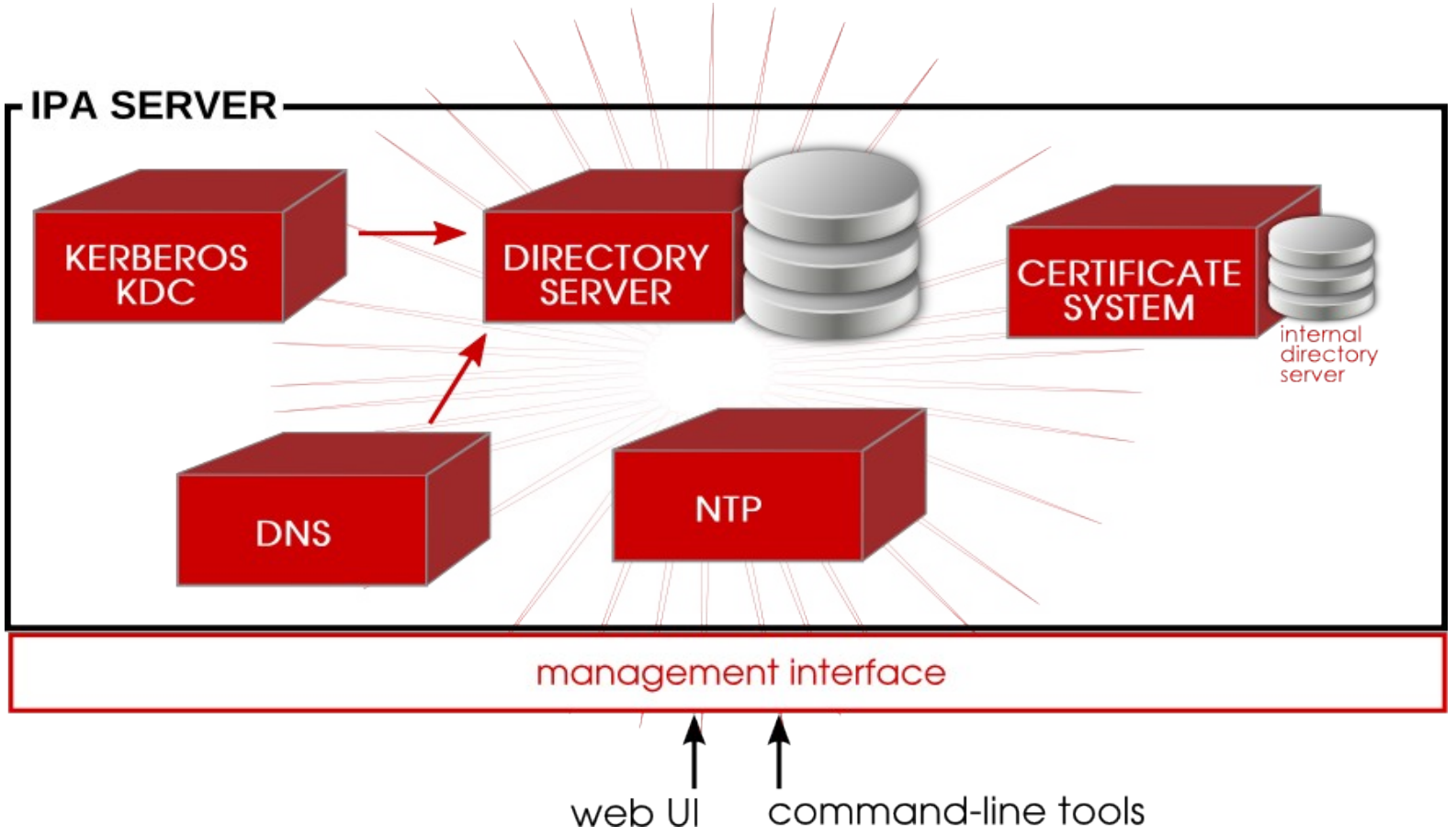
- La validación de DNSSEC se realiza agregando "+*dnssec*" a las consultas con dig
- Configuramos el archivo named.conf:

```
dnssec-enable yes;
dnssec-validation yes;
```

- Agregamos la llave de la zona:

```
trusted-keys {
    "example.com." 257 3 5 "AwEDSFASDF...";
};
```

IPA Server



identity | policy | audit

IPA Server

- Instalación

  [root@server ~]# yum install ipa-server bind bind-dyndb-ldap

  [root@server ~]# ipa-server-install

- Iniciar el servicio

  [root@server ~]# service sshd restart

  [root@server ~]# kinit admin

  Password for admin@EXAMPLE.COM:

IPA Server

- Instalar DNS

  [root@server ~]# ipa-dns-install -p [*password*] –ip-address=192.168.122.71 --no-forwarders

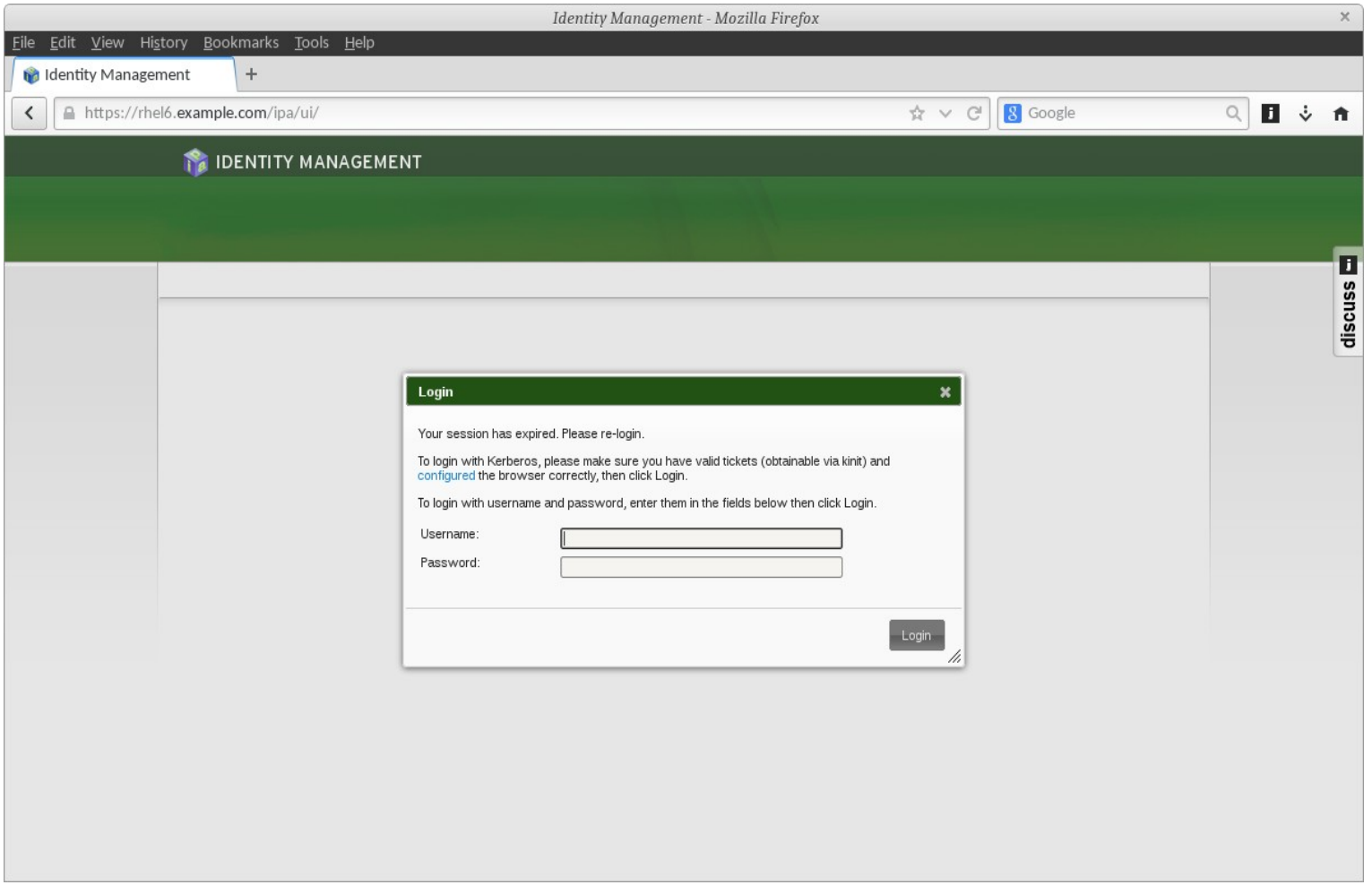
- Configurar rndc

  [root@server ~]# /usr/sbin/rndc-confgen -a

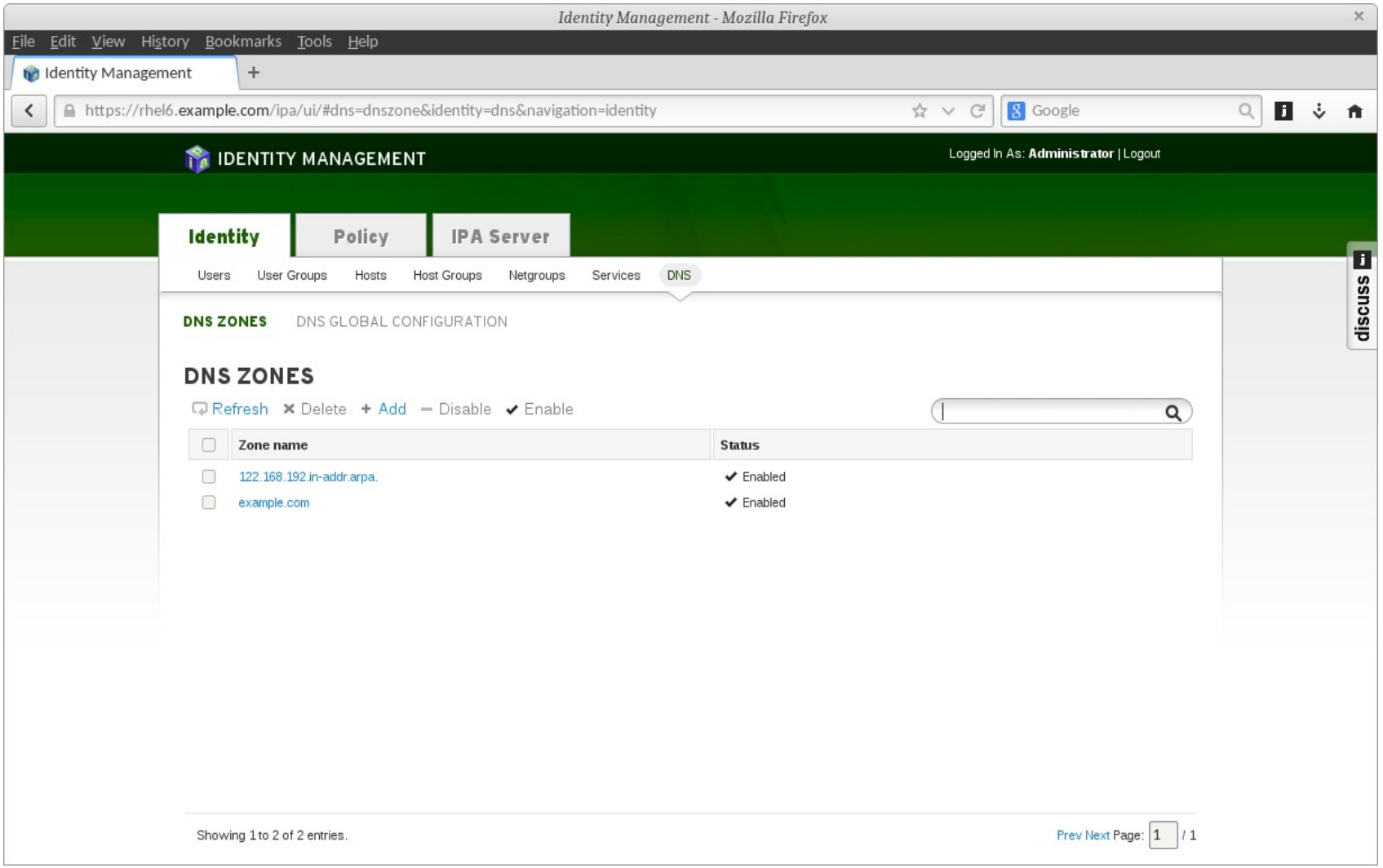
  [root@server ~]# chown root:named /etc/rndc.key

  [root@server ~]# chmod 0640 /etc/rndc.key

IPA Server


Identity Management - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Identity Management
https://rhel6.example.com/ipa/ui/
Google
IDENTITY MANAGEMENT
discuss
Login
Your session has expired. Please re-login.
To login with Kerberos, please make sure you have valid tickets (obtainable via kinit) and configured the browser correctly, then click Login.
To login with username and password, enter them in the fields below then click Login.
Username:
Password:
Login

IPA Server


Identity Management - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Identity Management
https://rhel6.example.com/ipa/ui/#dns=dnszone&identity=dns&navigation=identity
Google
IDENTITY MANAGEMENT
Logged In As: Administrator | Logout
Identity
Policy
IPA Server
Users
User Groups
Hosts
Host Groups
Netgroups
Services
DNS
discuss
DNS ZONES
DNS GLOBAL CONFIGURATION
DNS ZONES
Refresh
Delete
Add
Disable
Enable
Zone name
Status
122.168.192.in-addr.arpa.
Enabled
example.com
Enabled
Showing 1 to 2 of 2 entries.
Prev Next Page: 1 / 1

Troubleshooting

- Errores comunes
- Análisis de falla

Troubleshooting

- Errores comunes

1. *CNAME apuntando a registros NS*

```
domain.com.    IN    NS     ns1.domain.com.
domain.com.    IN    NS     ns2.domain.com.
domain.com.    IN    CNAME  ns9.example-server.net ---> INCORRECTO
```

2. *DNS con IP's en el mismo segmento ó en el mismo servidor físico*
3. *Pegamento adecuado (GLUE)*

   Los GLUE records son registros de tipo A asociados con los registros NS que proporcionan información "bootstrapping" al DNS

Troubleshooting

4. *Registros MX duplicados*

   domain.com. IN MX mail.domain.com.

   domain.com. IN MX mail.domain.com ----> DUPLICADO

5. *Puerto 53 bloqueado*

   Asegurarse que el firewall no bloquee el puerto 53 TCP/UDP ya que se utilizan por el DNS:

   Puerto 53 UDP= Peticiones DNS

   Puerto 53 TCP= Transferencia de Zonas

6. *Registros MX apuntando a registros CNAME*

   domain.com. IN MX 10 mail.domain.com.

   mail IN CNAME domain.com. ----------> INCORRECTO

Troubleshooting

7. *Los registros MX no deben contener IP's*

   domain.com. IN 10 MX mail.domain.com. ----> CORRECTO

   domain.com. IN 20 MX 11.22.33.44 ----> INCORRECTO

8. *Los registros NS no deben contener IP's*

   domain.com. IN NS dns0.domain.com. ----> CORRECTO

   domain.com. IN NS 75.86.13.20 ----> INCORRECTO

Troubleshooting

- Análisis de falla
  - Verificar ejecución de demonio named (nslookup, host, dig)
  - Revisar logs (/var/log/messages, named.log)
  - Revisar puertos (netstat, iptables, telnet)
  - Revisar archivos de configuración:

```
# named-checkconf /etc/named.conf
# named-checkzone example.com /var/named/example.com.zone
```

  - Verificar conectividad de red (ping, traceroute, tracepath)
  - Verificar los forwarders del DNS
  - Determinar el alcance de la falla
  - Verificar con herramientas de consulta desde Internet

Dudas?

Gracias.