Universidad Nacional Autónoma de Nicaragua

UNAN – León

Facultad de Ciencias Y Tecnología

Ingeniería en Telemática

Estudio de las vulnerabilidades del DNS y la implementación de las extensiones DNSSEC mediante casos prácticos en una red virtualizada.

Monografía para optar al título de

Ingeniero en Telemática

Autor(es):

Br. Denis Joel Medina Canales

Br. José Antonio Pérez Hernández

Br. José Benito Salgado Ruíz

Tutor:

M.Sc. Aldo René Martínez

León, noviembre 2019

Dedicatoria(s)

Eres una mujer que simplemente me hace llenar de orgullo, te amo y no hay manera de devolverte todo lo que me has ofrecido. Esta monografía es un logro más que llevo a cabo, y sin lugar a dudas ha sido en gran parte gracias a ti; no sé en donde me encontraría de no ser por tu ayuda, tu compañía y tu amor. Te doy mis sinceras gracias madre.

También le agradezco a quien ha moldeado mi camino y me ha dirigido por el sendero correcto, a Dios, el que en todo momento está conmigo ayudándome a aprender de mis errores y a no cometerlos otra vez, eres quien guía el destino de mi vida. Te lo agradezco padre celestial.

Denis Joel Medina Canales

A todos los seres que amo, en especial a mi madre.

José Antonio Pérez Hernández

A mis seres queridos.

José Benito Salgado Ruíz.

Agradecimientos

A todos los docentes por el esfuerzo y dedicación en compartir sus conocimientos y valores, permitiendo nuestro crecimiento profesional, y en especial a nuestro tutor M.Sc. Aldo Martínez, por su apoyo incondicional y su amistad.

Índice de contenidos

Índice de figuras

Índice de tablas

1 INTRODUCCIÓN

DNS (Sistema de Nombres de Dominio), proporciona un esquema jerárquico de nombres basado en dominios y una base de datos distribuida para implementar este esquema. Su función principal es la de relacionar direcciones de host y servicios de red con sus direcciones IP correspondientes y viceversa. Las extensiones de seguridad DNSSEC nacen de la necesidad provocada por las diferentes vulnerabilidades en los servidores DNS tales como el envenenamiento de caché, denegación de servicio, etc. Dichas extensiones permiten aumentar la seguridad de estos evitando ataques.

Existen diferentes herramientas para la implementación de DNSSEC las cuales se implementan según el tipo de infraestructura, para el escenario virtual se utiliza la herramienta Bind9 que se configura en Centos 7, así como también este mismo sistema operativo se utiliza como servidor DNS, estas herramientas son unas de las más utilizadas en los DNS.

El presente documento detalla el estudio de las vulnerabilidades del DNS y la implementación de las extensiones DNSSEC mediante casos prácticos en una red virtual, montando un ambiente de laboratorio con el fin de configurar y gestionar las extensiones de seguridad al DNS. Una vez comprobado el funcionamiento de los DNS en la red privada, se implementan las extensiones DNSSEC en los servidores configurados del escenario virtualizado.

Este trabajo monográfico contiene el desarrollo de los contenidos teóricos que son necesarios para el entendimiento o profundización del tema, de igual manera el desarrollo y procedimientos de las configuraciones necesarias para la implementación de las DNSSEC, en un entorno de red virtual.

1.1 Antecedentes

Desde la creación del sistema DNS, este ha carecido de un diseño que asegure la información intercambiada entre Clientes y Servidores, actualmente la solución que se presenta son las Extensiones de Seguridad (DNSSEC) para DNS, que mediante una jerarquía de firmas criptográficas provee autenticación del origen y seguridad de los datos.

Durante la década de 1980, se conectaron redes adicionales, en particular LANs, (Tanenbaum, 2003) afirma. "ARPANET (Advanced Research Projects Agency Network) es la abuela de todas las redes de computadoras de área amplia" (p.41), fue una red de computadoras creada por encargo del Departamento de Defensa de los Estados Unidos (DOD) para utilizarla como medio de comunicación entre las diferentes instituciones académicas y estatales. Conforme crecía el escalamiento, encontrar hosts llegó a ser muy costoso, por lo que se creó el **DNS** (**Sistema de Nombres de Dominio**). DNS es un pilar fundamental en el Internet, ya que sin un sistema que realice el intercambio de un nombre de dominio a una dirección IP, sería casi imposible recordar cada una de las direcciones IPs, de la red de redes Internet.

En Nicaragua, según las autoridades DAFI (Departamento de Análisis Forenses Cibernéticos), sólo las instituciones financieras privadas se ocupan activamente de fomentar la concientización sobre seguridad cibernética lo que conlleva a que las empresas e instituciones u otros que empleen el uso de TIC no tomen en cuenta la importancia de la implementación de las extensiones DNSSEC (Symantec, 2014, p.70).

Actualmente en Nicaragua el gobierno no ha desarrollado ninguna campaña nacional de concientización sobre seguridad cibernética. El Comité de Ciencia y Tecnología (CONICYT) de Nicaragua, trabaja para desarrollar una política nacional y aumentar la conciencia de seguridad cibernética (IDB, 2016, p.90).

En otros países se han realizado distintos estudios acerca de la implementación y configuración de DNSSEC. A continuación, se presentan algunos de estos trabajos:

- **"La seguridad de DNS: La investigación de la DNSSEC mediante la experimentación práctica",** Da Silva, 2009. Esta monografía aborda la comprensión sobre temas de seguridad, claves en el DNS, las mejoras propuestas por DNSSEC en los servidores DNS, evalúa el uso del experimento, la facilidad y la seguridad de esta propuesta en un escenario práctico.

- **"Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNSSEC)",** Sánchez, 2012. El documento monográfico explica el estudio del arte del Sistema de Nombre de Dominios y las Extensiones de Seguridad para DNS, también la presentación de fallas de seguridad del protocolo DNS y métodos de "ataques" comunes al Sistema DNS y por último la relevancia conforme a que las extensiones no deben considerarse como una solución integral, ya que, en el contexto de los estándares de DNS, solo se aseguran la autenticación y la integridad de los datos.

- **"Despliegue de DNSSEC, Validación en servidores de nombres de almacenamiento en caché recursiva"**, Rijswijk- Deij, 2012. En este artículo se muestra el despliegue de DNSSEC y sus dos caras: las cuales son la implementación en el cliente de DNS, también a lo que se conoce como validación DNSSEC y la puesta en marcha, así como también el lado del servidor, el cual lleva por nombre DNSSEC de firma; también describe las decisiones a tomar y las implicaciones al habilitar el soporte de DNSSEC en una organización, conteniendo listas de comprobación necesarias para el proceso de decisión y la descripción de costos y beneficios de la implementación DNSSEC.

- **"Extensiones de seguridad del DNS (DNSsec)"**, Barreira, Grassi y Silvera, 2012. Este proyecto documentado, aborda la implementación de seguridad adicional a los sistemas de nombres de dominios por medio de las DNSSEC, y los conceptos fundamentales de los DNS y las extensiones de seguridad para una mayor comprensión sobre sus funcionamientos.

- **"Propuesta de implementación de las extensiones de seguridad DNSSEC en los servidores DNS internos de la Universidad Técnica Particular de Loja",** Guanolique Pereira, 2016. El documento abarca el desarrollo de la implementación de las extensiones DNSSEC en los servidores DNS internos de la Universidad Técnica Particular de Loja (UTPL). Además, se presenta la metodología que contiene la situación actual, así como también la infraestructura de la UTPL y los resultados en prueba de la implementación gracias a las configuraciones en el ambiente de laboratorio empleado.

- **“Análisis de seguridad del sistema DNS (Domain Name System)”,** Armas, 2017. Esta tesis trata del análisis de seguridad del sistema DNS y las diferentes formas de hacer frente a esta problemática, mediante el análisis de resultados, el cual ayudará a proponer mecanismos de seguridad.

1.2 Planteamiento del Problema

DNS es el sistema base fundamental del funcionamiento de internet, este protocolo se encarga de traducir un nombre de dominio a una dirección IP, la petición o mensaje DNS carece de un sistema de seguridad por lo que la vulnerabilidad del mensaje el cual es un texto plano; conforma un entorno propicio para una multitud de ataques.

Un servidor DNS tradicional no incluye métodos de seguridad, muchas empresas no implementan la seguridad en el DNS, debido a que el Proveedor de Servicios de Internet (ISP), no ofrece dicho servicio. Los servidores DNS de la red privada no poseen las extensiones de seguridad DNSSEC, lo cual provoca que los servicios que proveen los servidores DNS sean vulnerables ante cualquier ataque generado por terceros, lo que afecta a todos los usuarios que dependen del servidor DNS vulnerable, es por esta razón que decidió efectuar la implementación de extensiones de seguridad, haciendo uso de softwares actuales necesarios que ayuden a la seguridad de nuestra red privada.

Pregunta general

¿Es factible implementar las extensiones DNSSEC en los servidores DNS de una red privada para asegurar la comunicación y fomentar una cultura de seguridad en la capa de aplicación?

Preguntas específicas

- ¿Cuáles son los fundamentos de las extensiones DNSSEC y sus principales componentes?
- ¿Cómo crear una infraestructura de red local virtualizada con los servicios DNS y Web, y con protocolos de enrutamiento dinámicos?
- ¿Cómo ejecutar ataques de DNS Spoofing y Denegación de servicio al DNS en la red virtualizada, mediante casos prácticos en GNS3?
- ¿Cuáles son los procedimientos para implementar DNSSEC en una red privada?

1.3 Justificación

Teniendo como referencia los problemas expuestos anteriormente, se decidió crear este documento, en el cual se plasman contenidos teóricos y se describe el desarrollo de la implementación de las extensiones DNSSEC en los servidores DNS en una red virtualizada. Esto es de gran ayuda para la persona, grupos, entidades y otros que estén interesados y tengan la necesidad de adquirir conocimientos que les permitan obtener un nivel de seguridad adicional en el DNS (Sistema de Nombres de Dominio).

1.3.1 Originalidad

En internet existen trabajos y documentos similares a este tema y el despliegue de las DNSSEC en distintos entornos prácticos, no obstante, el presente trabajo es efectuado en un entorno virtual y así mismo aporta a toda la comunidad informática una guía completa de la implementación de las extensiones de seguridad al DNS, por lo que este documento ayuda de manera relevante a mejorar la seguridad de dicho servidor.

1.3.2 Alcance

Una vez realizadas las configuraciones y la implementación de las DNSSEC en los servidores DNS de la red virtualizada, se obtiene lo siguiente:

- Autenticación y validación de origen.
- Seguridad de los datos mediante mensajes cifrados entre cliente y servidor.

1.3.3 Producto

Este documento presenta las siguientes características:

- **Sencillo:** Porque el desarrollo y procedimientos están elaborados de forma que se comprenda fácilmente.
- **Guiado:** Se describe de forma detallada los pasos necesarios para la implementación de las DNSSEC en los servidores DNS.
- **Completo:** Porque dispone de las configuraciones requeridas para la implementación de dichas extensiones y los resultados puestos en prueba.

1.3.4 Impacto

Este trabajo es de utilidad para que las entidades o personas relacionadas al área de informática cuenten con un material de apoyo eficaz. Con esto, se está aportando un documento que sirva como referencia necesaria para enfrentar los retos que sufre un servidor DNS y sobre todo tomar en cuenta la importancia en la seguridad de este.

1.4 Objetivos

1.4.1 Objetivo General

- Estudiar las vulnerabilidades del DNS e implementar las extensiones de seguridad DNSSEC mediante casos prácticos en una red virtualizada.

1.4.2 Objetivos Específicos

- Exponer los fundamentos de DNSSEC y sus principales componentes.
- Crear una infraestructura de red local virtualizada con los servicios DNS y Web, y con protocolos de enrutamiento dinámicos.
- Ejecutar ataques de DNS Spoofing y Denegación de servicio al DNS configurados en la red virtualizada, mediante casos prácticos desplegados en GNS3.
- Detallar los procedimientos necesarios para implementar DNSSEC en la red virtualizada previamente configurada.

2 MARCO TEÓRICO

2.1 Historia del DNS

Tanenbaum (2003) menciona que en la década de los 60's surge el crecimiento de ARPANET (Advanced Research Projects Agency Network) en algunas Universidades de Estados Unidos, por lo que en los años 70's no se contaba aún con el protocolo DNS sólo había un archivo, hosts.txt, en el que se listaban todos los hosts y sus direcciones IP. Debido a que la red era conformada por unas cuantas máquinas grandes de tiempo compartido en su momento este método no presentó inconvenientes. No obstante, cuando muchos más hosts se conectaron a esta red el archivo creció de una manera considerable volviéndolo obsoleto, para resolver tal problema se inventó el DNS (Sistema de Nombres de Dominio).

Guanolique, C (2016) refiere que el DNS fue desarrollado en noviembre de 1983 por Paul Mockapetris (RFC 882 y RFC 883) y luego revisado en 1987 definiéndose en las RFC 1034 y 1035. El DNS tiene un papel importante en la evolución del Internet, sin él no funcionaría lo que conocemos ahora como la Word Wide Web.

2.2 ¿Qué es el DNS?

Es un sistema que permite la traducción de nombres de dominio a direcciones IP, esta información es almacenada en una base datos distribuida de forma jerárquica, trabaja con el formato cliente (resolver) – servidor (Servidor DNS), facilitando a los usuarios poder navegar a través de internet.

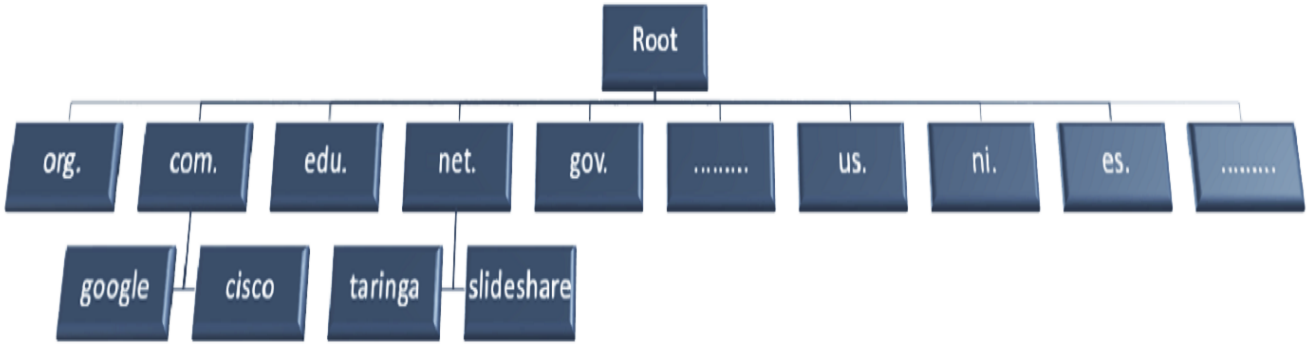


Ilustración 1: Árbol parcial inverso de la estructura del DNS

En la figura 1 se puede apreciar que el sistema de nombres de domino tiene una estructura de árbol inversa la cual es similar a la estructura de directorios de un sistema operativo Windows o Linux. Además, se puede ver que la raíz (Root) no tiene etiqueta, mientras que las etiquetas de los demás espacios de nombres (hojas) corresponden a un dominio, en el cual puede haber más subdominios.

2.3 Nombres de Dominio del DNS

Los nombres de dominio son etiquetas separadas con un punto ".", cada dominio se divide en subdominios, los cuales, a su vez, también se dividen, y así sucesivamente. Los dominios de nivel superior se dividen en dos categorías: genéricos y de país. Los dominios genéricos originales son com (comercial), edu (instituciones educativas), gov (el gobierno federal de Estados Unidos), int (ciertas organizaciones internacionales), mil (las fuerzas armadas de Estados Unidos), net (proveedores de red) y org (organizaciones no lucrativas). Los dominios de país incluyen una entrada para cada país, como se define en la ISO 3166. (Tanenbaum, 2003).

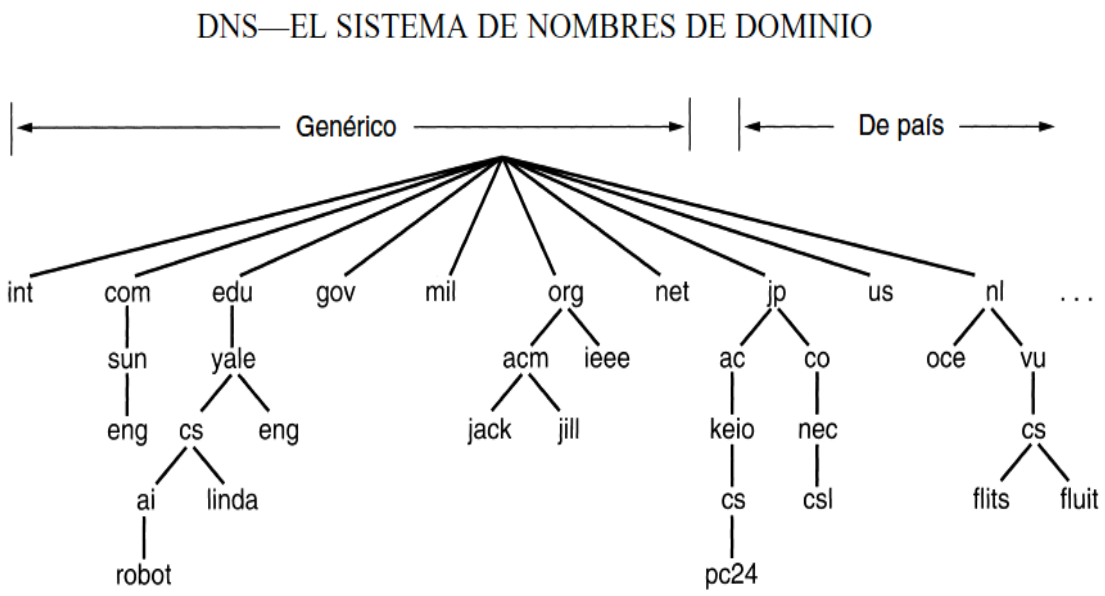


Ilustración 2: Parte del espacio de nombres de dominio de Internet
Fuente: (Tanenbaum, 2003)

2.4 Funcionamiento del DNS

Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado resolver, y le pasa el nombre como parámetro. El resolver envía un paquete UDP a un servidor DNS local, que después busca el nombre y devuelve la dirección IP al resolver, que entonces lo devuelve al solicitante. Una vez que tiene la dirección IP, el programa puede establecer una conexión TCP con el destino, o enviarle paquetes UDP. (Tanenbaum, 2003).

2.4.1 Protocolo DNS

El protocolo DNS consta de dos partes principales: un protocolo de pregunta/respuesta utilizado para realizar consultas para nombres de dominios, y otro protocolo para el intercambio de registros de bases de datos (transferencias de zona) y notificaciones a los Servidores esclavos, como consecuencia de un cambio en la zona principal (DNS Notify) y en las actualizaciones dinámicas de la zona (Dynamic updates). (Sánchez, 2012).

2.4.2 Consultas DNS

Las consultas recibidas por el servidor de nombres de dominio pueden ser de dos tipos, recursivas e iterativas.

Consulta Recursiva: Es cuando se realiza una petición de resolución de nombres al servidor DNS local. Si el servidor no dispone de dicha información reenvía la petición al servidor de nombres con autoridad que la contiene. De forma recursiva se buscará la información y será devuelta al cliente.

(1) – El host pregunta por www.unanleon.edu.ni al servidor DNS local de su zona (dominio).
(2) - El servidor DNS de la zona pregunta al DNS server con dominio .ni
(3) - El servidor DNS .ni pregunta al servidor DNS con dominio .edu.ni
(4) - El servidor DNS .edu.ni le devuelve la @IP del servidor unanleon.edu.ni al dominio .ni
(5) (6) - Se devuelve la @IP del servidor unanleon.edu.ni al cliente.

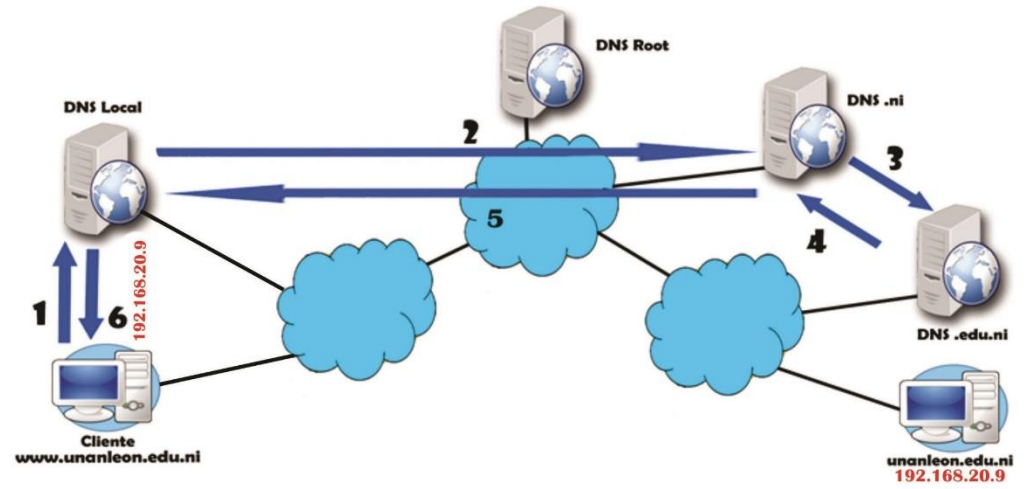


Ilustración 3: Consulta Recursiva

Consulta Iterativa: El servidor DNS local responde al cliente en función del contenido de su caché. Si no dispone de la información solicitada, entonces realiza una resolución iterativa: consulta iterativamente los servidores de los dominios hasta resolver la dirección buscada, comenzando siempre por un servidor raíz.

(1) - El host cliente pregunta por unanleon.edu.ni a su servidor DNS.
(2) - El servidor DNS local pregunta a su DNS Root.
(3) - El root server DNS le devuelve la @IP del servidor DNS con dominio .ni
(4) - El DNS local pregunta al DNS .ni.
(5) - DNS .ni devuelve la @IP de DNS .edu.ni
(6) - DNS local pregunta a DNS .edu.ni
(7) - DNS .edu.ni devuelve la @IP del servidor unanleon.edu.ni
(8) - DNS local devuelve la @IP del servidor (unanleon.edu.ni) al cliente.

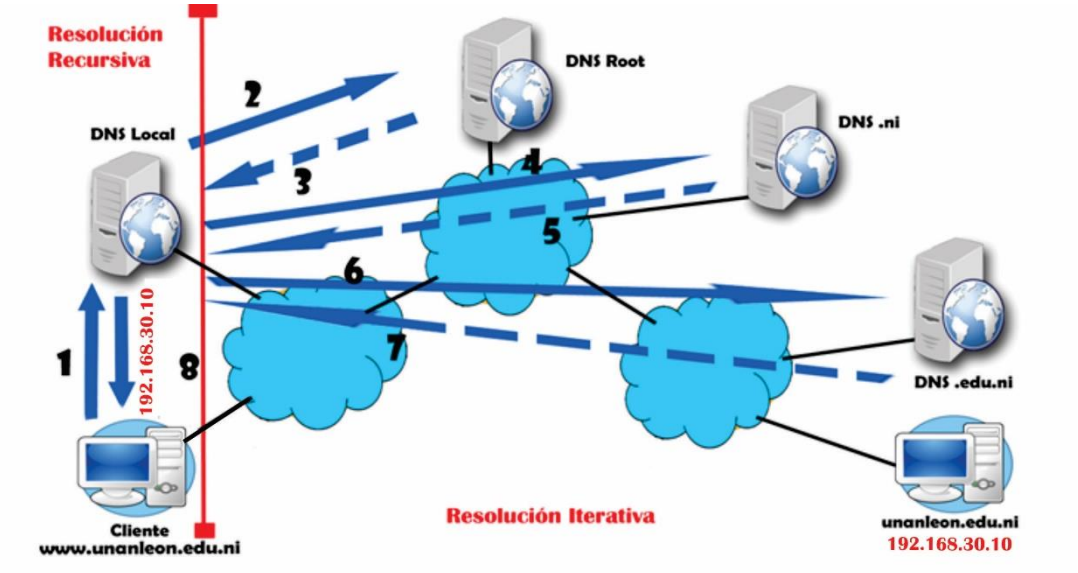


Ilustración 4: Consulta Iterativa

2.4.3 Servidor de nombres

Es el que guarda la información acerca del espacio de dominio de nombres en unos ficheros llamados "archivos de zona", por lo que responde las consultas proporcionadas por los clientes. El espacio de nombres DNS se divide en zonas no traslapantes.

Tanenbaum (2003) afirma que cada zona contiene una parte del árbol y esta misma contiene servidores de nombres que poseen la información de autorización correspondiente a esa zona.

2.4.4 Servidor DNS raíz

Un servidor raíz, o root server en inglés, es un servidor DNS que tiene conocimientos acerca de donde se encuentran los servidores de nombres autoritarios para cada una de las zonas de más alto nivel en Internet. Una consulta proveniente de cualquier dominio, el servidor raíz proporciona al menos el nombre y la dirección del servidor autorizado de la zona de más alto nivel para el dominio solicitado. De manera que el servidor del dominio proporcionará una lista de los servidores autorizados para la zona de segundo nivel, hasta obtener una respuesta razonable.

En toda internet existen 13 servidores raíz, cuyos nombres se escriben "letra.root-servers.org", aunque siete de ellos no son realmente servidores únicos, sino que representan múltiples servidores

distribuidos a lo largo del mundo. Estos servidores reciben miles de consultas por segundo, y a pesar de dicha carga de consultas, la resolución de nombres trabaja con bastante eficiencia. Los 13 servidores de nombres raíz son operados por 12 organizaciones independientes.

Tabla 1: Lista de servidores raíz en el mundo

Letra del DNS root server	Dirección IPv4	Dirección IPv6	Gestor
A	198.41.0.4	2001:503:ba3e::2:30	VeriSign
B	192.228.79.201	2001:478:65::53	USC-ISI
C	192.33.4.12	2001:500:2::c	Cogent Communications
D	199.7.91.13	2001:500:2d::d	University of Maryland
E	192.203.230.10		NASA
F	192.5.5.241	2001:500:2f::f	ISC
G	192.112.36.4		U.S. DoD NIC
H	128.63.2.53	2001:500:1::803f:235	US Army Research Lab
I	192.36.148.17	2001:7FE::53	Autonomica
J	192.58.128.30	2001:503:c27::2:30	VeriSign
K	193.0.14.129	2001:7fd::1	RIPE NCC
L	199.7.83.42	2001:500:3::42	ICANN
M	202.12.27.33	2001:dc3::35	WIDE Project

Fuente: (root-servers.org, s.f.)

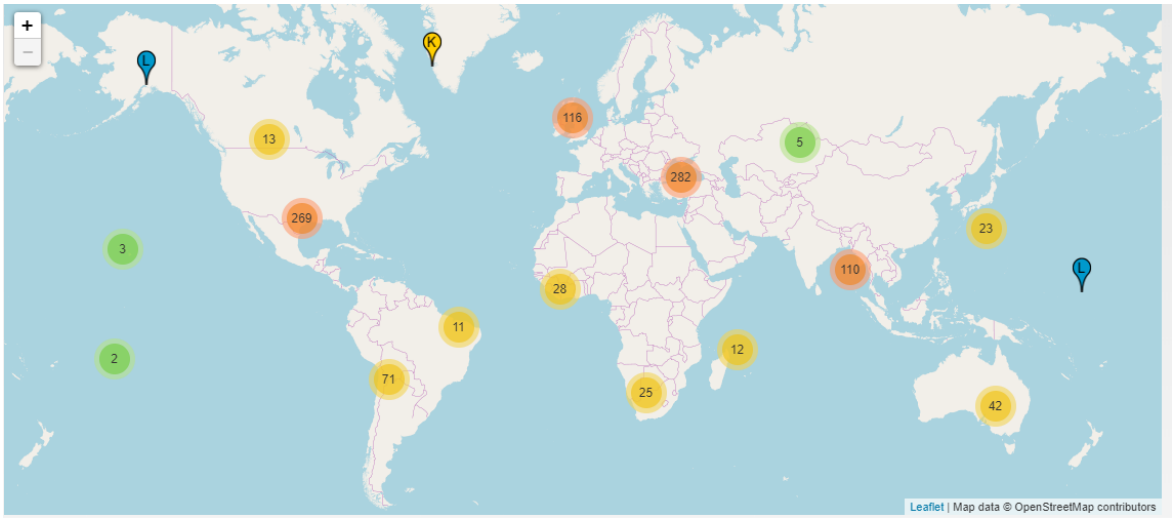


Ilustración 5 - Servidores raíz en el mundo

Fuente: (root-servers.org, s.f.)

2.4.5 Tipos de Servidores DNS

Tabla 2: Tipos de Servidores DNS

Tipo	Descripción
Primarios	Almacenan la información de su zona en una base de datos local. Son los responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.
Secundarios	Obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina transferencia de zona.
Maestros	Son los que transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca, busca un servidor maestro y realiza la transferencia de zona. Un servidor maestro para una zona puede ser a la vez un servidor primario

	o secundario de esa zona. Estos servidores extraen la información desde el servidor primario de la zona. Así se evita que los servidores secundarios sobrecarguen al servidor primario con transferencias de zonas.
Locales o solo caché	Estos no tienen autoridad sobre ningún dominio: se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apunta la respuesta en su memoria caché y le comunica la respuesta al cliente.

Fuente: (Comer, 1996)

2.4.6 Zona de autoridad

La información de un espacio de nombres de dominio se guarda en un Servidor de Nombres. Esto es la implementación física del llamado software DNS. Generalmente, los servidores tienen información completa acerca de una parte del espacio de nombres de dominio, que llamamos zona de autoridad. Más exactamente una zona es la porción del espacio de nombres de dominio de la que es responsable un determinado servidor DNS. La zona de autoridad de estos servidores abarca al menos un dominio y también pueden incluir subdominios, aunque a veces los servidores de un dominio pueden delegar sus dominios en otros servidores. (Sánchez, 2012).

La diferencia entre una zona y un dominio es que la primera contiene los nombres de dominio y datos que representan a un dominio y un dominio es un nombre que agrupa a otras máquinas o dominios inferiores.

2.4.7 Mensaje DNS

El mensaje DNS se divide en 5 secciones: cabecera (HEADER), pregunta (QUESTION), respuesta (ANSWER), autoridad (AUTHORITY) y adicional (ADITIONAL). Las tres últimas secciones mencionadas se encuentran en la categoría de Registros de Recursos (RR) por lo que comparten el mismo formato. (Armas, 2017).

Los mensajes DNS son los paquetes de datos intercambiados entre Servidores de Nombres (Maestro – Esclavo) o Servidores y Resolvers (Cliente). Comprender como se encuentran conformados es muy importante ya que corresponden a la base del protocolo de comunicación en DNS (preguntas y respuestas). Existe un formato único, el cual es usado para todas las operaciones DNS (consultas, respuestas, transferencia de zonas, notificaciones y actualizaciones dinámicas). (Sánchez, 2012).

Por lo general para el intercambio de preguntas y respuestas en DNS, se utiliza a UDP como protocolo de transporte, sin embargo, si éstas sobrepasan el límite de 512 bytes, se reemplaza el uso de UDP por TCP, dado el eventual truncado de mensajes que se produce al utilizar UDP con una cantidad de datos superior a este número.

2.4.8 Formato del mensaje

Como se había mencionado anteriormente cada mensaje posee el mismo formato genérico en donde se distinguen cinco secciones, las cuales son:

Sección	Significado/Uso
Sección 1	*HEADER:* Encabezado del mensaje
Sección 2	*QUESTION SECTION:* La consulta DNS para el que se solicita una respuesta.
Sección 3	*ANSWER SECTION:* Contiene el o los Registros de Recursos correspondientes a la consulta.
Sección 4	*AUTHORITY SECTION:* Detalla de forma explícita los Registros de Recursos de los Servidores de Nombres (Registros NS) referentes a la petición solicitada.
Sección 5	*ADDITIONAL SECTION:* Contiene la información que permite completar el resto de las secciones. Por ejemplo en el caso de existir varios servidores de nombres se explicitan sus direcciones IP para que puedan ser contactados

Ilustración 6: Formato de mensaje DNS.

Fuente: (Sanchéz, 2012)

Sección "HEADER"

Esta sección indica si el mensaje se trata de una consulta o una respuesta DNS, si es una consulta estándar u otro tipo de operación; además el número de entradas que tendrán las secciones respuesta, autoridad y adicional.

Tabla 3: Campos del encabezado DNS

Campo	Descripción
ID	Identificador de mensaje por la petición que crea el cliente cuando este genera una consulta el cual es usado luego por el mensaje de respuesta. El cliente no acepta la respuesta si el ID de consulta y respuesta es diferente.
QDCOUNT	Especifica el número de entradas en la sección "QUESTION". Normalmente se especifica el valor 1.
ANCOUNT	Especifica el número de Registros de recursos en la sección "ANSWER".
NSCOUNT	Especifica el número de Registros de recursos en la sección "AUTHORITY".
ARCOUNT	Especifica el número de Registros de recursos en la sección "ADITIONAL".

QR	Campo de un bit que especifica si el mensaje es una consulta (0), o una respuesta (1).
OPCODE	Específica el tipo de consulta.
AA	Flag de respuesta autoritativa. Solo válido en respuestas, si está activo, especifica que el servidor de nombres que responde tiene autoridad para el nombre de dominio enviado en la consulta.
RD	Flag de recursividad, indica al servidor de nombres que se pide resolución recursiva.

Sección "QUESTION"

En esta sección se especifican los parámetros los cuales definen lo que se solicita; es decir lleva la información sobre la consulta al servidor DNS.

Tabla 4: Campos de la Sección "QUESTION"

Campo	Descripción
QNAME	Especifica el nombre de dominio o la IP que está siendo consultada.
QTYPE	Especifica el tipo de Registro de recurso que está siendo solicitada.
QCLASS	La clase de Registro de recurso que está siendo solicitada.

2.4.9 Registros de Recursos

Son las porciones de información asociada a un nombre de dominio, esta información está almacenada en los servidores DNS autoritativos para cada dominio. (Armas, 2017).

Cuando un resolver da un nombre de dominio al DNS, lo que recibe son los registros de recursos asociados a ese nombre. Por lo tanto, la función real del DNS es relacionar los dominios de nombres con los registros de recursos. (Tanenbaum, 2003).

Un registro de recursos tiene cinco tuplas:

- Nombre_dominio: Indica el dominio al que pertenece este registro. Por lo general, existen muchos registros por dominio y cada copia de la base de datos contiene información de muchos dominios. Este campo es la clave primaria de búsqueda usada para atender las consultas.
- Tiempo_de_vida (TTL): Es una indicación de la estabilidad del registro. La información altamente estable recibe un valor grande, como 86,400 (la cantidad de segundos en un día). La información altamente volátil recibe un valor pequeño, como 60 (1 minuto).
- Clase: Este campo es el tercero en cada registro de recursos el cual define la clase a la que pertenece este mismo. Para la información de Internet, siempre es IN. Para información que no es de Internet, se pueden utilizar otros códigos.
- Tipo: Indica el tipo de registro y los más utilizados son:

Tabla 5: Principales tipos de Registros de Recursos DNS

Tipo de Registro	Descripción	Valor
SOA (Start of Authority)	Proporciona el nombre de la fuente primaria de información sobre la zona del servidor de nombres.	Parámetros para esta zona.
A (Address)	Dirección IP de un host, algunos hosts tienen dos o más conexiones de red, en cuyo caso tendrán un registro de recursos tipo *A* por cada conexión de red. DNS se puede configurar para iterar a través de éstos, regresando el primer registro en la primera solicitud, el segundo registro en la segunda solicitud, y así sucesivamente.	Entero de 32 bits.

MX	Se trata de un intercambiador de correo electrónico.	Prioridad, dominio dispuesto a aceptar correo electrónico.
NS	Especifica servidores de nombres para cada dominio.	Nombre de un servidor para un dominio.
CNAME	Permite la creación de alias, que permite direccionar la petición a nuestro dominio.	Nombre de Dominio.
PTR	Puntero que se usa para asociar un nombre de dominio a una dirección IP con el fin de realizar una búsqueda inversa; es decir que con la correspondiente dirección IP, se puede obtener la dirección web del equipo.	Alias de una dirección IP.
HINFO	Información del Host, tipo, modelo y sistema operativo de este.	CPU y SO en ASCCI.
TXT	Permite a los dominios identificarse de forma arbitraria.	Texto ASCCI no interpretado.

- Valor: Este campo puede ser numérico, una cadena de caracteres o un nombre de dominio.

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     www.djj.com. root.www.djj.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      www.djj.com.
@       IN      A       12.0.0.10
www     IN      A       15.0.0.10
emp1    IN      A       15.0.0.10
```

Ilustración 7: Tipos de Registros de Recursos correspondiente al dominio "djj.com"

2.5 Vulnerabilidades en el DNS

El DNS tradicional carece de un diseño que asegure el intercambio de información entre un cliente y un servidor, este servicio ha sido comúnmente atacado por delincuentes informáticos que se aprovechan de las vulnerabilidades que este presenta. El RFC 2828 define que "vulnerabilidad es un defecto o debilidad en un sistema de diseño, ejecución o funcionamiento y gestión que podría ser aprovechada para violar la política de seguridad del sistema".

La vulnerabilidad en un sistema DNS es una debilidad de diseño, funcionamiento o gestión que podría ser aprovechada para robo de información, modificación de archivos de caché en el servidor DNS, re direccionamiento de usuarios a sitios falsificados o mal uso de servidores DNS públicos. (Armas, 2017).

2.6 Ataques al DNS

El RFC 2828 indica que "ataque es un intento deliberado (método o técnica) para evadir los servicios de seguridad y violar la política de seguridad de un sistema".

2.6.1 Envenenamiento de caché (Caché Poisoning)

Es un ataque que consiste en almacenar registros falsos en la caché de un DNS recursivo, básicamente el host "víctima" puede ser redirigido a un host malicioso a través de la inyección de un

registro del tipo “A” falsificado. Para ello el “atacante” puede hacer uso de la técnica “hombre en el medio” para capturar y modificar respuestas DNS.

Para que una respuesta sea aceptada por un servidor DNS recursivo que inició una consulta, tanto el ID de la consulta como el ID de la respuesta deben coincidir. De este modo un atacante puede inundar de respuestas (con distintos ID) al servidor hasta lograr acertar con el ID de una consulta previa. Si la respuesta del atacante llega antes que la legítima, el servidor que realizó la consulta la aceptará y la almacenará en la caché. (Armas, 2017).

La técnica más efectiva para lograr el envenenamiento de caché es el método Kaminsky, el cual consiste en el envío de la información falseada en registros adicionales. Para esto el atacante debe conocer los servidores autoritativos que desea suplantar. Es una búsqueda que realiza con el objetivo de suplantar a dichos servidores, lo que le permitirá enviar registros adicionales en sus respuestas.

(1) – El atacante manda una petición DNS al servidor víctima para un nombre de host que quiere comprometer. (1.5) Teniendo en cuenta que el servidor DNS victima pedirá en breve la dirección de www.ejemplo.ni al DNS root, el atacante comienza a inundar con paquetes de respuestas DNS con el fin de que ns.ejemplo.ni tome por buena la respuesta fraudulenta y asigne a www.ejemplo.ni, una IP falsa correspondiente a la que asigna el atacante.
(2) – El servidor DNS victima consulta al DNS root por www.ejemplo.ni.
(3) – DNS root devuelve la @ip del servidor DNS con dominio .ejemplo.ni
(4) – El servidor DNS víctima pide la dirección www.ejemplo.ni al servidor DNS con dominio .ejemplo.ni.
(5) – El servidor de nombres real (ns1.ejemplo.ni) proporciona una respuesta legitima a la petición con un QUERY ID de 1500, pero como el atacante ya ha enviado anteriormente una respuesta al DNS victima con el mismo QUERY ID (paso 1.5), si esta respuesta legitima llega demasiado tarde es ignorada y por consecuente el servidor DNS víctima almacenara en su caché el registro falsificado.
(6) – Con la dirección IP falsa del servidor atacante almacenada en la memoria caché, el servidor ha quedado comprometido por lo que todas las peticiones al servidor DNS para www.ejemplo.ni se redirigen al sitio controlado por el atacante.

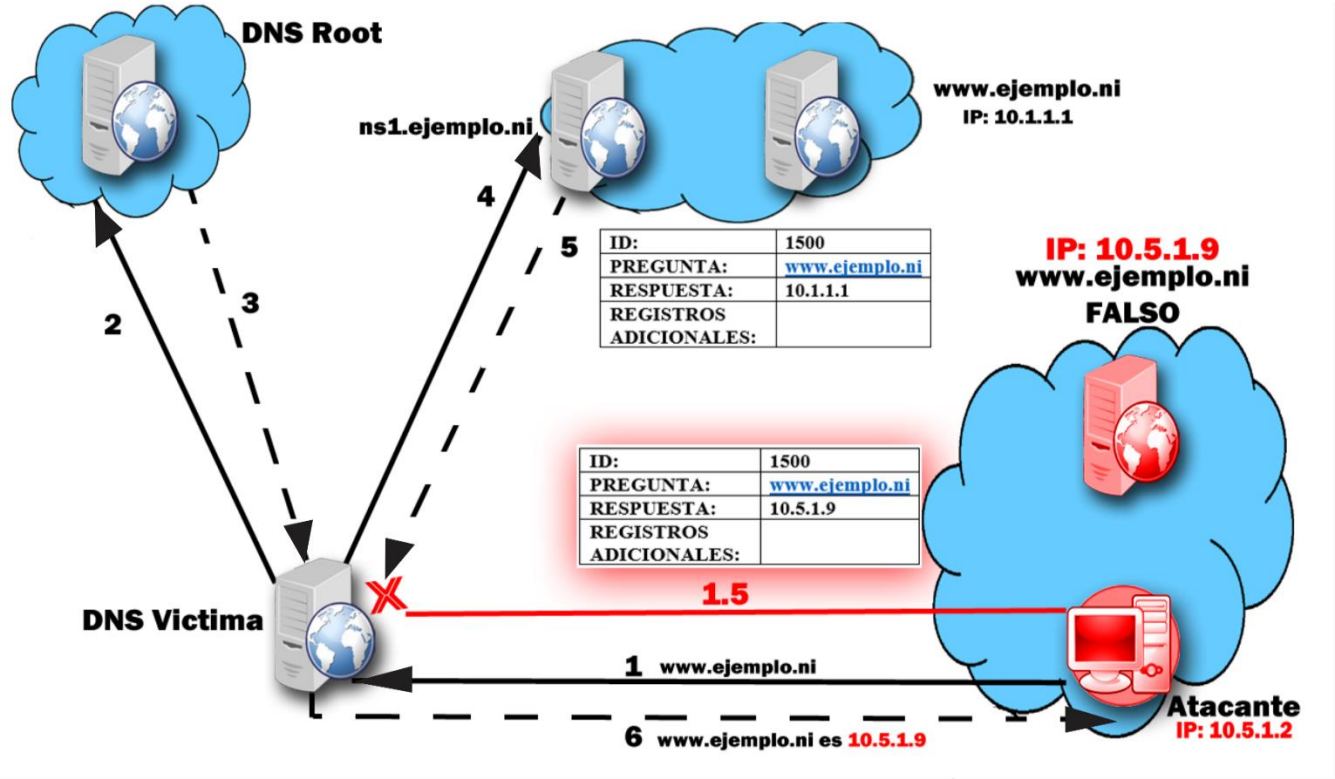


Ilustración 8: Ejemplo de ataque de envenenamiento de caché

2.6.2 DNS Spoofing

Es la suplantación de una identidad por un nombre de dominio. Se trata de la falsificación de una relación "Nombre de dominio-IP" ante una consulta de resolución de nombre, es decir, resolver con una dirección IP falsa un cierto nombre DNS o viceversa. Es un método para alterar las direcciones de los servidores y de esta forma poder tener control sobre las consultas que se realizan.

Los servidores DNS permiten la resolución de nombres en direcciones IP. De esta forma no es necesario para el ser humano recordar las direcciones IP de cada sitio que desea visitar. Aprovechando esta dependencia que existe con los servidores de nombres de dominio, muchos de los atacantes se benefician de este nodo, dentro de la ruta de comunicación, cuando se consulta un sitio web. En otras palabras, alteran las direcciones IP de los servidores DNS de la víctima para que apunten a servidores maliciosos.

Debido al DNS Spoofing es posible realizar diferentes tipos de ataques a través del uso de servidores DNS maliciosos. Uno de los ataques que puede tener más impacto es el de montar sitios

falsos que sean réplica de aquellos que el atacante desee obtener información sensible por parte de la potencial víctima. De esta forma cuando el usuario intente acceder a ese sitio, será re direccionado al sitio espejo y el atacante obtendrá las credenciales. Este tipo de ataques incluso darán resultado sobre aquellos sitios que cifran la conexión, es decir que utilizan el protocolo HTTPS. En otras palabras, como el usuario atacado accede a un sitio malicioso, el cifrado es inexistente y por la tanto se posibilita el robo de las credenciales.

(1) La víctima hace una consulta DNS, a un servidor DNS legítimo.
(2) El atacante obtiene el control de la consulta DNS de la víctima.
(3) La víctima es re dirigida a un sitio web malicioso manipulado por el atacante.

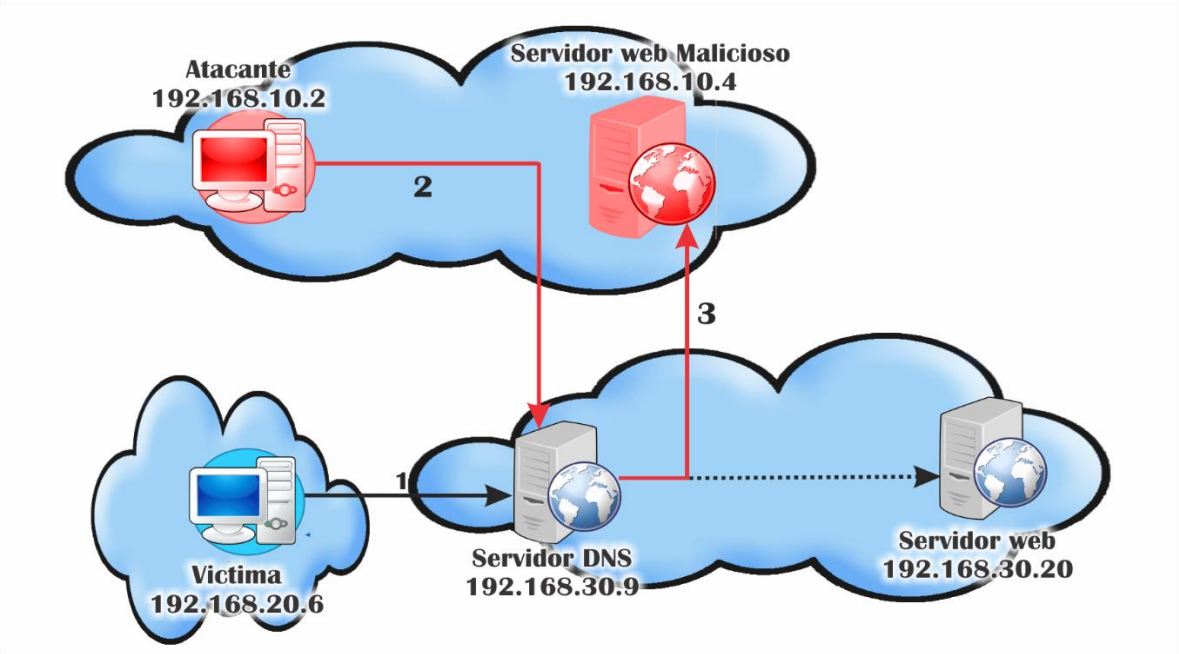


Ilustración 9: Ejemplo de DNS Spoofing

2.6.3 DNS Hijacking

Este ataque consiste en interceptar el intento de un cliente DNS de traducir un nombre de dominio a una dirección IP, con el fin de redirigir al usuario a un sitio diferente. Este ataque suele ser usado con fines maliciosos redirigiendo a los usuarios a sitios web controlados por el atacante. Una vez que los usuarios ingresan a estas páginas, pueden ser víctimas de ataques de phishing o inyección de malware. (Armas, 2017).

El ataque de DNS Hijacking se puede lograr a través de un ataque conocido como hombre en el medio (MITM), este consiste en que el atacante tiene conexiones independientes con las víctimas y transmite mensajes entre ellos, haciéndoles creer que están comunicándose directamente entre sí a través de una conexión privada, cuando en realidad toda la conversación o comunicación es controlada por el atacante, o en otras palabras suplanta a un nodo (un servidor DNS, por ejemplo) por medio de ARP Poisoning. De este modo, se podrá comenzar el sniffing de los paquetes que la víctima envíe (entre ellos las consultas DNS), con lo que el atacante podría anticiparse y responder a las consultas DNS de un usuario.

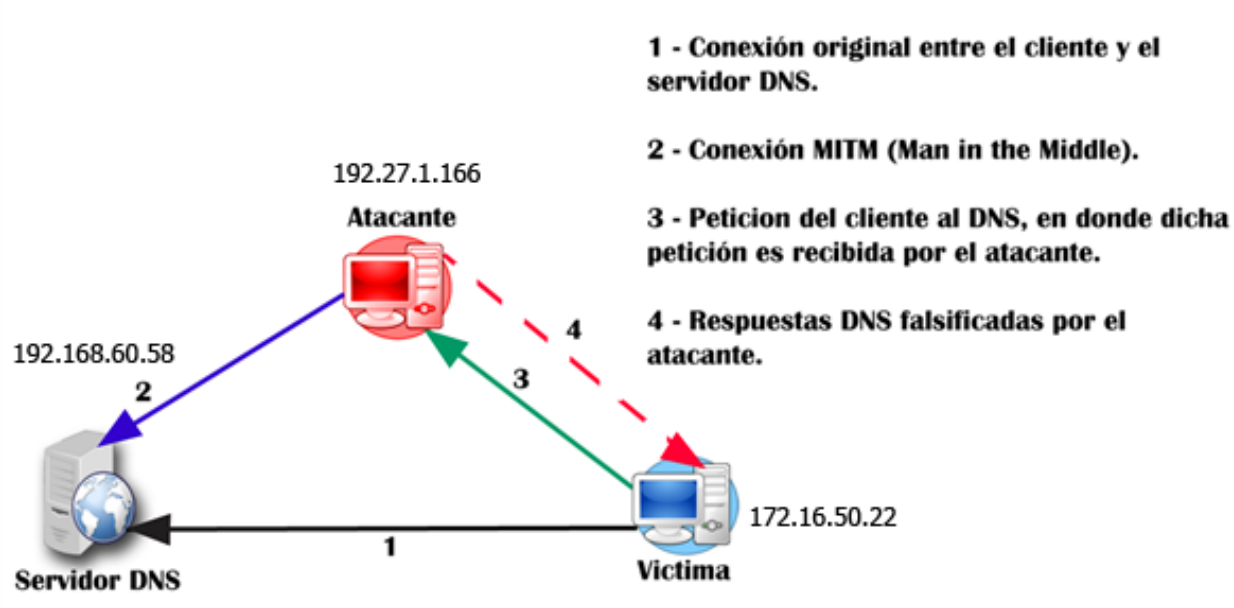


Ilustración 10: Ataque DNS Hijacking por Man in the Middle

2.6.4 Denegación de Servicio

Denegación de servicio, también conocido como ataque DoS (Denial of Service), es un intento malicioso de hacer que un sitio o una aplicación web no estén disponibles para los usuarios legítimos saturando de forma intencionada la infraestructura que soporta el sitio con un enorme volumen de tráfico falso, hasta el punto de que el sitio no puede procesar más solicitudes o lo hace muy lentamente. Un ataque de denegación de servicio puede dirigirse al nivel de red de la pila del protocolo de Internet (como una inundación de solicitudes “SYN” de TCP/IP) o al nivel de aplicación (como una inundación de solicitudes de DNS o HTTP).

Los servidores DNS recursivos son más vulnerables a este tipo de ataques, debido a que utilizan más recursos en la resolución de las consultas. Esto se debe a que necesitan solicitar la información a otros servidores, lo que significa gastar más tiempo y recursos. Además, estos servidores generalmente están abiertos a todo público por lo que son conocidos como open resolvers. (Armas, 2017).

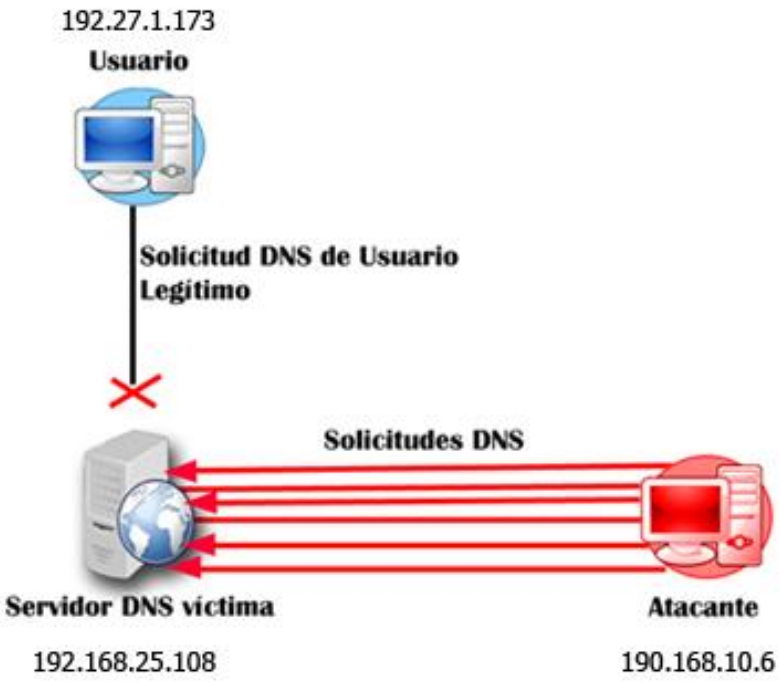


Ilustración 11: Ataque DoS

2.6.5 Denegación de Servicio por Amplificación DNS

Este ataque tiene como objetivo saturar la interfaz de red del servidor víctima para que usuarios legítimos no puedan acceder a los servicios ofrecidos por dicho servidor**.** (Armas, 2017).

La denegación de servicio por amplificación DNS consiste esencialmente en saturar la interfaz de red del servidor víctima por medio de la inundación de respuestas DNS, aprovechando los recursos de servidores DNS públicos para la generación de los paquetes. A estos servidores DNS públicos se les llama servidores de amplificación, como una forma de distinguirlos del servidor víctima.

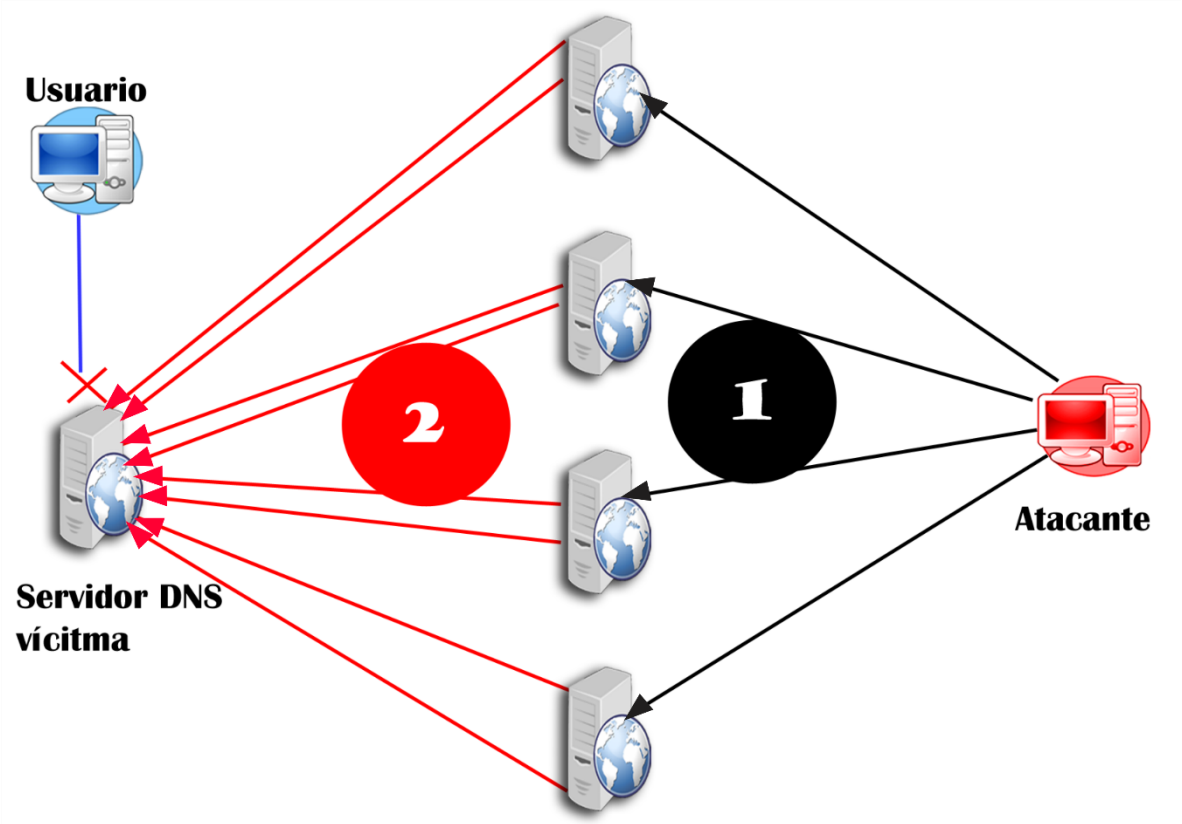


Ilustración 12: Ataque DoS Por Amplificación DNS

Como se puede apreciar en la ilustración 12 este ataque está conformado por dos etapas:

- Etapa 1: El atacante realiza el envío de solicitudes (forjadas con la dirección IP del servidor víctima) falsificadas a los servidores de amplificación, provocando que todas las respuestas correspondientes se dirijan al servidor víctima.
- Etapa 2: Los servidores de amplificación realizan el envío masivo de respuestas DNS hacia el servidor DNS víctima, impidiendo que los usuarios accedan a los servicios debido a que la interfaz de red del servidor víctima está saturado.

2.7 Extensiones de Seguridad DNSSEC

Como se ha mencionado anteriormente debido a las diferentes vulnerabilidades al protocolo DNS, los entes y autoridades correspondientes han venido trabajando en mecanismo de seguridad de lo cual carece el DNS tradicional, en donde las DNSSEC han sido el resultado de estos trabajos.

Tabla 6: Cronograma de las DNSSEC

Año	
1990	Se descubre una falla importante en el DNS y se inicia el diálogo al respecto de la seguridad del DNS.
1995	Las DNSSEC se convierten en un tema oficial de la IETF.
1999	Se completa el protocolo de las DNSSEC (RFC2535) y se desarrolla BIND9 como la primera implementación con funcionalidades DNSSEC.
2001	El manejo de claves genera problemas operativos que impiden la implementación de las DNSSEC para redes de gran tamaño. La IETF decide volver a redactar el protocolo.
2005	Las normas DNSSEC son redactadas nuevamente en varias RFC: 4033, 4034 y 4035. En octubre, Suecia (.se) habilita las DNSSEC en su zona.
2007	En julio, el ccTLD .pr (Puerto Rico) habilita las DNSSEC seguido por el .br (Brasil) en septiembre y el .bg (Bulgaria) en octubre.
2008	Se publica la norma NSEC3 (RFC 5155). En septiembre, el ccTLD .cz (República Checa) habilita las DNSSEC.
2009	Verisign y EDUCAUSE realizan un banco de pruebas de las DNSSEC para registrantes .edu seleccionados. Se firma la zona raíz para uso interno de ICANN y Verisign. ICANN y Verisign practican la firma del ZSK con el KSK.
2010	El primer servidor raíz comienza a atender la raíz firmada mediante la metodología DURZ (raíz intencionalmente no validable). Todos los servidores raíz comienzan a atender la raíz firmada usando la metodología DURZ. ICANN realiza su primera ceremonia KSK en Culpeper, Vancouver, Estados Unidos. ICANN publica el anclaje de veracidad de la raíz y los operadores de raíz comienzan a atender la zona de raíz firmada con claves reales; la raíz firmada está ahora disponible. Verisign y EDUCAUSE habilitan las DNSSEC para el dominio .edu. Verisign habilita las DNSSEC para el dominio .net.

2011	En febrero, se traspasa a Verisign el registro .gov con las DNSSEC habilitadas. En marzo se firma el .com y se mejora el servicio de gerencia de DNS de Verisign para cumplir por completo con las DNSSEC. Se firman 59 TLD con anclajes de veracidad en la zona raíz.
2012	En enero, Comcast anuncia que sus clientes están usando servidores de resolución con validación de DNSSEC. Al mes de marzo se alcanzaron los 90 TLD firmados.

2.7.1 ¿Qué es DNSSEC?

Es un conjunto de extensiones de seguridad que se implementan en el DNS, dicho conjunto provee un nivel adicional de seguridad que protege al DNS contra diferentes tipos de ataques, con la finalidad de que los datos sean válidos para los clientes que generan consultas al DNS.

2.7.2 Funcionamiento de DNSSEC

DNSSEC permite firmar criptográficamente una zona haciendo que los datos pedidos por el cliente sean autenticados por éste, a través de una clave pública siguiendo algoritmos y procesos de encriptación que puedan validar su contenido. (Guanolique, 2014).

Cuando un cliente hace una consulta al servidor DNS, el servidor devuelve las firmas digitales, además de los recursos de registros solicitados. Un cliente u otro servidor, pueden obtener la clave pública, del par de claves pública o privada del servidor consultado, y validarlas para saber que los datos no han sido manipulados, para lo cual el cliente o el servidor deben configurarse con un ancla de confianza para la zona firmada por el servidor de la zona superior. Por consiguiente, DNSSEC proporciona integridad y autenticidad de los datos, pero no garantiza la confidencialidad de los datos.

2.7.3 Registros de Recursos para DNSSEC

DNSSEC hace uso de cuatro nuevos Registros de Recursos y dos bits (CD = Comprobación Desactivada y AD = Dato Autenticado) presentes en el encabezado de un paquete DNS. Por otro lado, un Servidor Resolver que realiza una consulta, utiliza los mecanismos de extensión para DNS (EDNS0) y activa el bit DO (DNSSEC OK), presente en el Registro de Recurso Opcional (RR OPT). Por lo tanto, mediante la activación del bit anterior un Resolver está indicando que tiene la capacidad de procesar información relacionada con DNSSEC. Por otro lado, un Servidor de Nombres que recibe solicitudes en las cuales el bit DO no estuviera presente, no responde con Registros de Recursos

relacionados con DNSSEC, con lo que se contribuye a mejorar el rendimiento de DNS, ya que evita tener que retornar información que posteriormente no será procesada. (Sánchez, 2012).

2.7.4 Tipos de Registros de Recursos para DNSSEC

Los Registros de Recursos para DNSSEC son:

- DNSKEY (DNS public Key): Registro de Recurso habilitado para almacenar claves públicas, que posteriormente serán usadas por DNSSEC en procesos de autenticación.
- RRSIG (Resource Record Signature): Contiene la firma para un conjunto de Registros de Recursos RRSet (Resource Record Set) con un nombre particular, clase y tipo. El registro RRSIG se genera en el proceso de firmado de una zona utilizando la clave privada y cuyo par (clave pública) es almacenada en el registro DNSKEY.
- NSEC (Next Secure): Permite validar la estructura de una zona y los Registros de Recurso que esta contiene.
- DS (Delegation Signer): Permite crear una cadena de confianza o de autoridad de una zona padre firmada, hacia una zona hija firmada. DS está relacionado con el Registro DNSKEY, ya que contiene un resumen (hash o digesto) de la clave (KSK) almacenada en este último.

El bit AD solo tiene sentido en una respuesta DNS, si el mismo almacena el valor 1, está indicando que todos los RRsets en la sección “answer” y toda información pertinente a Registro de Recursos de Respuestas Negativas en la sección “authority” son auténticas, es decir que la validación ha sido exitosa. Es importante destacar que regularmente los Servidores de Nombres Autoritativos, nunca envían respuestas con este bit activo, dado que éstos no verifican las firmas que envían como respuestas. Lo que implica que el bit AD solo es activado por aquellos servidores que han comprobado firmas, tales como los Servidores Resolver.

El bit CD lo activa el Cliente que realiza la consulta para indicar al Servidor de Nombres que está dispuesto a aceptar datos que no hayan sido validados, por lo tanto, el segundo simplemente devuelve una respuesta aun cuando la validación no haya sido exitosa.

Por otro lado, se presenta el caso de un Servidor Resolver con soporte para DNSSEC pero que no realiza validaciones, puede en principio confiar en datos ya validados, si es que se encuentra en una cadena de confianza. Sin embargo, el mejor de los casos se presenta en el uso de stub resolvers que puedan realizar la validación criptográfica y, en consecuencia, al enviar una consulta, el bit CD

se activa a 1. Lo anterior provee un esquema de DNS seguro de extremo a extremo, sin la necesidad de servidores intermedios en los que se deba confiar.

2.7.5 Servicios DNSSEC

DNSSEC ofrece servicios entre el servidor autoritativo y los resolvers security aware (Entidad que actúa en el rol de un resolver, comprendiendo las extensiones de seguridad del DNS). (Barreira, Grassi & Silvera, 2012).

- Autenticación de origen
- Integridad de datos DNS
- Autenticación de negación de existencia de datos DNS.

El protocolo DNS requirió de un cambio para incorporar estos servicios. En donde DNSSec añade cuatro nuevos registros de recursos:

- Resource Record Signature: RRSIG
- DNS Public Key: DNSKEY
- Delegation Signer: DS
- Next Secure: NSEC y NSEC3

También se redefinen dos bits reservados en el encabezado de mensaje:

- Checking Disabled: CD
- Authenticated Data: AD

Con el fin de soportar los mensajes de mayor tamaño del DNS que resultan de agregar estos nuevos registros, los servidores de nombre que soporten DNSSec también deben soportar EDNS0 y el bit DNSSec OK –DO− del encabezado EDNS, con el cual un resolver security-aware pueda indicar en sus consultas si desea recibir registros DNSSec en los mensajes de respuesta.

2.7.6 Cadenas de confianza

La construcción de una cadena de confianza es fundamental para la rápida implementación de DNSSEC en una jerarquía DNS, ya que, sin esta característica, cada Servidor Resolver configurado con DNSSEC, debería tener un punto de anclaje seguro o SEP (estructura DNS con soporte para las

extensiones de seguridad) por cada dominio seguro en Internet, lo que claramente no permitiría un despliegue a escala global de tales extensiones de seguridad. (Sánchez, 2012).

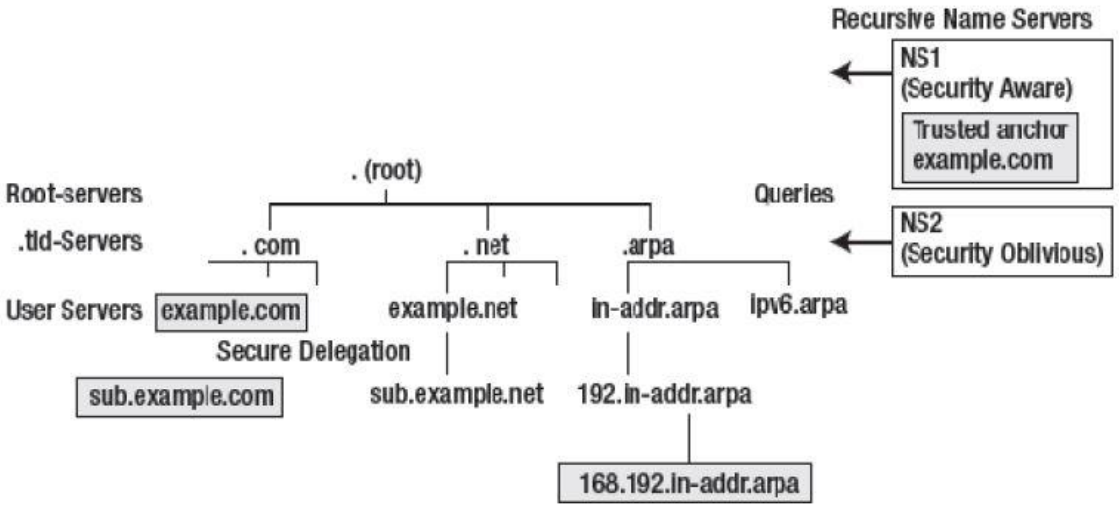


Ilustración 13: Creación de la cadena de confianza
Fuente: (Sanchéz, 2012)

En la ilustración 13 se puede apreciar que el dominio “example.com” y “sub.example.com” (Zona Hija) están asegurados, en otras palabras, la Zona Hija debe de estar asegurada también para que exista una delegación segura, gracias a las zonas seguras creadas por el dominio “example.com” las cuales son delegadas a partir de este mismo, conlleva a que exista una cadena de confianza gracias al Registro de Recurso DS (Delegation Signer).

Una cadena de confianza puede ser construida tanto hacia arriba como hacia abajo en una jerarquía DNS, por lo que si el dominio de nivel superior .com fue asegurado, el dominio “example.com” puede unirse a la cadena.

2.7.7 Autenticación de datos y concepto de ZSK y KSK

Cuando se ha recorrido la cadena de confianza hasta la zona de donde obtenemos los registros de los cuales estamos interesados, y verificada cada iteración de la cadena de confianza, se asume que la clave pública KSK de esta zona es auténtica, esta clave pública tendrá que verificar la ZSK que se utilizará para verificar los datos del DNS, compartiremos los conceptos de Clave de Zona (ZSK) y Clave de Claves para mayor entendimiento.

El RFC 4641, describe que las claves criptográficas que se usan para el firmado de registros que se asocian a un dominio pueden ser de dos tipos:

- ZSK (Zone Signing Key): La función de esta es proteger los Registros de Recursos individuales de una zona dada, la ZSK está almacenada en un Registro de Recurso DNSKEY, obtenerla no es problema, sino autenticarla, dicha autenticación se efectúa en un esquema conocido de la encriptación de clave pública (verificación de firmas digitales)
- KSK (Key Signing Key): Se utiliza para firmar claves en particular la ZSK en donde también se encarga de proteger a esta última.

Estas firmas digitales son almacenadas en un nuevo registro de recurso, el RR RRSIG. Por lo general, hay una sola clave privada que firma los datos de una zona, pero es posible el uso de múltiples claves simultáneas para firmar registros. Si un resolver security-aware aprende confiablemente. (Barreira, Grassi & Silvera, 2012).

Una vez obtenida la clave, el resolver será capaz de autenticar los datos de la zona verificando las firmas digitales, de la siguiente manera:

- Se obtiene el RRset cuya integridad y autenticidad se quiere verificar.
- Se realiza el hash correspondiente a dicho RRset: H(RRset2)
- El resolver des-encripta la firma digital RRSIG correspondiente a dicho RRset, DZSK(RRSIG).
- Si la comparación DZSK(RRSIG) == H(RRset) da correcta, se puede asegurar la autenticidad e integridad de los datos obtenidos.

Uso separado de Claves:

Cuando las claves de zona ya han sido verificadas y por ende actualizadas no es necesaria la interacción entre una Zona Padre y una Zona Hija. La KSK es una clave de gran fortaleza porque puede configurarse con longitudes de clave mayores, esta no consume muchos recursos, debido a que solamente se utiliza para el firmado de una pequeña porción de datos de una zona en especifica.

Para la mayoría de los métodos de administración de claves y firmado de zonas, la clave *KSK* es usada con menor frecuencia que la clave *ZSK*. Una vez que un conjunto de claves es firmado con una clave *KSK*, todas las claves del conjunto pueden ser usadas como claves *ZSK*, si alguna de ellas fuera comprometida, ésta simplemente se elimina del conjunto de claves y el nuevo conjunto de claves debe refirmarse nuevamente con la clave *KSK*. (Sánchez, 2012, p.49)

Revocación de Claves:

Las claves no tienen una fecha de expiración explicita, por lo que es importante reemplazarlas cada periodo de tiempo, la revocación de claves también conocido como "key rollover", es necesario porque permite evitar que un atacante malicioso pueda deducir cualquier tipo de información sin importar el algoritmo usado y longitud de la clave. Otro aspecto muy importante es el hecho de que cuando se realiza la revocación de una clave de zona, la zona debe de ser refirmada en su totalidad, lo que provoca que RRset DNSEKEY se incluya. Para poder realizar la revocación de claves se debe contar con capacidad de procesamiento ya que gracias esta capacidad se puede generar las firmas criptográficas correspondientes.

En el caso de que una clave del tipo KSK es revocada, solamente RRSIG del RRset DNSKEY será refirmado, en donde se utilizaran ambas claves: ZSK y la nueva clave KSK; por consecuencia el administrador de nivel superior es notificado, es decir, la nueva clase KSK se debe de agregar al Registro DS en la zona del nivel superior.

2.7.8 Servidores de Nombres Autoritativos

Llegados a este punto en donde se ha compartido el funcionamiento de manera general de las extensiones de seguridad DNSSEC, se presentan las características que posee la entidad denominada Servidor Autoritativo, que junto a la entidad Resolvers y Clientes DNS, son fundamentales dentro de un proceso de validación de respuesta DNS.

Este servidor debe soportar EDNS debido a las siguientes razones:

- Debe ser capaz de recibir e interpretar paquetes de datos DNS, en donde el bit DO se encuentra activo, esto indica que el cliente ha realizado una consulta, por lo que este espera una respuesta DNSSEC.

- La configuración con EDNS, permite soportar paquetes DNS de mayor longitud, utilizando el protocolo UDP, en otras palabras, supongamos que el paquete posee una longitud de xxxxx bytes, este se puede enviar en un único paquete UDP, esto es posible gracias al agregado de firmas y claves relacionadas, debido a que las respuestas DNSSEC poseen mayor longitud en comparación a una respuesta DNS tradicional.

El Servidor Autoritativo tiene que generar firmas de manera offline para todos los datos sobre los que tiene autoridad, porque de manera online resultaría poco factible debido a las limitaciones de velocidad y todo lo que concierne a la estabilidad y disponibilidad por cada consulta que se reciba.

2.7.9 Servidores de Nombres Caché

El Servidor de Nombres Caché que posee capacidades DNSSEC, intentará mandar sus respectivas consultas, señalando que tiene la capacidad para tratar las respuestas DNSSEC que reciba, por ende, este puede interpretar los nuevos RR de DNSSEC. Estos Servidores validarán las respuestas DNSSEC por su propia cuenta y de esta manera se comprobarán las firmas.

Este servidor cumple dos roles diferentes:

- Recibe consultas de los clientes y comúnmente devuelve una respuesta, en donde utiliza un mismo canal de comunicación, por lo que se considera que cumple el rol de un Servidor.
- Se encarga de resolver las consultas recibidas, por medio del envío de consultas a los Servidores de Nombres Autoritativos, en donde este se comporta como un Cliente.

Los Servidores de Nombres de Caché se distinguen en el RFC 4035, en donde el lado del Servidor es llamado Servidor de Nombres Recursivo y Servidor Resolver para el lado del Cliente.

Cuando un Cliente con soporte para DNSSEC envía una consulta, el Servidor de Nombres Cache activará el bit AD solo si considera que todos los Conjuntos de Registros de Recursos en las secciones “Answer” y “Authority” de la respuesta, son auténticas, o bien si considera que todos los Conjuntos de Registros de Recursos en la sección “Answer” y cualquier Registro de Recurso de Respuesta Negativa en la sección “Authority” son auténticas. (Sánchez, 2012).

Si el Cliente activa el bit CD, el Servidor de Nombres Cache, no realizará ninguna validación, simplemente se comportará como un proxy, reenviando todos los datos, tal cual los ha recibido,

incluido cualquier Registro DNSSEC. Estos datos, incluso podrían provenir de la memoria caché, e incluso tratarse de datos manipulados conteniendo firmas inválidas. En este caso toda la responsabilidad reside en el Cliente. En conclusión, la activación del bit CD en una consulta, simplemente está afirmando que es el Cliente el que realizará la verificación de las firmas. Y lo que es más importante aún, el bit CD no tiene ninguna influencia sobre el bit AD, un Servidor de Nombres Caché podría establecer el bit AD en una respuesta, aun cuando el bit CD haya sido activado en una consulta, indicando que los datos son auténticos.

2.7.10 Stub Resolver

Es la implementación de DNS en un dispositivo host cliente, es un conjunto de librerías encargadas de dialogar con los programas (navegador web, clientes de correo, cliente FTP, etc.) cuando éstos intentan resolver un nombre de dominio. Este conjunto de librerías por lo general se implementa como parte del Sistema Operativo Cliente y no involucran mucho código, dado que la mayor funcionalidad se encuentra en los Servidores de Nombres Cache. (Sánchez, 2012).

2.7.11 Beneficios de DNSSEC

Las extensiones de seguridad DNSSEC tratan de solucionar las vulnerabilidades del DNS esto claramente aplica en el caso en que un usuario o entidad (grupo de personas, empresas, entre otros) trata de obtener alguna información proveniente de internet en donde se espera que dicha información solicitada sea verídica.

A través de los RFC 4033, 4034, 4035, nos dan la pauta de cómo estas extensiones beneficiarían al usuario y a los administradores de servidores de nombres de dominio a que sus datos sean los que corresponda a cada sitio o dominio web. En consecuencia, no sólo los clientes y servidores DNS se beneficiarían de estas extensiones de seguridad, sino que también las diferentes aplicaciones y protocolos que funcionan en la capa de aplicación del modelo TCP/IP. (Guanolique, 2014).

2.7.12 Desventajas de DNSSEC

Las extensiones de seguridad del DNS son complejas de gestionar y que las transferencias de las zonas entre los servidores primario y secundario son de gran tamaño. Estas extensiones no garantizan la confidencialidad de los datos, ya que las peticiones no son encriptadas, tampoco protege contra los Ataques de Denegación de Servicio. (Guanolique, 2014).

Otro problema es quién será la persona que estará a cargo de la administración de las claves, las cuales deben ser actualizadas periódicamente para garantiza la seguridad de las mismas. El proceso para el despliegue de DNSSEC, conlleva a mucha configuración manual, lo que incrementa la posibilidad de error humano en el proceso de configuración.

3 DISEÑO METODOLÓGICO

En esta sección se enumeran los diferentes materiales empleados en este trabajo monográfico en el que se describen los pasos a través de los cuales se abordó la investigación.

Materiales empleados:

Software

Los materiales software utilizados son:

Tabla 7 - Materiales Software

Software	Descripción	Link de descarga
Virtual Box (Versión 6.0.14)	Software de virtualización para arquitecturas x86/64, que permite instalar sistemas operativos adicionales, conocidos como sistemas invitados, dentro de otro sistema operativo anfitrión, cada uno con su entorno independiente.	https://download.virtualbox.org/virtualbox/6.0.14/VirtualBox-6.0.14-133895-Win.exe
GNS3 (Versión 2.1)	Simulador gráfico de red que permite diseñar topologías de red complejas y luego ejecutarlas.	https://www.gns3.com/software/download
Sistema Operativo CentOS 7	Arquitectura x64_86, sistema operativo de código abierto, en donde le ofrece al usuario la facilidad de instalación y uso, sistema resultante de la bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL	http://mirrors.ucr.ac.cr/centos/7.7.1908/isos/x86_64/CentOS-7-x86_64-DVD-1908.iso
Kali Linux 17.3	Arquitectura x64_86, es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general.	https://cdimage.kali.org/kali-2019.3/kali-linux-2019.3-amd64.iso
Sistema Operativo Lubuntu 12.04	Arquitectura x64_86, distribución oficial de Ubuntu, es un sistema operativo que minimiza los recursos necesarios para el funcionamiento de un equipo informático.	http://releases.ubuntu.com/12.04/ubuntu-12.04.5-desktop-i386.iso

Q4os-1.8.5	Arquitectura x64_86, sistema operativo de distribución Linux, ofrece un entorno de escritorio productivo. Se distingue por la velocidad y los requisitos de hardware muy bajos, también es muy aplicable para la virtualización y la computación en la nube.	https://archive.org/download/linuxtracker.org_p1/q4os-1.8.5-i386.iso
Imagen IOS Router c3725 (GNS3)	Imagen Cisco IOS de GNS3, dispositivo hardware emulado que permite la interconexión de ordenadores en red.	https://archive.org/download/c2691-adventerprisek9-mz.124-15.T14/c3725-adventerprisek9-mz.124-25d.bin
Ethernet switch (GNS3)	Dispositivo de hardware o conmutador virtual que viene por defecto en GNS3, se utiliza para filtrar y encaminar paquetes de datos entre segmentos de redes locales y ofrecer conexión a los equipos que conforman una subred LAN.	Dispositivo incluido en el software de GNS3.
Apache2 (Versión 2.4)	Es un servidor web HTTP para plataformas Unix, Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción del sitio virtual. Es de código abierto para la creación de páginas y servicios web.	https://www-us.apache.org/dist//httpd/httpd-2.4.41.tar.gz
DNSSEC	Extensión DNSSEC, permite implementar seguridad a los DNS de la red por medio del cifrado de información y la autenticación del origen.	Extensiones incluidas en el sistema operativo
Bind9 (Versión 9.10.3)	Es un programa de software libre, utilizado en sistemas operativos como UNIX/LINUX, diseñado para trabajar con las extensiones de seguridad DNSSEC.	ftp://ftp.isc.org/isc/bind9/9.10.3/bind-9.10.3.tar.gz

Haveged	Demonio de Linux que se basa en el algoritmo havege, cuya función es generar entropía en el sistema.	https://centos.pkgs.org/7/atomic-x86_64/haveged-1.9.1-2.el7.art.x86_64.rpm.html

Hardware

Los materiales hardware utilizados son:

Tabla 8: Materiales Hardware

Hardware	Descripción
Laptop DELL Inspiron 15 7000 Series	Máquina laptop, con procesador Intel Core i5, Memoria RAM de 8 Gb DDR3, con sistema operativo anfitrión de Windows 10

3.1 Etapas del proyecto

Para cumplir con los objetivos de este este trabajo monográfico, se divide en las siguientes etapas:

3.1.1 Etapas I: de exploración

En esta etapa se recopila información acerca del tema, haciendo búsquedas e investigaciones en libros, sitios web, artículos, etc.

3.1.2 Etapa II: Creación del escenario virtual

En esta etapa se crea el diseño de la topología virtual en GNS3, procediendo a la instalación de las siguientes máquinas virtuales:

- CentOS 7, que funciona como servidor en el árbol jerárquico DNS, definiendo el DNS root, regionales, autoritativos y los locales, también como servidor web.
- Lubuntu 12.04 y Q4os-1.8.5, como máquinas clientes.

- Kali Linux 17.3, como máquina atacante.

3.1.3 Etapa III: Configuración del escenario virtual

Se realizan las configuraciones necesarias para la comunicación entre las máquinas virtuales. Se utiliza la herramienta Bind9 en las máquinas que operarán como servidores DNS creando las respectivas zonas y también Apache2 en los servidores Web.

3.1.4 Etapa IV: Realización de pruebas de seguridad en los servidores DNS, mediante casos prácticos que incluyen ataques a dichos servidores.

Para probar la seguridad de los servidores DNS se emplean métodos de ataques como envenenamiento ARP (Spoofing) y denegación de servicio al Sistema DNS.

3.1.5 Etapa V: Implementación de las DNSSEC

Por medio de la herramienta Bind9, se implementan las extensiones de seguridad DNSSEC en las máquinas virtuales que funcionan como servidores DNS mencionados anteriormente.

3.1.6 Etapa VI: Validación de los resultados

Se evalúa el nivel de seguridad que se puede obtener de las extensiones DNSSEC tras sufrir diferentes ataques y se verifica que tan seguros están los servicios de la red virtualizada.

3.1.7 Etapa VII: Entrega del informe Final

Redacción y organización del documento final, el cual contiene la explicación detallada de las etapas que se emplean en este documento monográfico y la solución a cada uno de los objetivos planteados.

4 DESARROLLO PRÁCTICO

4.1 Práctica 0: Diseño y configuración de un escenario virtual en GNS3 para la implementación de las extensiones de seguridad DNSSEC.

4.1.1 Objetivo General:

Diseñar y configurar un escenario virtual en GNS3 que permita entender la implementación de las extensiones de seguridad DNSSEC a los servidores DNSEC.

4.1.2 Objetivos específicos:

- Detallar las configuraciones y comandos utilizados para la realización de los ataques a los servidores DNS.
- Comprender el funcionamiento Jerárquico de los servidores DNS y los servidores Web.

4.1.3 Introducción:

Al elaborar la practica 0, los lectores o involucrados, serán capaces de concebir la implementación de las DNSSEC a los servidores DNS, en este escenario se llevarán a cabo diferentes ataques a dichos servidores para entender los vulnerables que son, y después implementaran la extensión de seguridad DNSSEC.

4.1.4 Tiempo estimado para la realización de la práctica:

6 horas.

4.1.5 Requerimientos:

Tabla 9: Requerimientos Hardware y Software en práctica 0

Hardware	Software
Computadora con los siguientes requisitos: ❖ Procesador icore5 con velocidad de 2.7 ghz Memoria RAM de 8 GB.	**Emulador de redes GNS3 con los siguientes elementos:** ❖ 8 servidores DNS (Centos 7). ❖ 9 switch. ❖ 2 servidores Web (Lubuntu 12.04). ❖ 2 máquinas Clientes (Windows, Linux). ❖ 1 máquina Atacante (Kali-Linux).

4.1.6 Conocimientos previos:

Para la implementación de esta práctica es necesario controlar y manejar el emulador GNS3, conocer sobre el sistema DNS y configuraciones básicas de los protocolos de enrutamiento dinámicos RIP y EIGRP.

4.1.7 Topología:

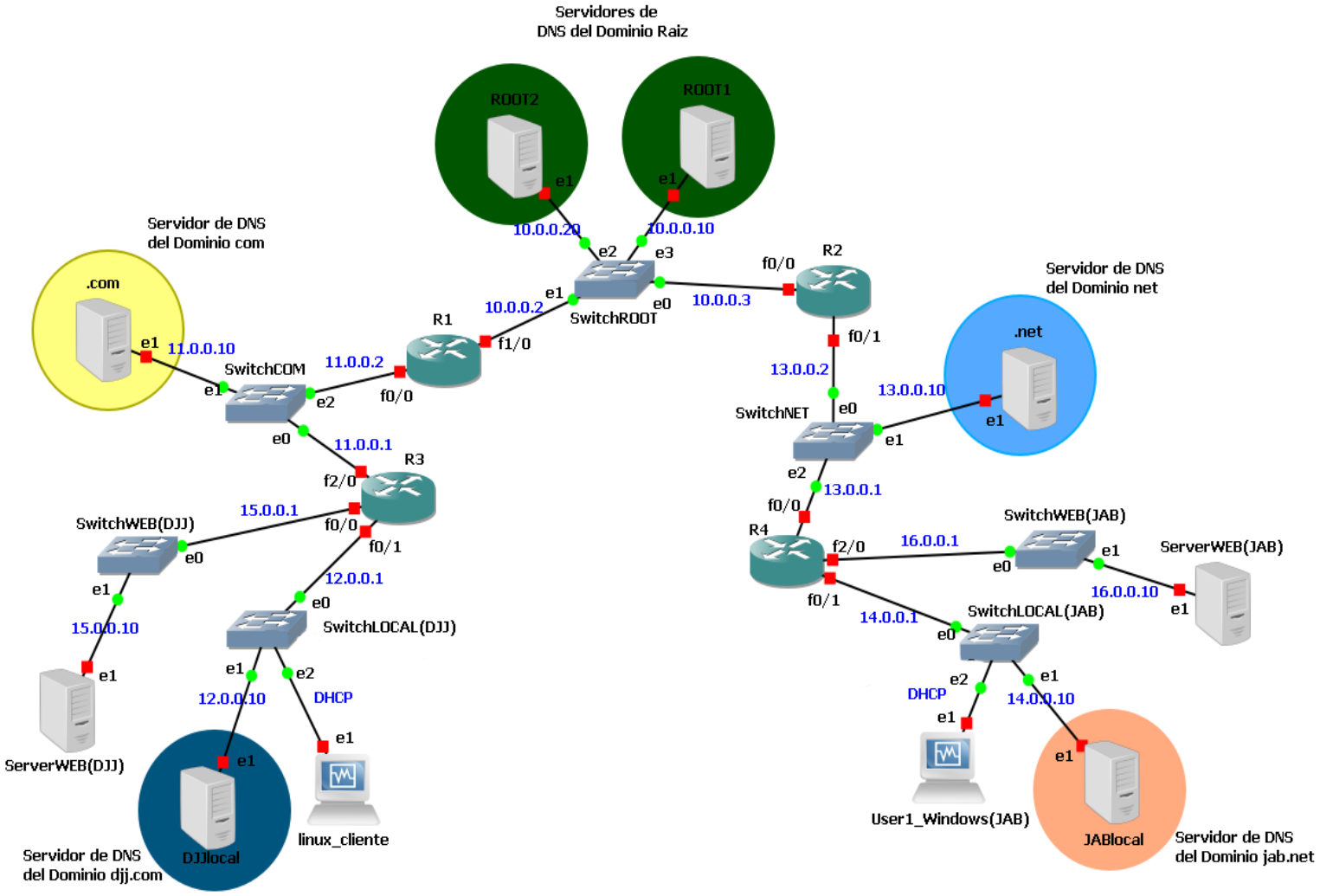


Ilustración 14: Topología de práctica 0

4.1.8 Funcionalidad:

- **General:**

Se logra la configuración de los servidores DNS, dos servidores Web y las máquinas clientes que obtienen ip por medio el protocolo DHCP.

- **Enrutamiento:**

Se consiguió la comunicación interna y externa de los equipos a través de los protocolos de enrutamiento dinámico RIP y EIGRP, dichos protocolos se configuraron en cada uno de los routers, y en las máquinas virtuales se implementó enrutamiento estático.

4.1.9 Enunciado:

Configurar varios servidores bajo el protocolo DNS, y otros servidores bajo el protocolo HTTP(Apache2), efectuar protocolos de enrutamiento dinámico (RIP, EIGRP) y enrutamiento estático, demostrar los ataques hacia el servidor DNS de cada empresa y luego implementar las extensiones de seguridad DNSSEC.

4.1.10 Configuración de enrutadores.

Tabla 10: Configuración RIP en routers

Entidad	Configuración RIP
R1	Interfaz hacia los Dominios Raiz R1#Configure terminal R1(config)#interface FastEthernet 1/0 R1(config –if)#ip address 10.0.0.2 255.255.255.0 Interfaz hacia el Dominio .com R1#Configure terminal R1(config)#interface FastEthernet 0/0 R1(config –if)#ip address 11.0.0.2 255.255.255.0 Configuración RIP R1(config)# router rip R1(config-router)#version 2 R1(config-router)# network 10.0.0.0 R1(config-router)# network 11.0.0.0
R3	Interfaz hacia el dominio .com R3#Configure terminal R3(config)#interface FastEthernet 2/0 R3(config –if)#ip address 11.0.0.1 255.255.255.0 Interfaz hacia el Servidor Web EmpDJJ R3#Configure terminal R3(config)#interface FastEthernet 0/0 R3(config –if)#ip address 15.0.0.1 255.255.255.0

	Interfaz hacia el Dominio EmpDJJ R3#Configure terminal R3(config)#interface FastEthernet 0/1 R3(config –if)#ip address 12.0.0.1 255.255.255.0 Interfaz hacia el Atacante R3#Configure terminal R3(config)#interface FastEthernet 1/0 R3(config –if)#ip address 172.168.0.0 255.255.255.0 Configuración RIP R3(config)# router rip R3(config-router)#version 2 R3(config-router)# network 11.0.0.0 R3(config-router)# network 12.0.0.0 R3(config-router)# network 15.0.0.0 R3(config-router)# network 172.168.0.0 R3(config-router)#ip dhcp pool clientesEmpDJJ R3(config-router)#network 12.0.0.0 255.255.255.0 R3(config-router)#default-router 12.0.0.1 R3(config-router)#dns-server 12.0.0.1 R3(config-router)# ip dhcp excluded-address 12.0.0.1 12.0.0.10

Tabla 11: Configuración EIGRP

Entidad	Configuración EIGRP
R2	Interfaz hacia los dominios raíz R2#Configure terminal R2(config)#interface FastEthernet0/0 R2(config –if)# ip address 10.0.0.3 255.255.255.0 Interfaz hacia el dominio .net R2#Configure terminal R2(config)#interface FastEthernet 0/1 R2(config –if)# ip address 13.0.0.2 255.255.255.0 Configuración EIGRP R2(config)# router eigrp 1 R2(config-router)# network 10.0.0.0 0.0.0.255 R3(config-router)# network 13.0.0.0 0.0.0.255 R3(config-router)# eigrp log-neighbor-changes
R4	Interfaz hacia el dominio .net R4#Configure terminal R4(config)#interface FastEthernet 0/0

	R4(config –if)#ip address 13.0.0.1 255.255.255.0 Interfaz hacia el servidor Web R4#Configure terminal R4(config)#interface FastEthernet 2/0 R4(config –if)#ip address 16.0.0.1 255.255.255.0 Interfaz hacia el servidor DNS jabnet R4#Configure terminal R4(config)#interface FastEthernet 0/1 R4(config –if)#ip address 14.0.0.1 255.255.255.0 Configuración EIGRP R4(config)# router eigrp 1 R4(config-router)# network 13.0.0.0 0.0.0.255 R4(config-router)# network 16.0.0.0 0.0.0.255 R4(config-router)# network 14.0.0.0 0.0.0.255 R4(config-router)# network 172.168.0.0 0.0.0.255 R4(config-router)# eigrp log-neighbor-changes R4(config-router)#ip dhcp pool clientesEmpJAB R4(config-router)#network 14.0.0.0 255.255.255.0 R4(config-router)#default-router 14.0.0.1 R4(config-router)#dns-server 14.0.0.10 R4(config-router)# ip dhcp excluded-address 14.0.0.1 14.0.0.10

4.1.11 Configuración de Red de las Máquinas Virtuales.

Tabla 12: Configuración de red las máquinas virtuales de la topología

Entidad	Configuraciones de Red (Máquinas Virtuales)
Servidor ROOT1	Identificación de las interfaces de red del servidor: • #ip add Configuración de la interfaz de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar, en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Una vez ahí, se busca el archivo con el nombre que se ha localizado anteriormente utilizando el comando “ip add”, en este caso es enp0s8; y se efectúa con el comando “ls” • # ls

	Este archivo llamado ifcfg-enp0s8 es donde se encuentra la configuración de la interfaz de red y se edita con el comando "nano" pero con privilegios root: • # sudo su • # nano ifcfg-enp0s8 Dentro del archivo ifcfg-enp0s8 se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no IPV6_FAILURE_FATAL=no NAME=enp0s8 UUID=f7... ...229 ONBOOT=yes IPADDR0=10.0.0.10 PREFIX0=24 DNS=10.0.0.10 ZONE=public Se reinician las interfaces: • # systemctl restart network.service Se crea un archivo, el cual contendrá las configuraciones de enrutamiento estático: • # cd /etc/sysconfig/network-scripts/ • # mkdir route-enp0s8 • # nano route-enp0s8 Se ingresan los siguientes contenidos al archivo route-enp0s8: ADDRESS0=11.0.0.0 NETMASK0=255.255.255.0

	GATEWAY0=10.0.0.2 ADDRESS1=12.0.0.0 NETMASK1=255.255.255.0 GATEWAY1=10.0.0.2 ADDRESS2=13.0.0.0 NETMASK2=255.255.255.0 GATEWAY2=10.0.0.3 ADDRESS3=14.0.0.0 NETMASK3=255.255.255.0 GATEWAY3=10.0.0.3 ADDRESS4=15.0.0.0 NETMASK4=255.255.255.0 GATEWAY4=10.0.0.2 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=10.0.0.3 ADDRESS6=172.168.0.0 NETMASK6=255.255.255.0 GATEWAY6=10.0.0.2 Activación del bit de forwarding • #nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • #net.ipv4.ip_forward = 1
Servidor ROOT2	Identificación de las interfaces de red del servidor: • #ip add

	Configuración de las interfaces de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando "ip add", en este caso la interfaz se identifica como enp0s8; y lo harás con el comando "ls": • # ls Se edita el archivo ifcfg-enp0s8 con privilegios de super usuario root, utilizando el comando "nano": • # sudo su • #nano ifcfg-enp0s8 Dentro del archivo se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no IPV6_FAILURE_FATAL=no NAME=enp0s8 UUID=f7... ...229 ONBOOT=yes IPADDR0=10.0.0.20 PREFIX0=24 DNS=10.0.0.10 ZONE=public Se reinician las interfaces: • # systemctl restart network.service

	Creación de un archivo, el cual contendrá las configuraciones de enrutamiento estático: • # cd /etc/sysconfig/network-scripts/ • # mkdir route-enp0s8 • # nano route-enp0s8 Se ingresan los siguientes contenidos al archivo route-enp0s8: ADDRESS0=11.0.0.0 NETMASK0=255.255.255.0 GATEWAY0=10.0.0.2 ADDRESS1=12.0.0.0 NETMASK1=255.255.255.0 GATEWAY1=10.0.0.2 ADDRESS2=13.0.0.0 NETMASK2=255.255.255.0 GATEWAY2=10.0.0.3 ADDRESS3=14.0.0.0 NETMASK3=255.255.255.0 GATEWAY3=10.0.0.3 ADDRESS4=15.0.0.0 NETMASK4=255.255.255.0 GATEWAY4=10.0.0.2 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=10.0.0.3 ADDRESS6=172.168.0.0 NETMASK6=255.255.255.0 GATEWAY6=10.0.0.2 Se activa el bit de forwarding #nano /etc/sysctl.conf

	Se cambia la siguiente opción a 1 #net.ipv4.ip_forward = 1
Servidor .com	Identificación de las interfaces de red del servidor: • #ip add Configuración de las interfaces de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando "ip add", en este caso la interfaz se identifica como enp0s8; y se realiza con el comando "ls": • # ls El archivo ifcfg-enp0s8 es donde se encuentra la configuración de la interfaz de red y se edita con el comando "nano" pero con privilegios root: • # sudo su • #nano ifcfg-enp0s8 En el archivo se realizan las siguientes modificaciones: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no IPV6_FAILURE_FATAL=no NAME=enp0s8 UUID=f7... ...229 ONBOOT=yes IPADDR0=11.0.0.10 PREFIX0=24

DNS=10.0.0.10
ZONE=public

Se reinician las interfaces:

- # systemctl restart network.service

Creación de un archivo, el cual contendrá las configuraciones de enrutamiento estático:

- # cd /etc/sysconfig/network-scripts/
- # mkdir route-enp0s8
- # nano route-enp0s8

Se ingresan los siguientes contenidos al archivo route-enp0s8:

ADDRESS0=10.0.0.0
NETMASK0=255.255.255.0
GATEWAY0=11.0.0.2

ADDRESS1=12.0.0.0
NETMASK1=255.255.255.0
GATEWAY1=11.0.0.1

ADDRESS2=13.0.0.0
NETMASK2=255.255.255.0
GATEWAY2=11.0.0.2

ADDRESS3=14.0.0.0
NETMASK3=255.255.255.0
GATEWAY3=11.0.0.2

ADDRESS4=15.0.0.0
NETMASK4=255.255.255.0
GATEWAY4=11.0.0.1

ADDRESS5=16.0.0.0
NETMASK5=255.255.255.0
GATEWAY5=11.0.0.2

ADDRESS6=172.168.0.0

	NETMASK6=255.255.255.0 GATEWAY6=11.0.0.1 Se activa el bit de forwarding • #nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • #net.ipv4.ip_forward = 1
Servidor .net	Identificación de las interfaces de red del servidor: • #ip add Configuración de las interfaces de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando "ip add", en este caso la interfaz se identifica como enp0s8; y lo harás con el comando "ls": • # ls El archivo ifcfg-enp0s8 es donde se encuentra la configuración de interfaz de red y se edita con el comando "nano" pero con privilegios de root: • # sudo su • #nano ifcfg-enp0s8 Dentro del archivo se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no

IPV6_FAILURE_FATAL=no
NAME=enp0s8
UUID=f7... ...229
ONBOOT=yes
IPADDR0=13.0.0.10
PREFIX0=24
DNS=10.0.0.10
ZONE=public

Se reinician las interfaces:

- # systemctl restart network.service

Se crea un archivo, el cual contendrá las configuraciones de enrutamiento estático:

- # cd /etc/sysconfig/network-scripts/
- # mkdir route-enp0s8
- # nano route-enp0s8

Se ingresan los siguientes contenidos al archivo route-enp0s8:

ADDRESS0=10.0.0.0
NETMASK0=255.255.255.0
GATEWAY0=13.0.0.2

ADDRESS1=11.0.0.0
NETMASK1=255.255.255.0
GATEWAY1=13.0.0.2

ADDRESS2=12.0.0.0
NETMASK2=255.255.255.0
GATEWAY2=13.0.0.2

ADDRESS3=14.0.0.0
NETMASK3=255.255.255.0
GATEWAY3=13.0.0.1

ADDRESS4=15.0.0.0

	NETMASK4=255.255.255.0 GATEWAY4=13.0.0.2 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=13.0.0.1 ADDRESS6=172.168.0.0 NETMASK6=255.255.255.0 GATEWAY6=13.0.0.1 Se activa el bit de forwarding • #nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • #net.ipv4.ip_forward = 1
Servidor djj.com (DJJlocal)	Identificación las interfaces de red de tu servidor: • #ip add Configuración de las interfaces de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando “ip add”, en este caso la interfaz se identifica como enp0s8; y lo harás con el comando “ls”: • # ls El archivo ifcfg-enp0s8 es donde se encuentra la configuración de interfaz de red y se edita con el comando “nano” pero con privilegios de root: • # sudo su • #nano ifcfg-enp0s8 Dentro del archivo se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no

BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=no
IPV6_DEFROUTE=no
IPV6_FAILURE_FATAL=no
NAME=enp0s8
UUID=f7... ...229
ONBOOT=yes
IPADDR0=12.0.0.10
PREFIX0=24
GATEWAY0=12.0.0.1
DNS=11.0.0.10
ZONE=public

Se reinician las interfaces:

- # systemctl restart network.service

Se crea un archivo, el cual contendrá las configuraciones de enrutamiento estático:

- # cd /etc/sysconfig/network-scripts/
- # mkdir route-enp0s8
- # nano route-enp0s8

Se ingresan los siguientes contenidos al archivo route-enp0s8:

ADDRESS0=10.0.0.0
NETMASK0=255.255.255.0
GATEWAY0=12.0.0.1

ADDRESS1=11.0.0.0
NETMASK1=255.255.255.0
GATEWAY1=12.0.0.1

ADDRESS2=13.0.0.0
NETMASK2=255.255.255.0
GATEWAY2=12.0.0.1

	ADDRESS3=14.0.0.0 NETMASK3=255.255.255.0 GATEWAY3=12.0.0.1 ADDRESS4=15.0.0.0 NETMASK4=255.255.255.0 GATEWAY4=12.0.0.1 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=12.0.0.1 ADDRESS6=172.168.0.0 NETMASK6=255.255.255.0 GATEWAY6=12.0.0.1 Se activa el bit de forwarding • #nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • #net.ipv4.ip_forward = 1
Servidor jab.net (JABlocal)	Identificación de las interfaces de red del servidor: • #ip add Configuración de las interfaces de red en CentOS7. Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando “ip add”, en este caso la interfaz se identifica como enp0s8; y lo harás con el comando “ls”: • # ls El archivo ifcfg-enp0s8 es donde se encuentra la configuración de interfaz de red y se edita con el comando “nano” pero con privilegios de root: • # sudo su

	• #nano ifcfg-enp0s8 Dentro del archivo se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no IPV6_FAILURE_FATAL=no NAME=enp0s8 UUID=f7... ...229 ONBOOT=yes IPADDR0=14.0.0.10 PREFIX0=24 GATEWAY0=14.0.0.1 DNS=13.0.0.10 ZONE=public Se reinician las interfaces: • # systemctl restart network.service Se crea un archivo, el cual contendrá las configuraciones de enrutamiento estático: • # cd /etc/sysconfig/network-scripts/ • # mkdir route-enp0s8 • # nano route-enp0s8 Se ingresan los siguientes contenidos al archivo route-enp0s8: ADDRESS0=10.0.0.0 NETMASK0=255.255.255.0 GATEWAY0=14.0.0.1 ADDRESS1=11.0.0.0 NETMASK1=255.255.255.0

	GATEWAY1=14.0.0.1 ADDRESS2=12.0.0.0 NETMASK2=255.255.255.0 GATEWAY2=14.0.0.1 ADDRESS3=13.0.0.0 NETMASK3=255.255.255.0 GATEWAY3=14.0.0.1 ADDRESS4=15.0.0.0 NETMASK4=255.255.255.0 GATEWAY4=14.0.0.1 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=14.0.0.1 ADDRESS6=172.168.0.0 NETMASK6=255.255.255.0 GATEWAY6=14.0.0.1 Se activa el bit de forwarding • #nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • #net.ipv4.ip_forward = 1
Servidor Web (DJJ)	Identificación de las interfaces de red del servidor: • #ip add Configuración de las interfaces de red en Lubuntu. Se abre el archivo de configuración a modificar, por lo que se debe ir la siguiente ruta: • # nano /etc/network/interfaces

	Se ingresan las siguientes configuraciones de red: auto eth2 iface eth2 inet static address 15.0.0.10 network 15.0.0.0 netmask 255.255.255.0 gateway 15.0.0.1 Se reinician las interfaces: • #/etc/init.d/networking restart
Servidor Web (JAB)	Identificación de las interfaces de red del servidor: • #ip add Configuración de las interfaces de red en Lubuntu Se abre el archivo de configuración a modificar, por lo que se debe ir la siguiente ruta: • # nano /etc/network/interfaces Se ingresan las siguientes configuraciones de red: auto eth2 iface eth2 inet static address 16.0.0.10 network 16.0.0.0 netmask 255.255.255.0 gateway 16.0.0.1 Se reinician las interfaces: • #/etc/init.d/networking restart
Máquina Linux_cliente	Identificación de las interfaces de red: • #ip add Configurar la interfaz de red en Q4os Se abre el archivo de configuración a modificar en la siguiente ruta: • # nano /etc/network/interfaces

	Se ingresa la siguiente configuración de red: auto eth2 iface eth2 inet dhcp Se reinician las interfaces: • #/etc/init.d/networking restart
Máquina User1 JAB	Configuración por DHCP desde el router R4.

4.1.12 Configuración de Bind en los servidores DNS.

Tabla 13: Configuración del servicio Bind en los servidores DNS

Entidad	Configuraciones

Servidor de dominio raíz ROOT1	Instalación de bind9:
	• yum update
	• yum install bind bind-utils
	
	El archivo de configuración de BIND se encuentra en:
	• nano /etc/named.conf
	
	Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en caso de tenerlo, por lo que se usan las siguientes líneas:
	
	• allow-query { localhost;IP/Red; };
	• allow-transfer { Ip segundo servidor; };
	
	Configuración de la zona primaria y zona inversa:
	
	zone “arpa” IN {
	type master;
	file “db.arpa”;
	allow-update { none; };
	};
	zone “in-addr.arpa” IN {
	type master;
	file “db.arpa.in-addr”;
	allow-update { none; };
	};

```
zone “10.in-addr.arpa” IN {
        type master;
        file “db.10”;
        allow-update { none; };
};
```

Se crean dos nuevos archivos correspondientes a la zona primaria:

- nano /var/named/db.arpa
- nano /var/named/db.10

Una vez creado los dos archivos, se ingresa lo siguiente:

```
  GNU nano 2.3.1              Fichero: db.arpa

$TTL 3H
@       IN SOA  ROOT-SERVER1.           root.ROOT-SERVER1. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN NS   ROOT-SERVER1.
@       IN NS   ROOT-SERVER2.

in-addr.arpa.   IN NS   ROOT-SERVER1.
in-addr.arpa.   IN NS   ROOT-SERVER2.
com     IN      A       11.0.0.10
djj.com IN      A       12.0.0.10
net     IN      A       13.0.0.10
jab.net IN      A       14.0.0.10


                        [ 17 líneas escritas ]
```

```
  GNU nano 2.3.1            Fichero: /var/named/db.10

$TTL 3H
@       IN SOA  ROOT-SERVER1.           root.ROOT-SERVER1. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      ROOT-SERVER1.
@       IN      NS      ROOT-SERVER2.

10.0.0.10.in-addr.arpa. IN      PTR     dnsroot1.
10.0.0.10.in-addr.arpa. IN      PTR     ROOT-SERVER1.

20.0.0.10.in-addr.arpa. IN      PTR     dnsroot2.
20.0.0.10.in-addr.arpa. IN      PTR     ROOT-SERVER2.
```

Creación del archivo para la zona inversa:

- nano /var/named/db.arpa.in-addr

```
 GNU nano 2.3.1       Fichero: /var/named/db.arpa.in-addr

$TTL 3H
@       IN SOA  ROOT-SERVER1.           root.ROOT-SERVER1. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN NS   ROOT-SERVER1.

10.in-addr.arpa.        IN      NS      ROOT-SERVER1.
11.in-addr.arpa.        IN      NS      dnscom.com.
12.in-addr.arpa.        IN      NS      www.djj.com.
13.in-addr.arpa.        IN      NS      dnsnet.net.
14.in-addr.arpa.        IN      NS      www.jab.net.
```

Servidor de dominio raíz ROOT2	Instalación de bind9:
	• yum update
	• yum install bind bind-utils
	
	El archivo de configuración de BIND se encuentra en la siguiente ruta:
	• nano /etc/named.conf
	
	Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en caso de tenerlo, por lo que se usan las siguientes líneas:
	• allow-query { localhost;IP/Red; };
	• allow-transfer { Ip segundo servidor; };
	
	Configuración de la zona primaria y zona inversa:
	
	zone “arpa” IN {
	type master;
	file “db.arpa”;
	allow-update { none; };
	};
	zone “in-addr.arpa” IN {
	type master;
	file “db.arpa.in-addr”;
	allow-update { none; };
	};
	zone “10.in-addr.arpa” IN {
	type master;

	file “db.10”; allow-update { none; }; };

Se crean dos nuevos archivos correspondientes a la zona primaria:

- nano /var/named/db.arpa
- nano /var/named/db.10

Una vez creado los dos archivos, se ingresa lo siguiente:

```
  GNU nano 2.3.1          Fichero: /var/named/db.arpa

$TTL 3H
@       IN SOA  ROOT-SERVER2.           root.ROOT-SERVER2. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN NS   ROOT-SERVER2.
@       IN NS   ROOT-SERVER1.

in-addr.arpa.   IN NS   ROOT-SERVER2.
in-addr.arpa.   IN NS   ROOT-SERVER1.
```

```
  GNU nano 2.3.1          Fichero: /var/named/db.10

$TTL 3H
@       IN SOA  ROOT-SERVER2.           root.ROOT-SERVER2. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      ROOT-SERVER1.
@       IN      NS      ROOT-SERVER2.

10.0.0.10.in-addr.arpa. IN      PTR     dnsroot1.
10.0.0.10.in-addr.arpa. IN      PTR     ROOT-SERVER1.

20.0.0.10.in-addr.arpa. IN      PTR     dnsroot2.
20.0.0.10.in-addr.arpa. IN      PTR     ROOT-SERVER2.
```

Creación del archivo para la zona inversa:

- nano /var/named/db.arpa.in-addr

```
 GNU nano 2.3.1       Fichero: /var/named/db.arpa.in-addr

$TTL 3H
@       IN SOA  ROOT-SERVER2.           root.ROOT-SERVER2. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      ROOT-SERVER2.

10.in-addr.arpa.        IN      NS      ROOT-SERVER2.
11.in-addr.arpa.        IN      NS      dnscom.com.
12.in-addr.arpa.        IN      NS      djj.com.
13.in-addr.arpa.        IN      NS      dnsnet.net.
14.in-addr.arpa.        IN      NS      jab.net.
```

Servidor de dominio .com

Instalación de bind9:

- yum update
- yum install bind bind-utils

El archivo de configuración de BIND se encuentra en la siguiente ruta:

- nano /etc/named.conf

Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en caso de tenerlo, por lo que se usan las siguientes líneas:

- allow-query { localhost;IP/Red; };
- allow-transfer { Ip segundo servidor; };

Configuración de la zona primaria y zona inversa:

```
zone “com” IN {
        type master;
        file “db.com”;
        allow-update { none; };
};

zone “11.in-addr.arpa” IN {
        type master;
        file “db.11”;
        allow-update { none; };
};
```

Se crea un archivo correspondiente a la zona primaria y dentro de este se configura lo siguiente:

- nano /var/named/db.com

```
  GNU nano 2.3.1          Fichero: /var/named/db.com

$TTL 3H
@       IN SOA  dnscom.com.             root.dnscom.com. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@               IN      NS      dnscom.com.
dnscom.com.     IN      A       11.0.0.10

djj.com.        IN      NS      www.djj.com.
djj.com.        IN      A       12.0.0.10
www.djj.com.    IN      A       15.0.0.10
```

Creación del archivo de la zona inversa:

- nano /var/named/db.11

```
  GNU nano 2.3.1          Fichero: /var/named/db.11

$TTL 3H
@       IN SOA  dnscom.com.             root.dnscom.com. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      dnscom.com.
10.0.0.11.in-addr.arpa. IN      PTR     dnscom.com.
```

Servidor de dominio .net	Instalación de bind9: • yum update • yum install bind bind-utils El archivo de configuración de BIND se encuentra en la siguiente ruta: • nano /etc/named.conf Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en caso de tenerlo, por lo que se usan las siguientes líneas: • allow-query { localhost;IP/Red; };

	• allow-transfer { Ip segundo servidor; };
	
	Configuración de la zona primaria y zona inversa:
	
	zone “net” IN {
	type master;
	file “db.com”;
	allow-update { none; };
	};
	zone “13.in-addr.arpa” IN {
	type master;
	file “db.13”;
	allow-update { none; };
	};

Se crea un archivo correspondiente a la zona primaria y dentro de este se configura lo siguiente:

- nano /var/named/db.net

```
  GNU nano 2.3.1          Fichero: /var/named/db.net

$TTL 3H
@       IN SOA  dnsnet.net.             root.dnsnet.net. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      dnsnet.net.
dnsnet.net.     IN      A       13.0.0.10

jab.net.        IN      NS      www.jab.net.
jab.net.        IN      A       14.0.0.10
www.jab.net.    IN      A       16.0.0.10
```

Creación del archivo de la zona inversa:

- nano /var/named/db.13

```
 GNU nano 2.3.1          Fichero: /var/named/db.13

$TTL 3H
@       IN SOA  dnsnet.net.             root.dnsnet.net. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      dnsnet.net.
10.0.0.13.in-addr.arpa. IN      PTR     dnsnet.net.
```

Servidor de dominio local djj.com (DJJlocal)

Instalación de bind9:

- yum update
- yum install bind bind-utils

El archivo de configuración de BIND se encuentra en la siguiente ruta:

- nano /etc/named.conf

Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en caso de tenerlo, por lo que se usan las siguientes líneas:

- allow-query { localhost;IP/Red; };
- allow-transfer { Ip segundo servidor; };

Configuración de la zona primaria y zona inversa:

```
zone “djj.com” IN {
        type master;
        file “db.djj.com”;
        allow-update { none; };
};

zone “12.in-addr.arpa” IN {
        type master;
        file “db.12”;
        allow-update { none; };
};
```

Se crea un archivo correspondiente a la zona primaria y dentro de este se configura lo siguiente:

- nano /var/named/db.djj.com

```
  GNU nano 2.3.1        Fichero: /var/named/db.djj.com

$TTL 3H
@       IN SOA  www.djj.com.            root.www.djj.com. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.djj.com.
@       IN      A       12.0.0.10
www     IN      A       15.0.0.10
emp1    IN      CNAME   www
```

Creación del archivo de la zona inversa:

- nano /var/named/db.12

```
  GNU nano 2.3.1          Fichero: /var/named/db.12

$TTL 3H
@       IN SOA  www.djj.com.            root.localhost. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.djj.com.
10.0.0.12.in-addr.arpa. IN      PTR     www.djj.com.
```

Servidor de dominio local jab.net (JABlocal)	Instalación de bind9: • yum update • yum install bind bind-utils El archivo de configuración de BIND se encuentra en la siguiente ruta: • nano /etc/named.conf Se agrega la dirección IP sobre la cual los equipos cliente realizan consultas al DNS, adicional a esto se agrega la dirección IP de un segundo servidor en

	caso de tenerlo, por lo que se usan las siguientes líneas:
	• allow-query { localhost;IP/Red; };
	• allow-transfer { Ip segundo servidor; };
	
	Configuración de la zona primaria y zona inversa:
	zone “jab.net” IN {
	type master;
	file “db.jab.net”;
	allow-update { none; };
	};
	
	zone “14.in-addr.arpa” IN {
	type master;
	file “db.14”;
	allow-update { none; };
	};

Se crea un archivo correspondiente a la zona primaria y dentro de este se configura lo siguiente:

- nano /var/named/db.jab.net

```
  GNU nano 2.3.1        Fichero: /var/named/db.jab.net

$TTL 3H
@       IN SOA  www.jab.net.            root.www.jab.net. (
                                        060220182       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.jab.net.
@       IN      A       14.0.0.10
www     IN      A       16.0.0.10
emp2    IN      CNAME   www
```

Creación del archivo de la zona inversa:

- nano /var/named/db.13

```
  GNU nano 2.3.1          Fichero: /var/named/db.14

$TTL 3H
@       IN SOA  www.jab.net.            root.localhost. (
                                        060220182       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.jab.net.
10.0.0.14.in-addr.arpa. IN      PTR     www.jab.net.
```

4.1.13 Configuración de Apache2 en los servidores web.

Tabla 14: Configuración del servicio Apache2 en los servidores web de la topología

Entidad	Configuraciones
ServerWeb (DJJ)	Instalación de Apache2 • sudo apt update • sudo apt install apache2 Creación de una carpeta dentro del directorio: • /var/www/ Creación de una página HTML “index.html” dentro de la carpeta creada anteriormente: • nano /var/www//DJJ/index.html Creación de un archivo de alojamiento virtual con las directivas apropiadas, es decir, que la configuración predeterminada que se encuentra en “/etc/apache2/sites-available/000-default.conf” no se modifica por lo que se crea un nuevo archivo: • etc/apache2/sites-available/djj.com.conf Configuración del archivo djj.com.conf <VirtualHost *:80> ServerAdmin admin@djj.com ServerName djj.com ServerAlias www.djj.com DocumentRoot /var/www/DJJ/ ErrorLog ${APACHE_LOG_DIR}/error.log CustomLog ${APACHE_LOG_DIR}/access.log combined </VirtualHost> Se habilita el archivo usando la herramienta a2ensite: • sudo a2ensite ejemplo.com.conf Se deshabilita el sitio por defecto definido en 000-default.conf: • sudo a2dissite 000-default.conf Se prueba la configuración en busca de errores: • sudo apache2ctl configtest Se reinicia Apache: /etc/init.d/networking apache2

ServerWeb (JAB)	Instalación de Apache2 • sudo apt update • sudo apt install apache2 Creación de una carpeta dentro del directorio: • /var/www/ Creación de una página HTML “index.html” dentro de la carpeta creada anteriormente: • nano /var/www//JAB/index.html Creación de un archivo de alojamiento virtual con las directivas apropiadas, es decir, que la configuración predeterminada que se encuentra en “/etc/apache2/sites-available/000-default.conf” no se modifica por lo que se crea un nuevo archivo: • etc/apache2/sites-available/jab.com.conf Configuración del archivo jab.com.conf <VirtualHost *:80> ServerAdmin admin@jab.com ServerName jab.com ServerAlias www.jab.com DocumentRoot /var/www/JAB/ ErrorLog ${APACHE_LOG_DIR}/error.log CustomLog ${APACHE_LOG_DIR}/access.log combined </VirtualHost> Se habilita el archivo usando la herramienta a2ensite: • sudo a2ensite ejemplo.com.conf Se deshabilita el sitio por defecto definido en 000-default.conf: • sudo a2dissite 000-default.conf Se prueba la configuración en busca de errores: • sudo apache2ctl configtest Se reinicia Apache: /etc/init.d/networking apache2

4.2 Práctica1: Ataque de envenenamiento ARP (Spooofing) utilizando Ettercap.

4.2.1 Objetivo General:

Mostrar mediante una topología, el ataque de Envenenamiento de caché utilizando Ettercap hacia los servidores DNS Locales.

4.2.2 Objetivos Específicos:

- Describir las configuraciones realizadas en el atacante para poder lograr el ataque de envenenamiento de caché a los DNS con Ettercap.
- Detallar el tráfico de red en los servidores DNS mediante la herramienta tcpdump.

4.2.3 Introducción:

El envenenamiento de caché DNS es un ataque que consiste en el ingreso de registros engañosos en la memoria caché de un servidor DNS, mediante el envío de peticiones DNS y una inundación de respuestas por parte del atacante. El propósito de este ataque es que el servidor DNS víctima remita a los usuarios legítimos a sitios controlados por el atacante.

Con la elaboración de la Práctica 1, el lector será capaz de comprender la vulnerabilidad de los servidores DNS.

4.2.4 Tiempo estimado para la realización de la práctica:

4 horas.

4.2.5 Requerimientos:

Tabla 15: Requerimientos Hardware y Software en práctica 1

Hardware	Software
Computadora con los siguientes requisitos: ❖ Procesador i core 5 con velocidad de 2.1 ghz Memoria RAM de 8 GB.	**Emulador de redes GNS3 con los siguientes elementos:** ❖ 4 Routers. ❖ 9 Switch. Software de virtualización, Virtual Box; enlazado a GNS3 con los siguientes elementos: ❖ 8 servidores DNS. ❖ 2 servidores Web. ❖ 2 VMs Clientes. ❖ 2 VMs Atacante.

4.2.6 Conocimientos previos:

Prerrequisitos: Haber concluido la practica 0.

Para la correcta realización e implementación de esta práctica es necesario conocer sobre el control y manejo del emulador GNS3, el software Kali Linux, la herramienta Ettercap, conceptos y configuraciones básicas sobre dicha herramienta.

4.2.7 Topología:

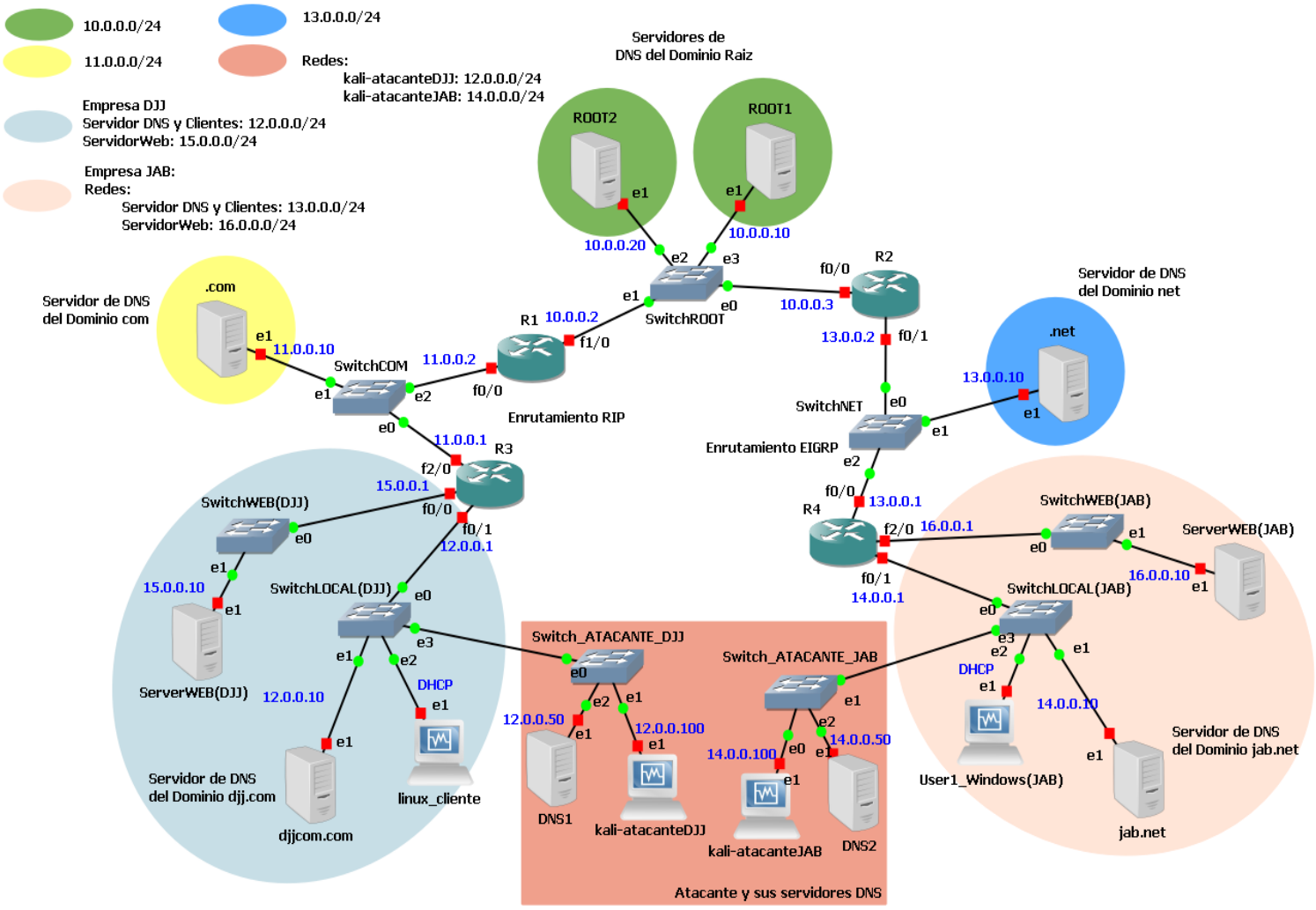


Ilustración 15: Topología de práctica 1

4.2.8 Funcionalidad:

- **General:**

Se logra la ejecución del envenenamiento ARP (Spoofing) al servidor DNS, permitiendo un escenario en donde los usuarios legítimos sean redirigidos hacia otro sitio controlado por el atacante.

- **Herramienta de ataque:**

Ettercap es una herramienta completa para los ataques Man in the Middle; dicha Herramienta ya se encuentra integrada en las distribuciones de Kali Linux, cuenta con la detección de conexiones durante la realización del ataque, filtrado de contenido sobre su ejecución y muchas otras funcionalidades interesantes. Admite la disección activa y pasiva de muchos protocolos e incluye muchas funciones para el análisis de redes y hosts.

4.2.9 Enunciado:

Realizar que la máquina atacante logre interceptar las peticiones proporcionadas por los usuarios victimas pertenecientes a los servidores DNS de dominio .com y .net, para luego redirigirla a una página falsa.

4.2.10 Evaluación del ataque:

El ataque consiste en que la máquina atacante (Kali-atacanteDJJ) se interponga entre la máquina victima (Linux_cliente), donde el atacante puede simplemente captar transmisiones de la víctima, y redigirlas a una página falsa, la cual es controlada por este mismo.

Esta es la página oficial que se les muestra a los hosts clientes, al conectarse al servidor Web www.djj.com, lo cual este es traducido en el servidor DNS djjcom.com.

Ilustración 16: Página web djj.com

4.2.11 Configuración de enrutamiento de la máquina atacante.

Tabla 16: Configuración de enrutamiento en la máquina atacante Kali Linux, empresa DJJ.com

Entidad	Configuración
Máquina Virtual Atacante (Kali-atacanteDJJ)	Identificar las interfaces de red del servidor: • # ip add Configurar la interfaz de red en Kali Linux Abrir el archivo de configuración a modificar, el cual se encuentra en la siguiente ruta: • # nano /etc/network/interfaces Se ingresan las siguientes configuraciones de red: auto eth1 iface eth1 inet static address 12.0.0.100

	network 12.0.0.0 netmask 255.255.255.0 gateway 12.0.0.1 Se reinicia la interfaz: • #/etc/init.d/networking restart

4.2.12 Configuración del servidor DNS atacante.

Tabla 17: Configuración del servidor DNS en la máquina atacante

Entidad	Configuración
Servidor DNS1 Atacante	Identificar las interfaces de red del servidor: • # ip add • Configurar la interfaz de red en CentOS7 Localización del archivo de configuración de la interfaz a modificar en la siguiente ruta: • # cd /etc/sysconfig/network-scripts/ Búsqueda del archivo con el nombre localizado anteriormente, utilizando el comando "ip add", en este caso la interfaz se identifica como enp0s8; y lo harás con el comando "ls": • # ls El archivo ifcfg-enp0s8 es donde se encuentra la configuración de interfaz de red y se edita con el comando "nano" pero con privilegios de root: • # sudo su • # nano ifcfg-enp0s8 Dentro del archivo se realiza lo siguiente: TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=static

	DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=no IPV6_AUTOCONF=no IPV6_DEFROUTE=no IPV6_FAILURE_FATAL=no NAME=enp0s8 UUID=f7... ...229 ONBOOT=yes IPADDR0=12.0.0.50 PREFIX0=24 ZONE=public Se reinician las interfaces: • # systemctl restart network.service Se crea un archivo, el cual contendrá las configuraciones de enrutamiento estático: • # cd /etc/sysconfig/network-scripts/ • # mkdir route-enp0s8 • # nano route-enp0s8 Se ingresan los siguientes contenidos al archivo route-enp0s8: ADDRESS0=10.0.0.0 NETMASK0=255.255.255.0 GATEWAY0=12.0.0.1 ADDRESS1=11.0.0.0 NETMASK1=255.255.255.0 GATEWAY1=12.0.0.1 ADDRESS2=13.0.0.0 NETMASK2=255.255.255.0 GATEWAY2=12.0.0.1 ADDRESS3=14.0.0.0

	NETMASK3=255.255.255.0 GATEWAY3=12.0.0.1 ADDRESS4=15.0.0.0 NETMASK4=255.255.255.0 GATEWAY4=12.0.0.1 ADDRESS5=16.0.0.0 NETMASK5=255.255.255.0 GATEWAY5=12.0.0.1 Se activa el bit de forwarding • nano /etc/sysctl.conf Se cambia la siguiente opción a 1 • net.ipv4.ip_forward = 1

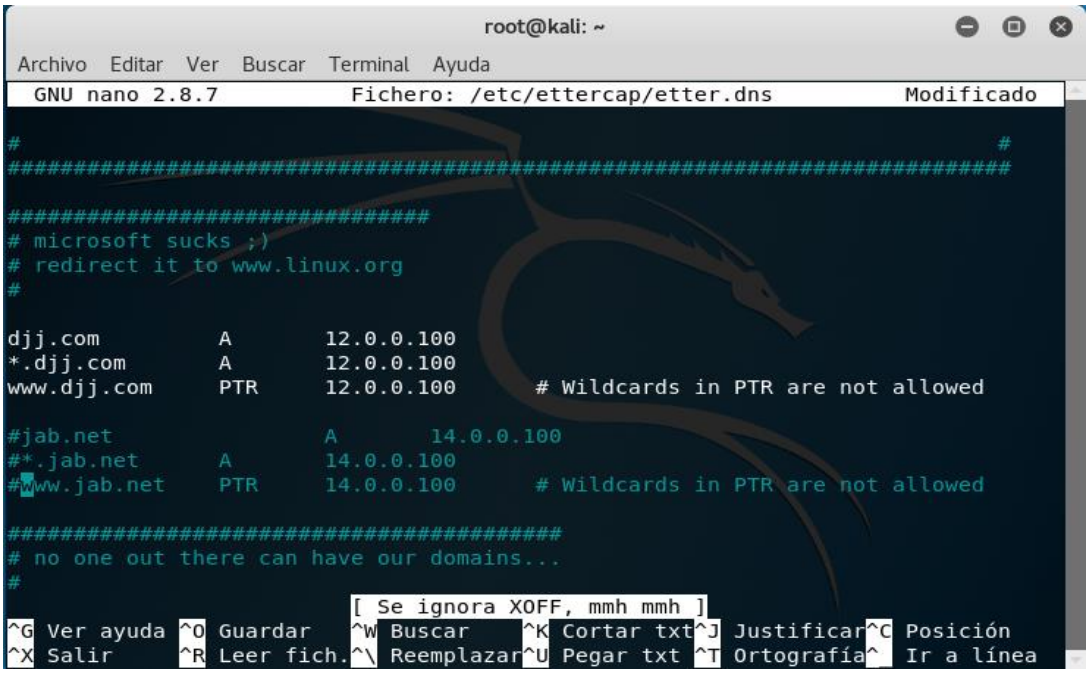
4.2.13 Configuración y realización del ataque de envenenamiento ARP (Spooofing) en la caché del servidor DNS djjcom.com.

Tabla 18: Configuración del ataque Spoofing

Entidad	Configuración

Máquina Virtual Atacante (Kali-atacanteDJJ)	Configuración del archivo etter.dns.
	`root@kali:~# nano /etc/ettercap/`
	`etter.conf  .etter.dns.swp  etter.nbns`
	`etter.dns  etter.mdns`
	`root@kali:~# nano /etc/ettercap/etter.dns`

Se edita el archivo colocando el nombre de dominio que se espera que la víctima utilice (www.djj.com) y la IP del host que contiene el sitio malicioso clonado (12.0.0.100). Se usan las teclas (Ctrl+O) para guardar los cambios del archivo y (Ctrl+x) para salir.



Para iniciar el ataque se debe ejecutar el siguiente comando en la consola:

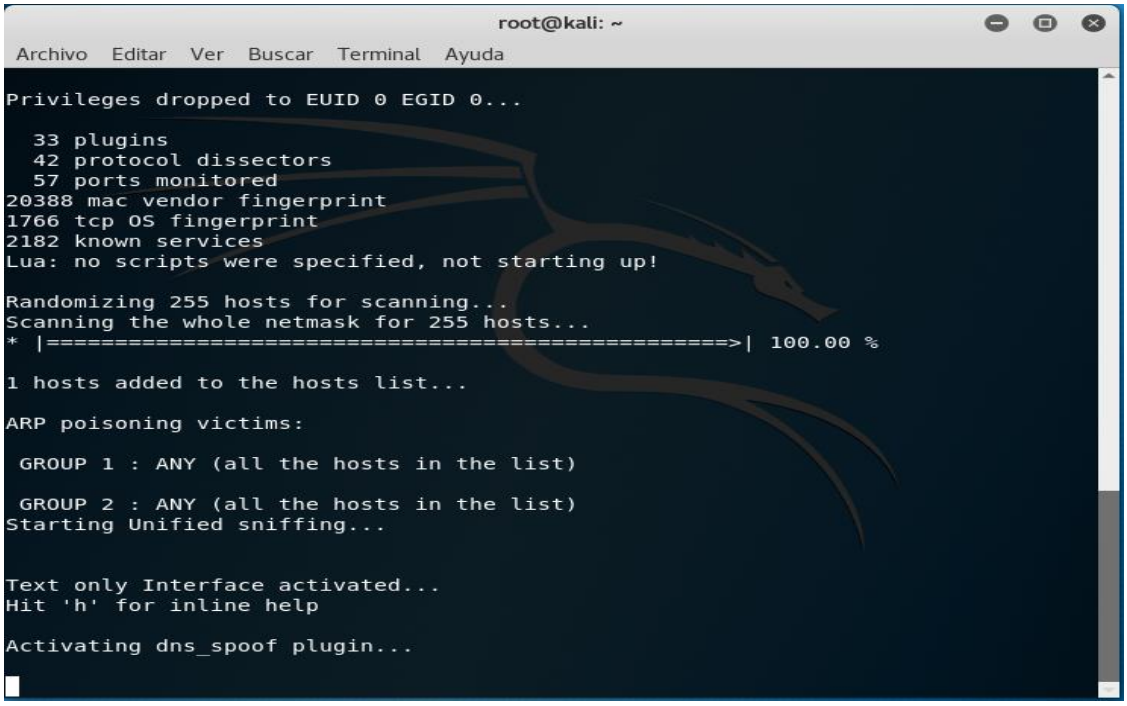
```
root@kali:~# ettercap -i eth1 -T -q -P dns_spoof -M arp:remote
```

Los parámetros de este comando, se explican a continuación:

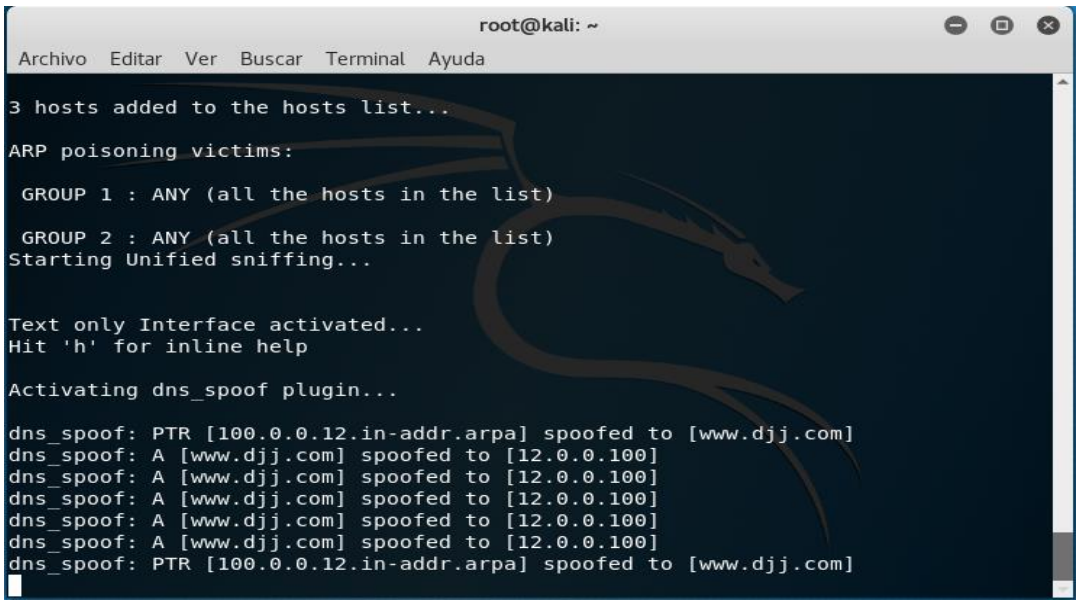
- -i: contiene la interfaz en el cual se encuentra el host a atacar.
- -T: especifica el uso de la interfaz basada texto.
- -q: modo silencioso (no muestra el contenido de los paquetes).
- -P: especifica el uso del complemento dns_spoof.
- -M arp: Inicia un ataque de envenenamiento MITM ARP para interceptar paquetes entre hosts.

Al ejecutar el comando se muestra la siguiente salida:

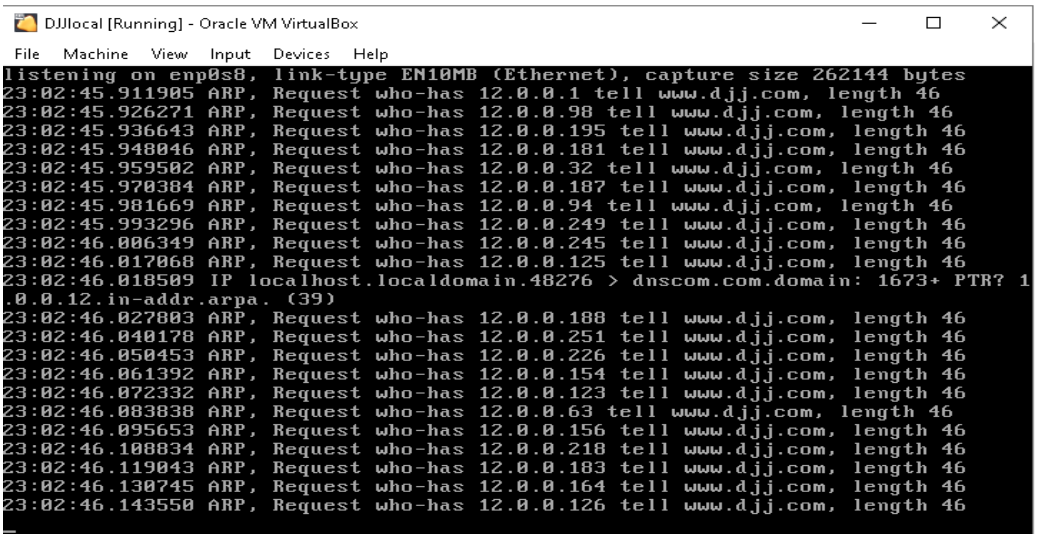
- Nombre y número de la figura: Esperando a la escucha de cualquier host víctima.
- Pruebas y evaluación del ataque de envenenamiento ARP utilizando Ettercap.



Como se observa, dicha herramienta ha capturado un host, el cual es www.djj.com que dicho dominio es traducido en el servidor DNS djjcom.com. Al instante de su captura se realiza el ataque.



Una vez iniciado ettercap, con la herramienta Tcpdump utilizada en el servidor víctima djjcom.com, se observa que este comienza a buscar los servidores DNS autoritativos para el dominio FQDN (Fully Qualified Domain Name) a suplantar. Cuando captura el tráfico del host comienza a enviar peticiones al servidor DNS víctima, para FQDN inexistentes, y a inundarlo de respuestas q no solicito, y suplantarlo a servidores DNS autoritativos.



Cuando el ataque ha concluido con éxito, la victima ingresa al sitio ww.djj.com a través de su navegador de uso; como se observa en la Figura, este es redirigido al sitio controlado por el atacante, lo cual comprueba el éxito del ataque.

4.2.14 Conclusión del ataque.

Por medio del envenenamiento ARP o por la suplantación de caché en un servidor DNS recursivo o la intercepción de una resolución DNS, el atacante puede redirigir al usuario a un sitio web controlado por él; en este sitio el usuario podrá ser víctima de otro tipo de ataques como phishing o inyección de malware.

4.3 Práctica 2: Ataque de Denegación de Servicio utilizando DNSblast.

4.3.1 Objetivo General:

Mostrar mediante una topología, el ataque de Ataque de Denegación de Servicio utilizando DNSblast hacia los servidores DNS Locales.

4.3.2 Objetivos Específicos:

- Describir las configuraciones realizadas en el atacante para poder lograr el ataque al DNS con DNSblast.
- Detallar el tráfico de red en los servidores DNS mediante la herramienta tcpdump.

4.3.3 Introducción:

Un ataque de denegación de servicio es una técnica en la cual el atacante satura los recursos (CPU, RAM, etc.) de un servidor DNS, mediante la inundación de peticiones. De esta forma los recursos del servidor DNS víctima son saturados y se niega el servicio a usuarios legítimos.

Con la elaboración de la Práctica 2, el lector será capaz de comprender la vulnerabilidad de los servidores DNS.

4.3.4 Tiempo estimado para la realización de la práctica:

9 horas.

4.3.5 Requerimientos:

Tabla 19: Requerimientos Hardware y Software en práctica 2

Hardware	Software
Computadora con los siguientes requisitos: ❖ **Procesador i core 5 con velocidad de 2.1 ghz Memoria RAM de 8 GB.**	**Emulador de redes GNS3 con los siguientes elementos:** ❖ 4 Routers. ❖ 9 Switch. Software de virtualización, Virtual Box; enlazado a GNS3 con los siguientes elementos: ❖ 8 servidores DNS. ❖ 2 servidores Web. ❖ 2 VMs Clientes. ❖ 2 VMs Atacante.

4.3.6 Conocimientos previos:

Prerrequisitos: Haber concluido la practica 0.

Para la correcta realización e implementación de esta práctica es necesario conocer sobre el control y manejo del emulador GNS3, el software Kali Linux, la herramienta DNSblast, conceptos y configuraciones básicas sobre dicha herramienta.

4.3.7 Topología:

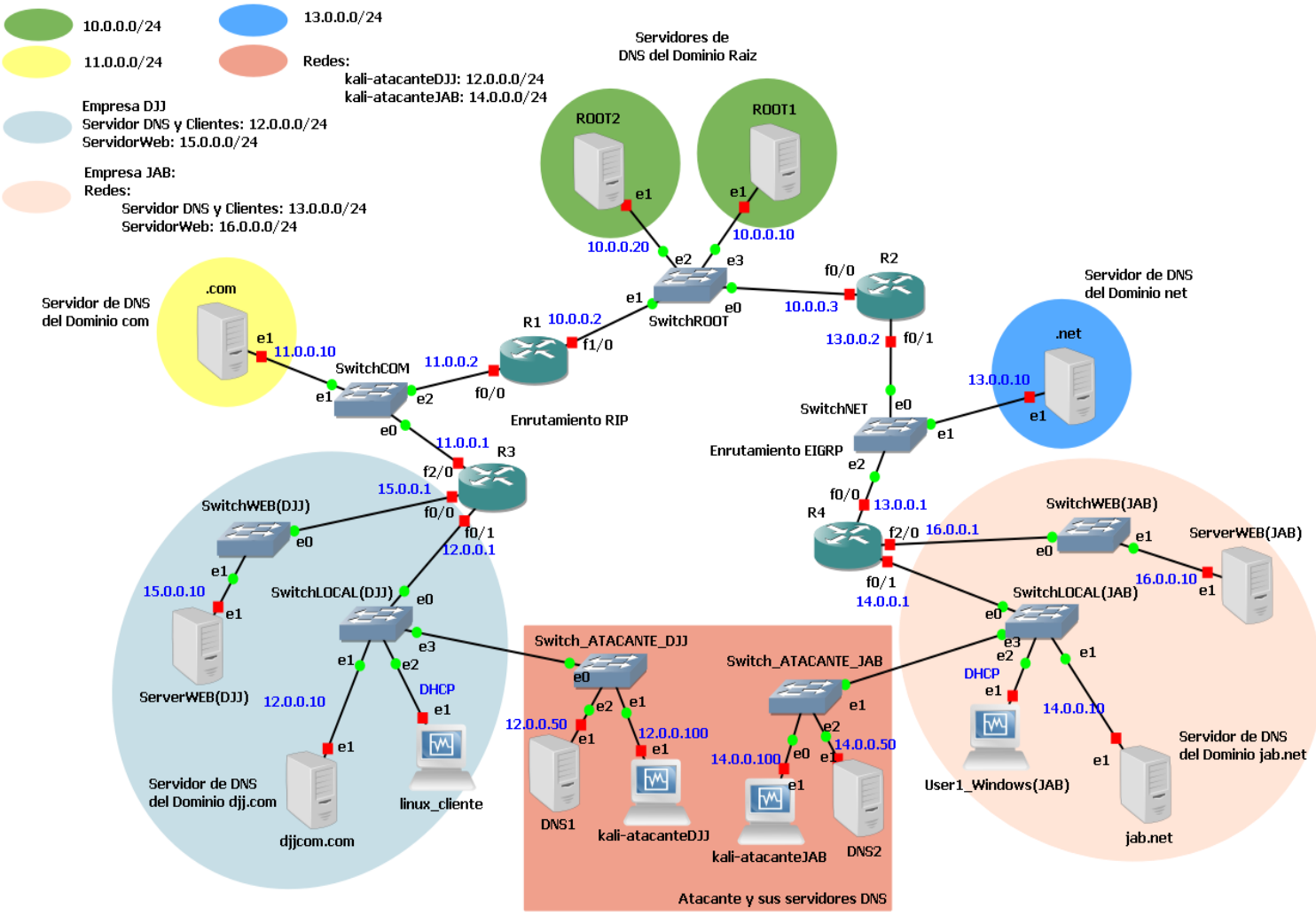


Ilustración 17: Topología de práctica 2

4.3.8 Funcionalidad:

- **General:**

Se logra la ejecución del ataque de denegación de servicio al servidor DNS, permitiendo un escenario en donde a los usuarios legítimos se les niega el servicio.

- **Herramienta de ataque:**

DNSblast es una herramienta que genera una gran cantidad de peticiones DNS que solicitan información de dominios aleatorios; y las dirige a un servidor DNS recursivo. El argumento principal para ejecutar esta herramienta es únicamente la dirección IP del servidor DNS explotado. Algunos argumentos opcionales que podemos ingresar son: número de paquetes a enviarse, paquetes por segundo que se desea enviar y el puerto destino. Además, permite el envío de peticiones DNS malformadas, esto aumenta la carga del servidor. El tipo de registro de recursos solicitado en la petición es aleatorio, puede ser tipo A, SOA, MX, TXT o AAA.

4.3.9 Enunciado:

Realizar que la máquina atacante logre inundar de peticiones al servidor DNS víctima, y así aumentar la carga del servidor, de esta forma lograr la negación del servicio a los usuarios legítimos.

4.3.10 Evaluación del ataque:

El ataque consiste en que la máquina atacante (Kali-atacanteJAB) se interponga entre el servidor DNS (jab.net) y la máquina victima (User1_Windows(JAB)), donde el atacante inunda de peticiones masivas al servidor DNS, y de esta manera se niega el servicio a los usuarios.

Esta es la página oficial que se les muestra a los hosts clientes, al conectarse al servidor Web www.jab.net, lo cual este es traducido en el servidor DNS jabnet.net.

Ilustración 18: Página Web jab.com

4.3.11 Configuración y realización del ataque de denegación de servicio en el servidor DNS jab.net.

Tabla 20: Configuración de ataque de denegación de servicio

Entidad	Configuración
Máquina Virtual Atacante (Kali-atacanteJAB)	Para la implementación del ataque DoS, se ejecuta la herramienta DNSblast desde la máquina atacante (Kali-atacanteJAB). Se despliega en consola la información sobre el ataque: paquetes enviados, paquetes recibidos por el servidor DNS víctima (jab.net).

Se genera peticiones DNS con DNSblast.

- /dnsblast <dir. ip de la víctima> <números de paquetes a enviar> <número de paquetes por segundos> <número de puerto>

```
root@kali:~/dnsblast# ./dnsblast 14.0.0.10 50000000 1000 53
Sent: [255472] - Received: [219780] - Reply rate: [863 pps] - Ratio: [86.03%]
```

La dirección IP 14.10.10.10 pertenece al servidor DNS víctima, 50000000 es el número de paquetes en total a enviar masivamente para saturar de peticiones al servidor DNS víctima, 1000 el número de paquete por segundos enviados y 53 el puerto destino de los paquetes. En el caso de no definir el puerto destino, por defecto se enviarán los paquetes al puerto 53.

Una vez iniciado el ataque se envía masivamente las peticiones al DNS, de esta forma los paquetes incrementan la carga de la CPU del servidor víctima. Además, su memoria caché comienza a llenarse de respuestas, lo que significa que su memoria RAM empieza a saturarse.

Captura de tráfico con Tcpdump durante ataque DoS (servidor DNS víctima).

- Tcpdump -i enp0s8

```
kt.com. (26)
21:23:09.569398 IP 14.0.0.1 > localhost.localdomain: ICMP host 10.0.0.10 unreach
able, length 36
21:23:09.578001 IP 14.0.0.100.52150 > localhost.localdomain.domain: 5313+ A? 1d8
5.com. (26)
21:23:09.596019 IP 14.0.0.100.52150 > localhost.localdomain.domain: 11969+ SOA?
3cs6.com. (26)
21:23:09.603367 IP 14.0.0.1 > localhost.localdomain: ICMP host 10.0.0.10 unreach
able, length 36
21:23:09.622289 IP 14.0.0.100.52150 > localhost.localdomain.domain: 20929+ A? 88
17.com. (26)
21:23:09.639340 IP 14.0.0.1 > localhost.localdomain: ICMP host 10.0.0.10 unreach
able, length 36
21:23:09.645381 IP 14.0.0.100.52150 > localhost.localdomain.domain: 24257+ A? 66
7z.com. (26)
21:23:09.659782 IP 14.0.0.100.52150 > localhost.localdomain.domain: 31425+ A? 42
mf.com. (26)
21:23:09.664679 IP 14.0.0.1 > localhost.localdomain: ICMP host 10.0.0.10 unreach
able, length 36
21:23:09.685692 IP 14.0.0.100.52150 > localhost.localdomain.domain: 38849+ A? 65
oi.com. (26)
21:23:09.691933 IP 14.0.0.100.52150 > localhost.localdomain.domain: 41665+ A? i7
11.com. (26)
21:23:09.707311 IP 14.0.0.1 > localhost.localdomain: ICMP host 10.0.0.10 unreach
able, length 36_
```

Con el fin de verificar la ejecución del ataque, se captura el tráfico entrante generado en la interfaz de red del servidor DNS víctima, utilizando la herramienta Tcpdump. Como se observa, los paquetes DNS generados por el atacante solicitan información de nombres de dominio inexistentes.

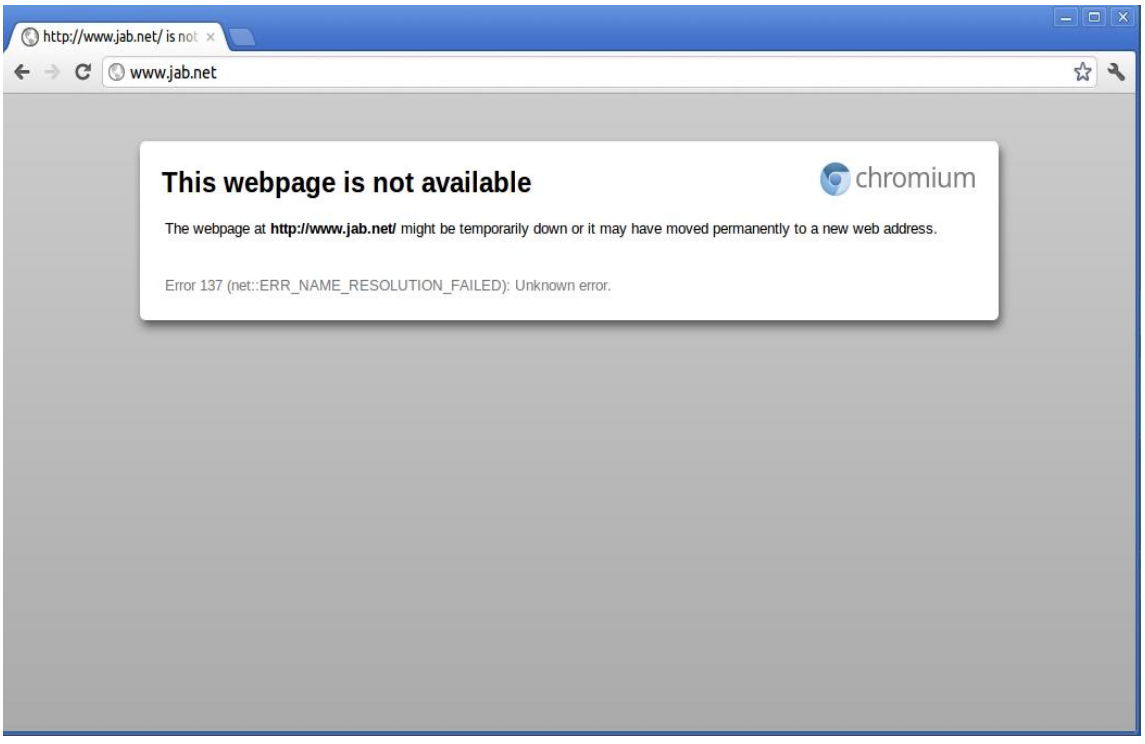
Consulta realizada al servidor DNS víctima, desde la maquina cliente.

- Nslookup jab.net

```
root@serverwebJAB:/home/serverweb# nslookup jab.net
;; connection timed out; no servers could be reached
```

Para comprobar la efectividad del ataque, se utiliza la herramienta nslookup desde la maquina cliente. Como se observa se hace una consulta DNS al servidor DNS victima preguntando sobre el dominio jab.net. Como se ilustra no se tiene respuesta del servidor.

Luego se ingresa a un navegador web desde la máquina cliente, y después se intenta acceder al sitio ww.jab.net, y a como se observa no se puede acceder al sitio.



4.3.12 Conclusión del ataque.

Por medio del ataque de denegación de servicios en un servidor DNS recursivo, otros servicios son afectados tales como el servicio web, de correo, etc; debido a que no se puede obtener las direcciones IP asociadas a los FQDNs de los hosts/recursos solicitados.

4.4 Practica 3: Implementación de las extensiones DNSSEC.

4.4.1 Objetivo General:

Dar a conocer mediante una topología, la implementación de las extensiones DNSSEC en los servidores DNS.

4.4.2 Objetivo Específico:

Especificar las configuraciones realizadas en los servidores para lograr la seguridad necesaria en los servidores DNS.

4.4.3 Introducción:

Las extensiones de seguridad DNS (DNSSEC) son una especificación que tiene como objetivo mantener la integridad de los datos de las respuestas DNS. DNSSEC firma todos los registros de recursos DNS (A, MX, CNAME, etc.) de una zona utilizando PKI (Public Key Infrastructure). Ahora los solucionadores de DNS habilitados para DNSSEC (como Google Public DNS) pueden verificar la autenticidad de una respuesta DNS (que contiene una dirección IP) utilizando el registro DNSKEY público.

Con la elaboración de la práctica 3, el lector será capaz de comprender como se implementa mediante la configuración, la seguridad e integridad a los servidores DNS.

4.4.4 Tiempo estimado para la realización de la práctica:

4 horas.

4.4.5 Requerimientos:

Tabla 21: Requerimientos Hardware y software en práctica 3

Hardware	Software
Computadora con los siguientes requisitos: ❖ **Procesador i core 5 con velocidad de 2.1 ghz Memoria RAM de 8 GB.**	**Emulador de redes GNS3 con los siguientes elementos:** ❖ 4 Routers. ❖ 9 Switch. Software de virtualización, Virtual Box; enlazado a GNS3 con los siguientes elementos: ❖ 8 servidores DNS. ❖ 2 servidores Web. ❖ 2 VMs Clientes. ❖ 2 VMs Atacante.

4.4.6 Conocimientos previos:

Prerrequisitos: Haber concluido la practica 1 y 2.

Para la correcta realización e implementación de esta práctica es necesario conocer sobre el control y manejo del emulador GNS3, las extensiones DNSSEC, conceptos y configuraciones sobre dichas extensiones.

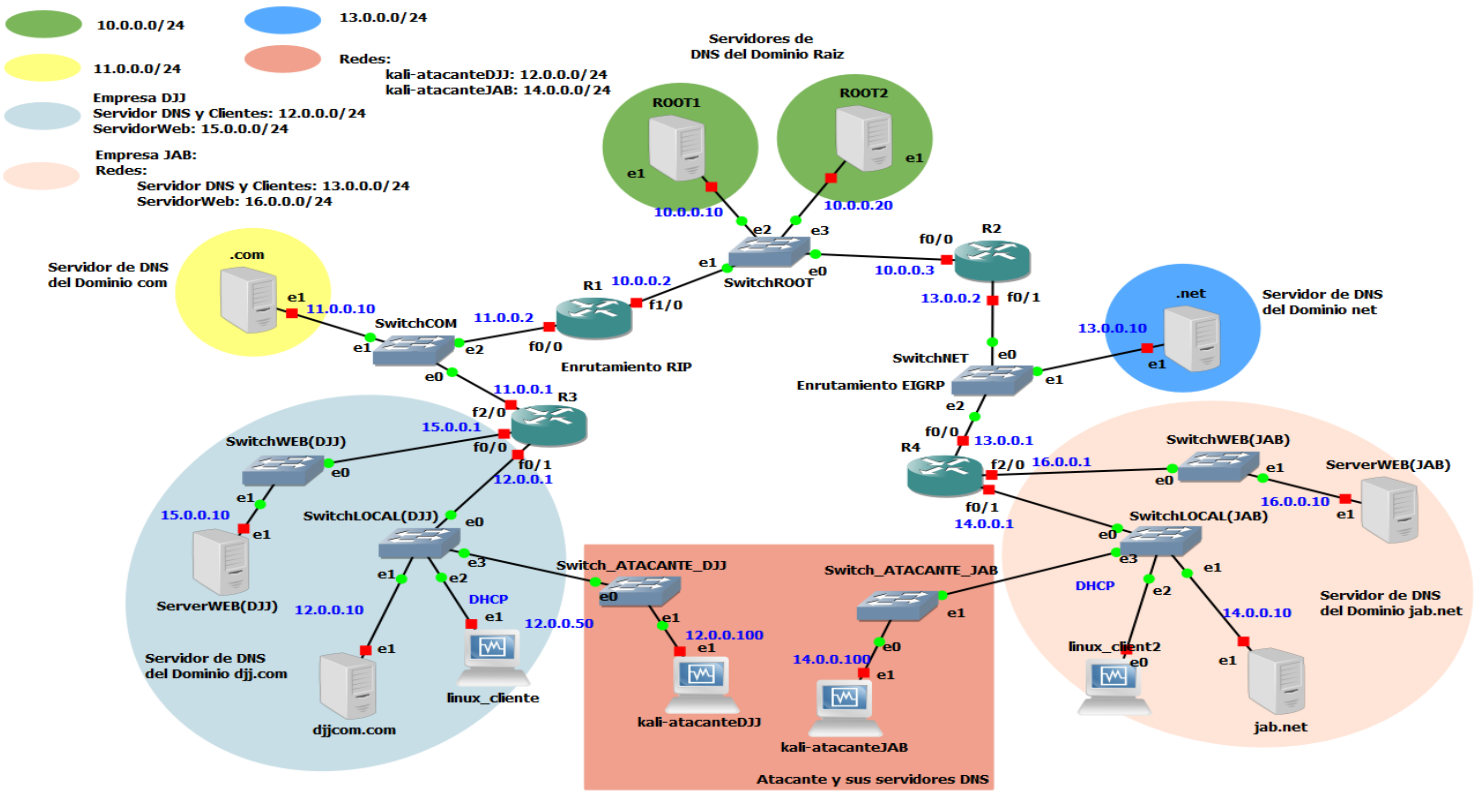


Ilustración 19: Topología de práctica 3

4.4.7 Funcionalidad:

- **General:**

Se logra la implementación de las extensiones DNSSEC de los servidores DNS, permitiendo un escenario en donde los usuarios legítimos tengan un servicio o sitio web seguro e íntegro.

- **Herramienta de ataque:**

El protocolo DNSSEC es una extensión del propio protocolo DNS aumentando su seguridad. Gracias a DNSSEC los clientes podrán obtener autenticación del origen de datos DNS, además también permite la integridad de estos datos haciendo que no se pueda modificar sin que lo sepamos. Todas las respuestas en DNSSEC son firmadas digitalmente, y el cliente DNS comprueba la firma digital para saber si la información recibida es idéntica a la información de los servidores DNS autorizados. Lo que no asegura DNSSEC es la disponibilidad ni tampoco la confidencialidad (las peticiones también se realizan sin cifrar como en DNS). Este protocolo fue diseñado para evitar

ataques típicos como el envenenamiento de caché DNS y ataques de denegación de servicio ya que limita el número de peticiones máximo.

4.4.8 Enunciado:

Realizar todas las respectivas configuraciones en todos los servidores DNS, para su correspondiente seguridad e integridad, etc.

4.4.9 Evaluación de la implementación de las extensiones DNSSEC:

Dicha implementación consiste en dar soporte a todos los servidores DNS, para lograr que las zonas de cada uno de dichos servidores estén cifradas, y así brinden un sitio seguro y estable a los usuarios finales.

4.4.10 Implementación de las extensiones DNSSEC en todos los Servidores DNS.

Tabla 22: Implementación de las extensiones DNSSEC en el servidor Root

Entidad	Configuración

Servidor DNS Root1	Principales opciones que controlan el soporte DNSSEC en Bind (en Centos 7 se encuentra en named.conf):
	`dnssec-enable yes;`
	`dnssec-validation yes;`
	`dnssec-lookaside auto;`
	• dnssec-enable, indica soporte DNSSEC como servidor autoritativo de un dominio (en los DNS primario/secundario).
	• dnssec-validation, que activará la validación de respuestas DNSSEC como servidor recursivo (en el DNS que usan nuestras máquinas como resolver).
	• (Opcional) dnssec-lookaside, se usa para configurar la Validación dinámica de Lookaside (DLV).

Ejecutar el siguiente comando:

```
[root@localhost Keys]#
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE arpa
Generating key pair.....................................++++++ .................
.....++++++
Karpa.+008+19341
[root@localhost Keys]# _
```

Este comando crea dos ficheros (un **.private** y un **.key**) del tipo **Karpa.+008+19341**. Donde el fichero .**private** contiene la clave privada y el **.key** la pública. El **+008** indica el algoritmo elegido **(-a RSASHA256)** y el +**19341** el identificador de la clave (lo usaremos luego en el registrador). Con -b 1024 elegimos el tamaño de la clave a generar**. -n ZONE** indica que la clave será usada para trabajar con zonas DNSSEC, **arpa** es el nombre de la zona.

Luego se procede a ejecutar lo siguiente:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK arpa
Generating key pair.....................................................+++ .........
.....................................................+++
Karpa.+008+03799
[root@localhost Keys]#
```

Una vez más se obtienen dos ficheros (un **.private** y un **.key**) del tipo **Karpa.+008+03799**. Donde, **008** es el algoritmo y **03799** el identificador/etiqueta de la clave. Con -b 2048 elegimos el tamaño de la clave a generar, 2048 es una buena opción para la KSK. -f KSK indica el tipo de clave.

Se realizan los mismos pasos para la zona inversa:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE 10.in-addr.arp
a
Generating key pair....++++++ ...................++++++
K10.in-addr.arpa.+008+42633
[root@localhost Keys]# _
```

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK 10.in-a
ddr.arpa
Generating key pair.....................................................+++ ........+
++
K10.in-addr.arpa.+008+62491
[root@localhost Keys]# _
```

Se transfieren todas las llaves públicas al archivo principal de configuración DNS para luego ser utilizadas.

```
[root@localhost Keys]# cat Karpa.+008+*.key >> db.arpa_
```

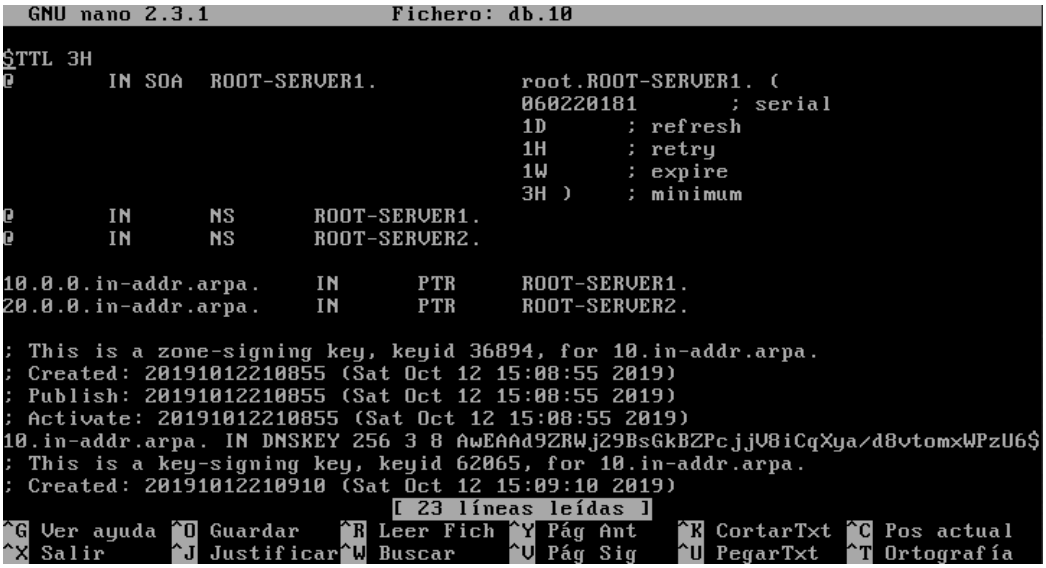
Acá se puede apreciar en el archivo db.arpa, los datos transferidos desde los archivos .key, previamente.

```
  GNU nano 2.3.1              Fichero: db.arpa

in-addr.arpa.    IN NS    ROOT-SERVER2.
com      IN      A        11.0.0.10
djj.com  IN      A        12.0.0.10
net      IN      A        13.0.0.10
jab.net  IN      A        14.0.0.10

; This is a key-signing key, keyid 3865, for arpa.
; Created: 20191012210627 (Sat Oct 12 15:06:27 2019)
; Publish: 20191012210627 (Sat Oct 12 15:06:27 2019)
; Activate: 20191012210627 (Sat Oct 12 15:06:27 2019)
arpa. IN DNSKEY 257 3 8 AwEAAbOXVVfOso8sjspzKkK+Wir97j2Foz9FnefuynXwhbR2O9iIzk4$
; This is a zone-signing key, keyid 24666, for arpa.
; Created: 20191012210554 (Sat Oct 12 15:05:54 2019)
; Publish: 20191012210554 (Sat Oct 12 15:05:54 2019)
; Activate: 20191012210554 (Sat Oct 12 15:05:54 2019)
arpa. IN DNSKEY 256 3 8 AwEAAdTd4M9cKb4VKL+sp9wQ+shHP82gjz9yN1Yh+Wsmbb35cW5YLAN$



^G Ver ayuda ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir      ^J Justificar^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía
```

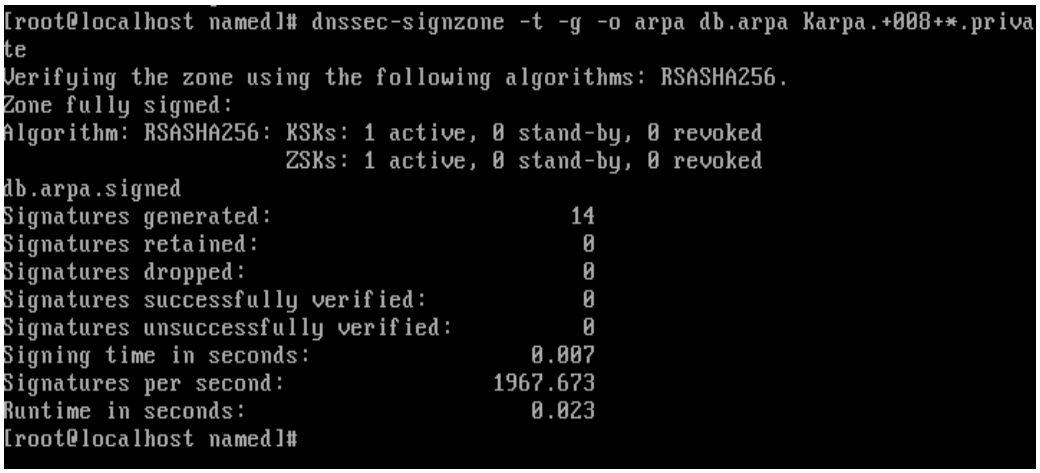
Se realizan los mismos pasos para la zona inversa:

```
[root@localhost named]# cat K10.in-addr.arpa.+008+*.key >> db.10
```

Así como en el archivo anterior, se pueden apreciar los datos transferidos en el archivo db.10 (de la zona inversa) desde los archivos .key, previamente.

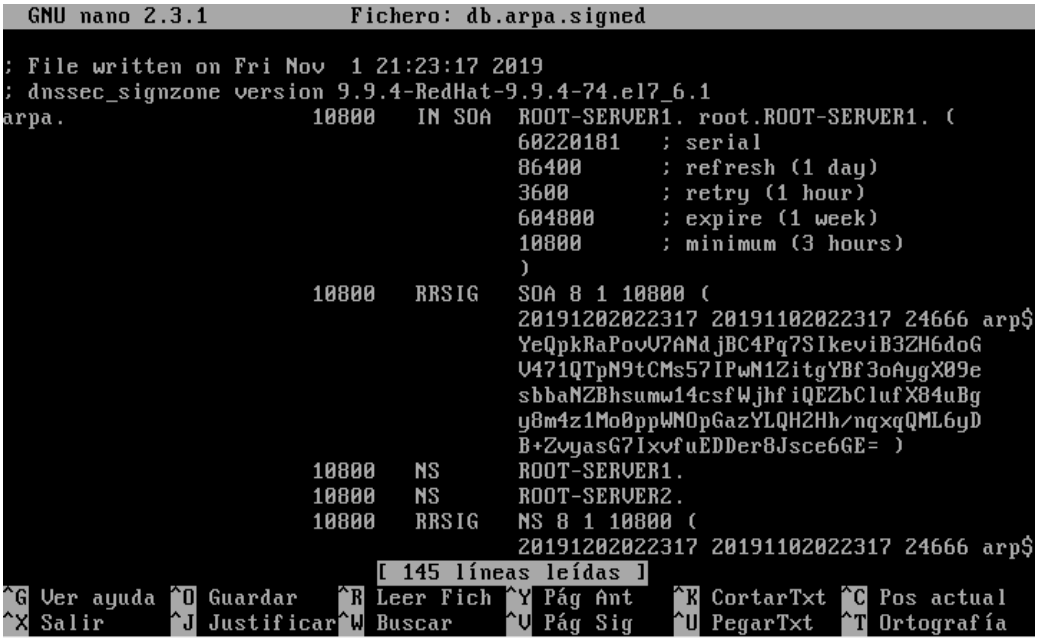


Comando de ejecución para el firmado de la zona:



- **dnssec-sigzone**. Herramienta para el firmado de zona.
- **-t.** TTL
- **-g**.
- **-o arpa.** Indica el ORIGIN de la zona, es decir el dominio.
- db.arpa. Archivo de configuración de la zona.
- **Karpa.+008+*.private.** Llaves privadas.

Archivo generado para la zona principal, al ejecutar el comando anterior.



Comando de ejecución para el firmado de la zona inversa:

```
[root@localhost named]# dnssec-signzone -t -g -o 10.in-addr.arpa db.10 K10.in-ad
dr.arpa.+008+*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.10.signed
Signatures generated:                       5
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.003
Signatures per second:               1470.588
Runtime in seconds:                     0.019
[root@localhost named]# _
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t.** TTL
- **-g**
- **-o 10.in-addr.arpa.** Indica el *ORIGIN* de la zona, es decir el dominio.
- db.10. Archivo de configuración de la zona.
- **K10.in-addr.arpa.+008+*.private.** Llaves privadas.

Archivo generado para la zona inversa ya cifrada, al ejecutar el comando anterior:

```
  GNU nano 2.3.1          Fichero: db.10.signed

; File written on Fri Nov  1 21:25:33 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-74.el7_6.1
10.in-addr.arpa.        10800   IN SOA  ROOT-SERVER1. root.ROOT-SERVER1. (
                                        60220181   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 3 10800 (
                                        20191202022533 20191102022533 36894 10.$
                                        eERchiwjmd/bIPCDvnEaQWRhOjBqlSzLf11K
                                        kzM15uKSAmzEuyk6FFqXhElyLVJR2nWbVIFp
                                        g9x/VWQHDhKvcULR8I5kSCMra/uh3HukwRjj
                                        FxYZcFMTY5qtydnciaoPulxra4AwKraT95nT
                                        3trBS3WK+FfAUQFVSOs4JeM+hek= )
                        10800   NS      ROOT-SERVER1.
                        10800   NS      ROOT-SERVER2.
                        10800   RRSIG   NS 8 3 10800 (
                                        20191202022533 20191102022533 36894 10.$
                                [ 73 líneas leídas ]
^G Ver ayuda ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir     ^J Justificar^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía
```

Una vez que se tiene la zona firmada, lo que queda es cambiar el archivo de zona actual por el generado anteriormente (“db.nombre_de_zona.signed”), tanto para la zona principal como para la zona inversa.

```
  GNU nano 2.3.1          Fichero: /etc/named.conf

        file "named.ca";
};

zone "arpa" IN {
        type master;
        file "db.arpa.signed";
        allow-update { none; };
};

zone "in-addr.arpa" IN {
        type master;
        file "db.arpa.in-addr";
        allow-update { none; };
};

zone "10.in-addr.arpa" IN {
        type master;
        file "db.10.signed";
        allow-update { none; };
};
```

Se reinicia el servicio de named:

```
[root@localhost named]# systemctl restart named
[root@localhost named]# _
```

Se ejecuta lo siguiente para verificar el registro DNSKEY:

```
[root@localhost named]# dig DNSKEY @10.0.0.10 arpa. +multiline | more_
```

- dig es una herramienta de línea de comando para consultar a un servidor de nombres por registros DNS.

Cuando se ha ejecutado el comando anterior la salida es:

```
; <<>> DiG 9.9.4-RedHat-9.9.4-74.el7_6.1 <<>> DNSKEY @10.0.0.10 arpa. +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4394
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;arpa.                  IN DNSKEY

;; ANSWER SECTION:
arpa.                  10800 IN DNSKEY 257 3 8 (
                              AwEAAbOXVVfOso8sjspzKkK+Wir97j2Foz9FnefuynXw
                              hbR2O9iIzk4jGkbdgd+zL0+bpeeHIWVQmSXHSwkq1Seg
                              dYuw3thQml6nEePA603mlwsC7DYx4MOLrHfR9Tasts2l
                              L1deGnWqZJDA5g+gepZF8+Y8QT3wpzApqVJDK+iTj3Fq
                              6nkQSxYZKs9Gd8RbOA49ReRkZw8FzvX/DTOiZo3/PgsJ
                              7PQDIirCxeR3A7dHM7+fvfXAaNjlxBPEED4tVcCte9/1
                              AxcZsDynOnoqpCTqkJM0z4qu0cuRNOsbW8hoi/BVW2l2
                              q4G270150kKf/3HF+GI4kFadTFOYma2dL/6cz6M=
                              ) ; KSK; alg = RSASHA256; key id = 3865
--Más--
```

Tabla 23: Implementación de las extensiones DNSSEC en el servidor com

Entidad	Configuración

Servidor DNS com	Principales opciones que controlan el soporte DNSSEC en Bind (en Centos 7 se encuentra en named.conf):
	
	`dnssec-enable yes;`
	`dnssec-validation yes;`
	`dnssec-lookaside auto;`
	
	• dnssec-enable, indica soporte DNSSEC como servidor autoritativo de un dominio (en los DNS primario/secundario).
	• dnssec-validation, que activará la validación de respuestas DNSSEC como servidor recursivo (en el DNS que usan nuestras máquinas como resolver).
	• (Opcional) dnssec-lookaside, se usa para configurar la Validación dinámica de Lookaside (DLV).

Ejecutar el siguiente comando:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE com
Generating key pair.............++++++ .....................++++++
Kcom.+008+24066
[root@localhost Keys]#
```

Este comando crea dos ficheros (un **.private** y un **.key**) del tipo **Kcom.+008+24066**. Donde el fichero .**private** contiene la clave privada y el .key la pública. El **+008** indica el algoritmo elegido **(-a RSASHA256)** y el +**24066** el identificador de la clave (lo usaremos luego en el registrador). Con -b 1024 elegimos el tamaño de la clave a generar. **-n ZONE** indica que la clave será usada para trabajar con zonas *DNSSEC*. **com** es el nombre de la zona.

Luego se procede a ejecutar lo siguiente:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK com
Generating key pair.........................+++ ..............................
.................+++
Kcom.+008+36829
[root@localhost Keys]#
```

Una vez más se obtienen dos ficheros (un **.private** y un **.key**) del tipo **Kcom.+008+36829**. Donde, **008** es el algoritmo y **36829** el identificador/etiqueta de la clave. Con -b 2048 elegimos el tamaño de la clave a generar, 2048 es una buena opción para la *KSK*. -f KSK indica el tipo de clave.

Se realizan los mismos pasos para la zona inversa:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE 11.in-addr.arp
a
Generating key pair.........................++++++ ..........................
........++++++
K11.in-addr.arpa.+008+10209
[root@localhost Keys]#
```

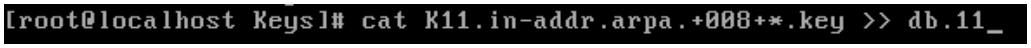
```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK 11.in-a
ddr.arpa
Generating key pair.........................................................
.........................................+++ ..........................+++
K11.in-addr.arpa.+008+29872
[root@localhost Keys]# _
```

Se transfieren todas las llaves públicas al archivo principal de configuración DNS para luego ser utilizadas:

```
[root@localhost Keys]# cat Kcom.+008+*.key >> db.com_
```

Acá se puede apreciar en el archivo db.com, los datos transferidos desde los archivos .key, previamente.

Se realizan los mismos pasos para la zona inversa:



Así como en el archivo anterior, se pueden apreciar los datos transferidos en el archivo db.11 (de la zona inversa) desde los archivos .key, previamente.

Comando de ejecución para el firmado de zona:

```
[root@localhost named]# dnssec-signzone -t -g -o com db.com Kcom.+008+*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.com.signed
Signatures generated:                       8
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.005
Signatures per second:               1540.238
Runtime in seconds:                     0.023
[root@localhost named]# _
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t.** TTL
- **-g.**
- **-o com.** Indica el ORIGIN de la zona, es decir el dominio.
- **db.com**. Archivo de configuración de la zona.
- **Kcom.+008+*.private.** Llaves privadas.

Archivo generado para la zona principal, al ejecutar el comando anterior.

```
  GNU nano 2.3.1          Fichero: db.com.signed

; File written on Fri Nov  1 22:05:38 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-74.el7_6.1
com.                    10800   IN SOA  dnscom.com. root.dnscom.com. (
                                        60220181   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 1 10800 (
                                        20191202030538 20191102030538 53160 com.
                                        GEdTehCwsdTckL1So3G6uqX7093+ZdRf2kHk
                                        23gOBHZhD+rpYVosGIpyZVAbdZoEDFbVyTcV
                                        FYs+91TLaq8v1z4vBOl6OnPg0v3JZjB7dnDC
                                        sWN8ZOsxf/HyGPpJW5oKAtBbBFmtADTsgOxH
                                        kIQ+gaOwkLFB/6usjmfygxd5nBI= )
                        10800   NS      dnscom.com.
                        10800   RRSIG   NS 8 1 10800 (
                                        20191202030538 20191102030538 53160 com.
                                        fBVcldyUpX0CTyyTbmj31yz+USkrM+xeDWwO
                              [ 97 líneas leídas ]
```

Comando de ejecución para el firmado de la zona inversa:

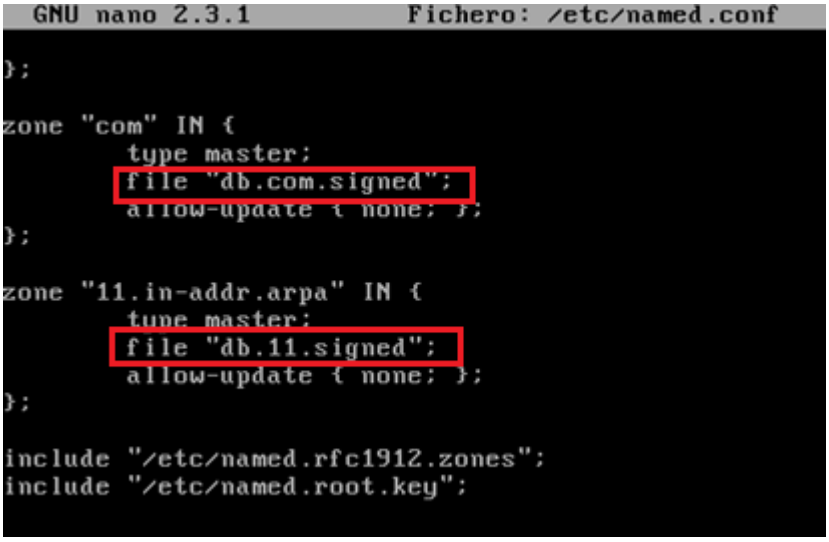
```
[root@localhost named]# dnssec-signzone -t -g -o 12.in-addr.arpa db.12 K12.in-ad
dr.arpa.+008+0*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.12.signed
Signatures generated:                       7
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.004
Signatures per second:               1519.426
Runtime in seconds:                     0.024
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t.** TTL
- **-g**
- **-o 11.in-addr.arpa.** Indica el *ORIGIN* de la zona, es decir el dominio.
- **db.11**. Archivo de configuración de la zona.
- **K11.in-addr.arpa.+008+*.private.** Llaves privadas.

Archivo generado para la zona inversa ya cifrada, al ejecutar el comando anterior.

```
 GNU nano 2.3.1          Fichero: db.11.signed

; File written on Fri Nov  1 22:06:39 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-74.el7_6.1
11.in-addr.arpa.        10800   IN SOA  dnscom.com. root.dnscom.com. (
                                        60220181   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 3 10800 (
                                        20191202030639 20191102030639 64080 11.$
                                        Ilk/QLhLB0miYz+hN/8mTA1Kb5qR68FRmso4
                                        YbEQICc2uoenvZov5nFsftCfYw2IRSXlc8qK
                                        3bx8p9Q12bjuJbQdELBTJIRZIlR6mRSZnBKY
                                        1sM9qxFJcBZLDQBJPQKV4eGP3XOf jHxCRtg3
                                        04TAv0Mu3fDhpHNC5mMCy2U5kYQ= )
                        10800   NS      dnscom.com.
                        10800   RRSIG   NS 8 3 10800 (
                                        20191202030639 20191102030639 64080 11.$
                                        JWIE6/xx+37BgbKPsR8649doRdTuLUfvu5tt
```

Una vez que se tiene la zona firmada, lo que queda es cambiar el archivo de zona actual por el generado anteriormente (“db.nombre_de_zona.signed”), tanto para la zona principal como para la zona inversa.



Se reinicia el servicio de named:

```
[root@localhost named]# systemctl restart named
[root@localhost named]# _
```

Se ejecuta lo siguiente para verificar el registro DNSKEY:

```
[root@localhost named]# dig DNSKEY @11.0.0.10 com. +multiline | more
```

- dig es una herramienta de línea de comando para consultar a un servidor de nombres por registros DNS.

Cuando se ha ejecutado el comando anterior, la salida es:

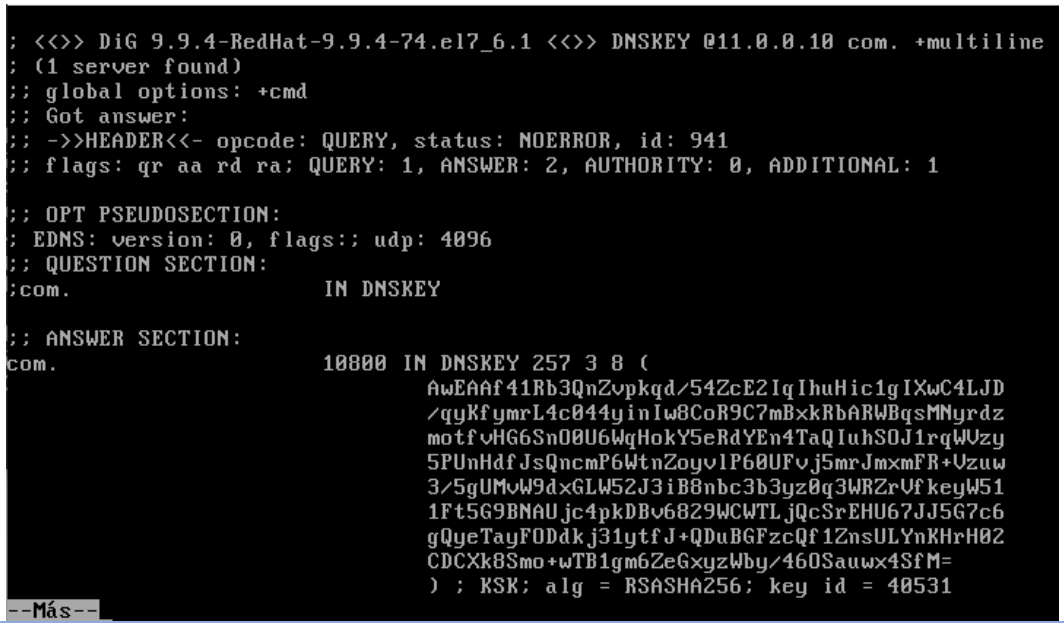


Tabla 24: Implementación de las extensiones DNSSEC en el servidor net

Entidad	Configuración

Servidor DNS net	Principales opciones que controlan el soporte DNSSEC en Bind (en Centos 7 se encuentra en named.conf):
	
	`dnssec-enable yes;`
	`dnssec-validation yes;`
	`dnssec-lookaside auto;`
	
	• dnssec-enable, indica soporte DNSSEC como servidor autoritativo de un dominio (en los DNS primario/secundario).
	• dnssec-validation, que activará la validación de respuestas DNSSEC como servidor recursivo (en el DNS que usan nuestras máquinas como resolver).
	• (Opcional) dnssec-lookaside, se usa para configurar la Validación dinámica de Lookaside (DLV).

Ejecutar el siguiente comando:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE net
Generating key pair.....................................++++++ .....++++++
Knet.+008+06960
[root@localhost Keys]# _
```

Este comando crea dos ficheros (un **.private** y un **.key**) del tipo **Knet.+008+06960**. Donde el fichero .**private** contiene la clave privada y el .key la pública. El +**008** indica el algoritmo elegido **(-a RSASHA256)** y el +**06960** el identificador de la clave (lo usaremos luego en el registrador). Con **-b 1024** elegimos el tamaño de la clave a generar**. -n ZONE** indica que la clave será usada para trabajar con zonas DNSSEC, **net** es el nombre de la zona.

Luego se procede a ejecutar lo siguiente:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK net
Generating key pair.................................+++ ..................
........+++
Knet.+008+31717
[root@localhost Keys]#
```

Una vez más se obtienen dos ficheros (un **.private** y un **.key**) del tipo **Knet.+008+31717**. Donde, **008** es el algoritmo y **31717** el identificador/etiqueta de la clave. Con **-b 2048** elegimos el tamaño de la clave a generar, **2048** es una buena opción para la KSK. **-f KSK** indica el tipo de clave.

Se realizan los mismos pasos para la zona inversa:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE 13.in-addr.arp
a
Generating key pair.........++++++ ......................................+++++
+
K13.in-addr.arpa.+008+07554
[root@localhost Keys]# _
```

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK 13.in-a
ddr.arpa
Generating key pair.....................................................
.....................................................................
.....................................+++ ...........................
.................+++
K13.in-addr.arpa.+008+39994
[root@localhost Keys]# _
```

Se transfieren todas las llaves públicas al archivo principal de configuración DNS para luego ser utilizadas.

```
[root@localhost Keys]# cat Knet.+008+*.key >> db.net
```

Acá se puede apreciar en el archivo db.net, los datos transferidos desde los archivos .key, previamente.

```
  GNU nano 2.3.1              Fichero: db.net

$TTL 3H
@       IN SOA  dnsnet.net.             root.dnsnet.net. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      dnsnet.net.
dnsnet.net.     IN      A       13.0.0.10

jab.net.        IN      NS      www.jab.net.
jab.net.        IN      A       14.0.0.10
www.jab.net.    IN      A       16.0.0.10
; This is a zone-signing key, keyid 21990, for net.
; Created: 20191014022745 (Sun Oct 13 20:27:45 2019)
; Publish: 20191014022745 (Sun Oct 13 20:27:45 2019)
; Activate: 20191014022745 (Sun Oct 13 20:27:45 2019)
net. IN DNSKEY 256 3 8 AwEAAePM1r/2zwCAJX9PBxvxWGcuSwhqYpuTKBXIP81Z797rjEkg+Bw/$
; This is a key-signing key, keyid 49660, for net.
; Created: 20191014022757 (Sun Oct 13 20:27:57 2019)
                              [ 33 líneas leídas ]
```

Se realizan los mismos pasos para la zona inversa:

```
[root@localhost Keys]# cat K13.in-addr.arpa.+008+*.key >> db.13
```

Así como en el archivo anterior, se pueden apreciar los datos transferidos en el archivo db.13 (de la zona inversa) desde los archivos .key, previamente.

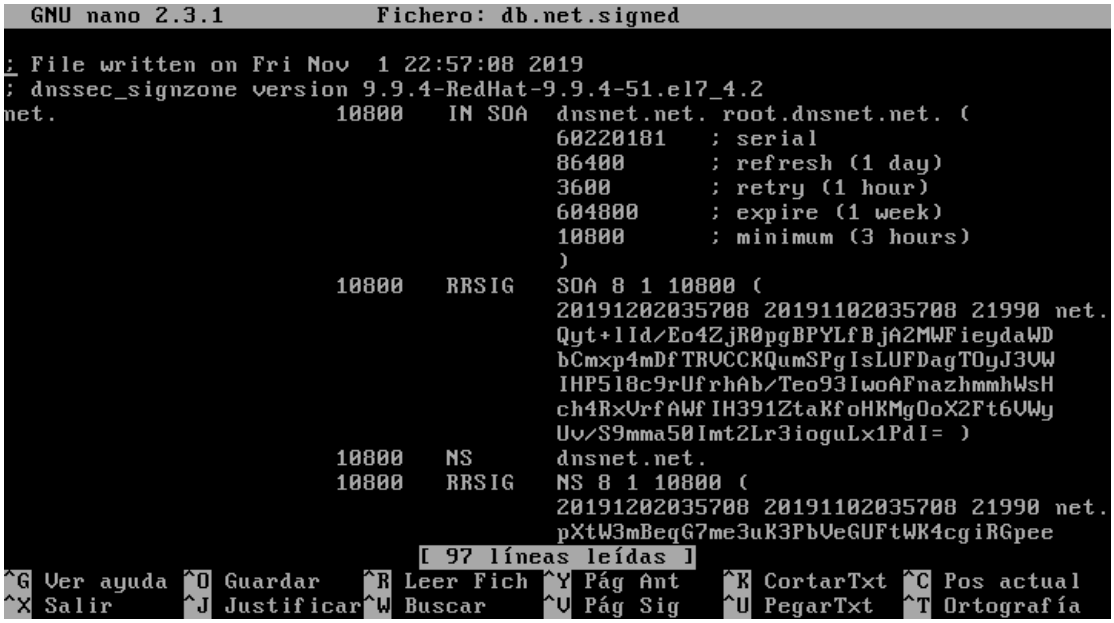
```
 GNU nano 2.3.1              Fichero: db.13

$TTL 3H
@       IN SOA  dnsnet.net.             root.dnsnet.net. (
                                        060220181       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      dnsnet.net.
10.0.0.13.in-addr.arpa. IN      PTR     dnsnet.net.

; This is a zone-signing key, keyid 19980, for 13.in-addr.arpa.
; Created: 20191014022836 (Sun Oct 13 20:28:36 2019)
; Publish: 20191014022836 (Sun Oct 13 20:28:36 2019)
; Activate: 20191014022836 (Sun Oct 13 20:28:36 2019)
13.in-addr.arpa. IN DNSKEY 256 3 8 AwEAAaNOMQm4aamPr04w5HiokNUUrGCoZU4nKsByS8b2$
; This is a key-signing key, keyid 38935, for 13.in-addr.arpa.
; Created: 20191014022849 (Sun Oct 13 20:28:49 2019)
; Publish: 20191014022849 (Sun Oct 13 20:28:49 2019)
; Activate: 20191014022849 (Sun Oct 13 20:28:49 2019)
13.in-addr.arpa. IN DNSKEY 257 3 8 AwEAAcqdIYxlRtHsaRI5BOqMneF805pflRNHaPbPkPoC$
```

Comando de ejecución para el firmado de zona:

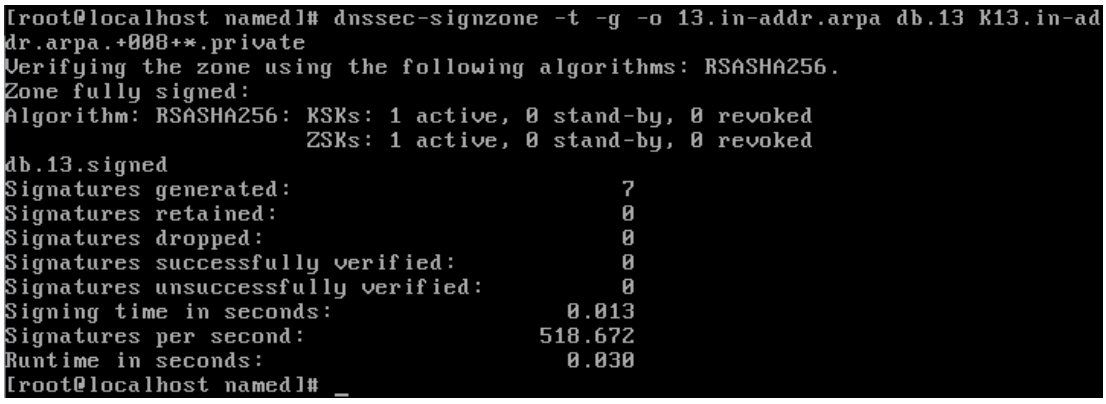
```
[root@localhost named]# dnssec-signzone -t -g -o net db.net Knet.+008+*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.net.signed
Signatures generated:                       8
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.004
Signatures per second:               1669.100
Runtime in seconds:                     0.019
[root@localhost named]#
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t**. TTL
- **-g**.
- **-o net.** Indica el ORIGIN de la zona, es decir el dominio.
- **db.net.** Archivo de configuración de la zona.
- **Knet.+008+*.private.** Llaves privadas.

Archivo generado para la zona principal, al ejecutar el comando anterior.

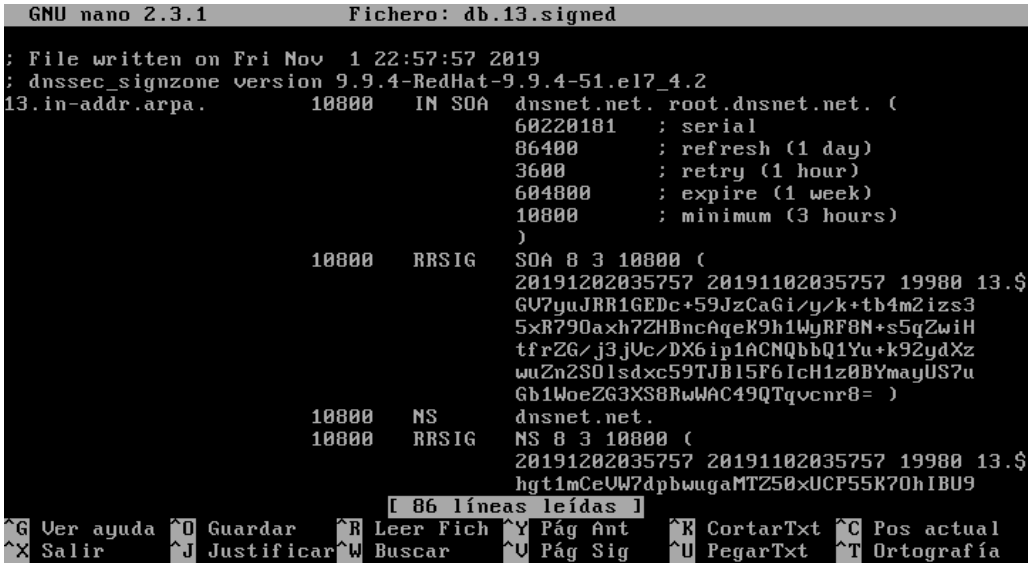


Comando de ejecución para el firmado de la zona inversa:

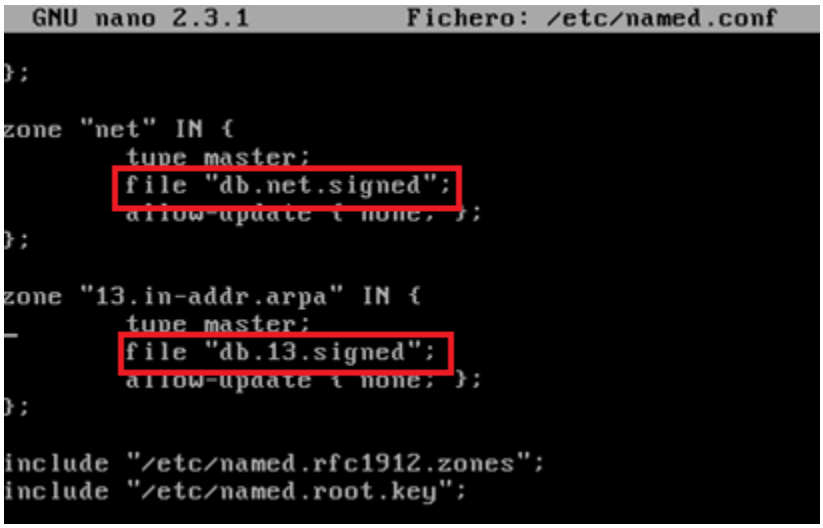


- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t.** TTL
- **-g**
- **-o 13.in-addr.arpa.** Indica el *ORIGIN* de la zona, es decir el dominio.
- **db.13**. Archivo de configuración de la zona inversa.
- **K13.in-addr.arpa.+008+*.private.** Llaves privadas.

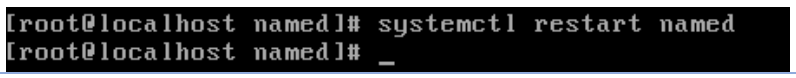
Archivo generado para la zona inversa ya cifrada, al ejecutar el comando anterior:



Una vez que se tiene la zona firmada, lo que queda es cambiar el archivo de zona actual por el generado anteriormente (“db.nombre_de_zona.signed”), tanto para la zona principal como para la zona inversa.



Se reinicia el servicio de named:



Se ejecuta lo siguiente para verificar el registro DNSKEY:

```
[root@localhost named]# dig DNSKEY @13.0.0.10 net. +multiline | more
```

- dig es una herramienta de línea de comando para consultar a un servidor de nombres por registros DNS.

Cuando se ha ejecutado el comando anterior, la salida es:

```
; <<>> DiG 9.9.4-RedHat-9.9.4-51.el7_4.2 <<>> DNSKEY @13.0.0.10 net. +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25970
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;net.                   IN DNSKEY

;; ANSWER SECTION:
net.                    10800 IN DNSKEY 256 3 8 (
                                AwEAAePM1r/2zwCAJX9PBxvxWGcuSwhqYpuTKBXIP81Z
                                797rjEkg+Bw/WDbmjizzaTNu6daQ1X9UG5Dug/5klkZK
                                +MKvrOvyZqnK86VF6EID5fwy261S7Uomf2t+B8DT33ke
                                j75RReZwjAhYuq1Z3NdEYFnPp7ndGukToS33ekYqxk7h
                                ) ; ZSK; alg = RSASHA256; key id = 21990
net.                    10800 IN DNSKEY 257 3 8 (
                                AwEAAa9yKPWLcPFhh8A9aIHqovqRSwNobEu0gfwwSVcN
                                Ud+zA2EMAdsagempaMo7IRD/jwexA5ww/MnuQxGgT+c6
                                Uf0F5Z1ZeVZeE+g+8hFMoHftbdUS119x6z8oGbsOxYhe
--Más--
```

Tabla 25: Implementación de las extensiones DNSSEC en el servidor djj.com

Entidad	Configuración

Servidor DNS djj.com	Principales opciones que controlan el soporte DNSSEC en Bind (en Centos 7 se encuentra en named.conf):
	`dnssec-enable yes;`
	`dnssec-validation yes;`
	`dnssec-lookaside auto;`
	• dnssec-enable, indica soporte DNSSEC como servidor autoritativo de un dominio (en los DNS primario/secundario).
	• dnssec-validation, que activará la validación de respuestas DNSSEC como servidor recursivo (en el DNS que usan nuestras máquinas como resolver).
	• (Opcional) dnssec-lookaside, se usa para configurar la Validación dinámica de Lookaside (DLV).

Ejecutar el siguiente comando:

```
[root@localhost keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE djj.com
Generating key pair.........++++++ ............++++++
Kdjj.com.+008+47679
[root@localhost keys]# _
```

Este comando crea dos ficheros (un **.private** y un **.key**) del tipo **Kdjj.com.+008+47679**. Donde el fichero .private contiene la clave privada y el .key la pública. El +**008** indica el algoritmo elegido (-a RSASHA256) y el +**47679** el identificador de la clave (lo usaremos luego en el registrador). Con **-b 1024** elegimos el tamaño de la clave a generar. **-n ZONE** indica que la clave será usada para trabajar con zonas DNSSEC, **djj.com** es el nombre de la zona.

Luego se procede a ejecutar lo siguiente:

```
[root@localhost keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK djj.com
Generating key pair.....................................................
.............+++ ...........................................................+++

Kdjj.com.+008+10801
[root@localhost keys]# _
```

Una vez más obtendremos dos ficheros (un **.private** y un **.key**) del tipo **Kdjj.com.+008+10801**. Donde, **008** es el algoritmo y **10801** el identificador/etiqueta de la clave. Con **-b 2048** elegimos el tamaño de la clave a generar, 2048 es una buena opción para la *KSK*. -f KSK indica el tipo de clave.

Se realizan los mismos pasos para la zona inversa:

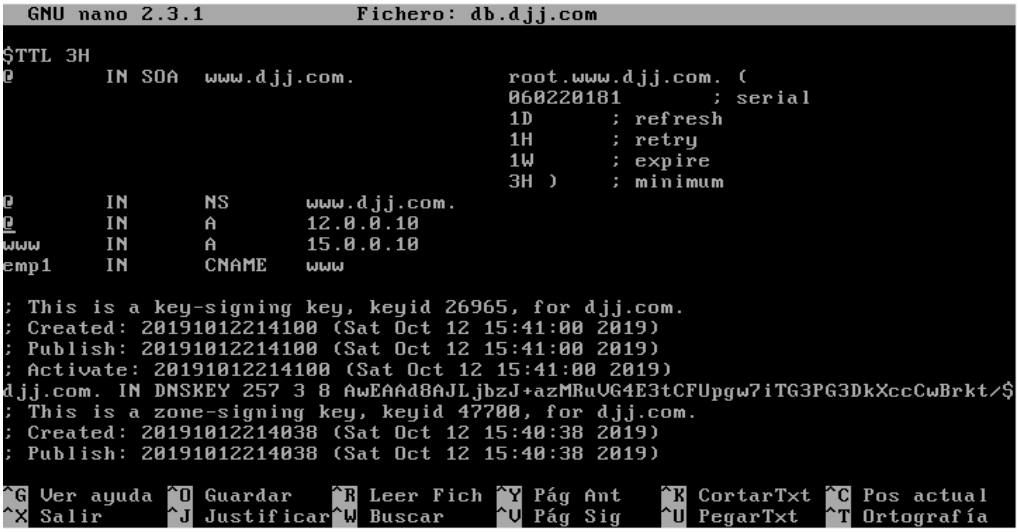
```
[root@localhost keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE 12.in-addr.arp
a
Generating key pair.....................................................
.........++++++ .........................++++++
K12.in-addr.arpa.+008+45561
[root@localhost keys]#
```

```
[root@localhost keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK 12.in-a
ddr.arpa
Generating key pair.....................................+++ ............+++
K12.in-addr.arpa.+008+36952
[root@localhost keys]# _
```

Se transfieren todas las llaves públicas al archivo principal de configuración DNS para luego ser utilizadas.

```
[root@localhost keys]# cat Kdjj.com.+008+*.key >> db.djj.com
```

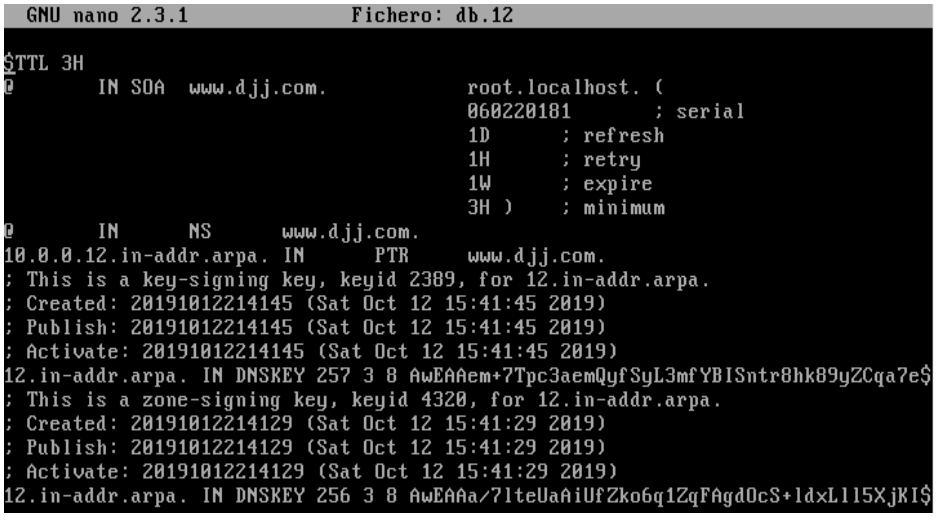
Acá se puede apreciar en el archivo db.djj.com, los datos transferidos desde los archivos .key, previamente.



Se realizan los mismos pasos para la zona inversa.

```
[root@localhost keys]# cat K12.in-addr.arpa.+008+*.key >> db.12_
```

Así como en el archivo anterior, se pueden apreciar los datos transferidos en el archivo db.12 (de la zona inversa) desde los archivos .key, previamente.



Comando de ejecución para el firmado de zona:

```
[root@localhost named]#
[root@localhost named]# dnssec-signzone -t -g -o djj.com db.djj.com Kdjj.com.+00
8+*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.djj.com.signed
Signatures generated:                       20
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                 0.016
Signatures per second:                1242.544
Runtime in seconds:                      0.037
[root@localhost named]#
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t**. TTL
- **-g**.
- **-o djj.com.** Indica el ORIGIN de la zona, es decir el dominio.
- **db.djj**.com. Archivo de configuración de la zona.
- **Kdjj.com.+008+*.private**. Llaves privadas.

Archivo generado para la zona principal, al ejecutar el comando anterior.

```
 GNU nano 2.3.1          Fichero: db.djj.com.signed

; File written on Wed Oct 23 22:59:13 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-51.el7_4.2
djj.com.                10800   IN SOA  www.djj.com. root.www.djj.com. (
                                        60220181   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 2 10800 (
                                        20191123035913 20191024035913 44181 djj$
                                        mop97Qam3rd00Mnmp6vmZFeiVFVYv5c25byA
                                        ceEBgnQjPHZQS/OETgayM0dfavCPFwp4RJAq
                                        mhpckx5g5pGm17Mti9Y/VdvlwrMFIP/M6xFl
                                        lCLd+St8a7Xhqix9u1ufcRJj/I/NLEvqu1mZ
                                        gPodCnMlx6sXV6GFTFW105fjV6g= )
                        10800   RRSIG   SOA 8 2 10800 (
                                        20191123035913 20191024035913 47700 djj$
                                        lJmxiZ3KG+M4hpD4wJqVqbmseoC80Id1js6j
                                        /6n9+6UEGvlwnrB5CaSC+qBBo3GUQuNt8c8L
```

Comando de ejecución para el firmado de la zona inversa:

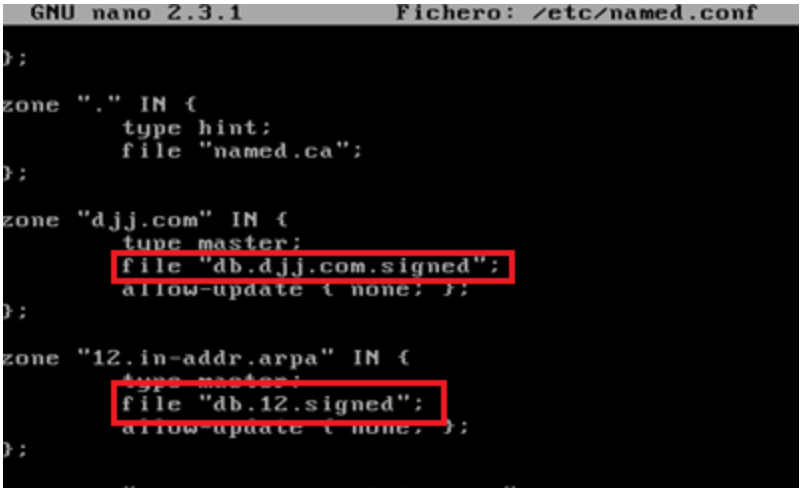
```
[root@localhost named]# dnssec-signzone -t -g -o 12.in-addr.arpa db.12 K12.in-ad
dr.arpa.+008+0*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.12.signed
Signatures generated:                       7
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.004
Signatures per second:               1519.426
Runtime in seconds:                     0.024
```

- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t** TTL
- **-g**
- **-o 12.in-addr.arpa**. Indica el *ORIGIN* de la zona, es decir el dominio.
- **db.12**. Archivo de configuración de la zona.
- **K12.in-addr.arpa.+008+*.private.** Llaves privadas.

Archivo generado para la zona inversa ya cifrada, al ejecutar el comando anterior:

```
  GNU nano 2.3.1           Fichero: db.12.signed

; File written on Wed Oct 23 23:01:15 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-51.el7_4.2
12.in-addr.arpa.        10800   IN SOA  www.djj.com. root.localhost. (
                                        60220181   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 3 10800 (
                                        20191123040115 20191024040115 4320 12.i$
                                        TaYtcT6KQZ40wvOt0/WcCMaXRxYZoMOCPhTa
                                        taEToKK3JtfVEpV4kEmIHU2+2eFn3uvYZRtS
                                        z0iU3Kd3gYWtbc6cxuEGW7N+XuCW0qn/TS60
                                        TIXTy4TgZWbWYUR+d5CbKsUwpZt3a+/1ZDP/
                                        BkDg/lzzP0HZw1q4rshbSx6wKh0= )
                        10800   NS      www.djj.com.
                        10800   RRSIG   NS 8 3 10800 (
                                        20191123040115 20191024040115 4320 12.i$
                                        InFeC00PN2Q038mXtz6Gms94Pk4/MfKAEW+r
                                [ 86 líneas leídas ]
```

Una vez que se tiene la zona firmada, lo que queda es cambiar el archivo de zona actual por el generado anteriormente (“db.nombre_de_zona.signed”), tanto para la zona principal como para la zona inversa.



Se reinicia el servicio de named:

```
[root@localhost named]# systemctl restart named
[root@localhost named]# _
```

Se ejecuta lo siguiente para verificar el registro DNSKEY:

```
[root@localhost named]# dig DNSKEY @12.0.0.10 djj.com. +multiline | more_
```

- dig es una herramienta de línea de comando para consultar a un servidor de nombres por registros DNS.

Cuando se ha ejecutado el comando anterior, la salida es:

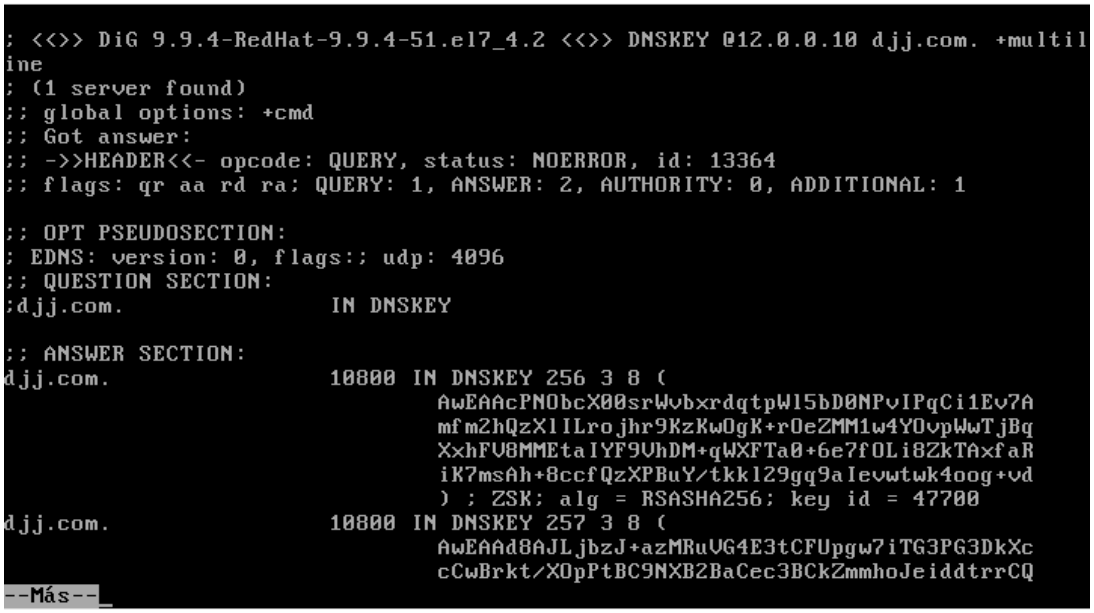


Tabla 26: Implementación de las extensiones DNSSEC en el servidor jab.net

Entidad	Configuración

Servidor DNS jab.net	Principales opciones que controlan el soporte DNSSEC en Bind (en Centos 7 se encuentra en named.conf):
	`dnssec-enable yes;`
	`dnssec-validation yes;`
	`dnssec-lookaside auto;`
	• dnssec-enable, indica soporte DNSSEC como servidor autoritativo de un dominio (en los DNS primario/secundario).
	• dnssec-validation, que activará la validación de respuestas DNSSEC como servidor recursivo (en el DNS que usan nuestras máquinas como resolver).
	• (Opcional) dnssec-lookaside, se usa para configurar la Validación dinámica de Lookaside (DLV).

Ejecutar el siguiente comando:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE jab.net
Generating key pair.............................++++++ .......++++++
Kjab.net.+008+27543
```

Este comando crea dos ficheros (un **.private** y un **.key**) del tipo **Kjab.net.+008+27543**. Donde el fichero .**private** contiene la clave privada y el .key la pública. El +**008** indica el algoritmo elegido **(-a RSASHA256)** y el +**27543** el identificador de la clave (lo usaremos luego en el registrador). Con -b 1024 elegimos el tamaño de la clave a generar. **-n ZONE** indica que la clave será usada para trabajar con zonas DNSSEC, jab.net es el nombre de la zona.

Luego se procede a ejecutar lo siguiente:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK jab.net

Generating key pair.....................................................
.....................................................................
.+++ .....+++
Kjab.net.+008+33574
```

Una vez más se obtiene dos ficheros (un **.private** y un **.key**) del tipo **Kjab.net.+008+33574**. Donde, **008** es el algoritmo y **33574** el identificador/etiqueta de la clave. Con **-b 2048** elegimos el tamaño de la clave a generar, 2048 es una buena opción para la *KSK*. -f KSK indica el tipo de clave.

Se realizan los mismos pasos para la zona inversa:

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 1024 -n ZONE 14.in-addr.arp
a
Generating key pair.....++++++ ........++++++
K14.in-addr.arpa.+008+04275
```

```
[root@localhost Keys]# dnssec-keygen -a RSASHA256 -b 2048 -n ZONE -f KSK 14.in-a
ddr.arpa
Generating key pair.....................+++ ..............................+++
K14.in-addr.arpa.+008+58984
[root@localhost Keys]#
```

Se transfieren todas las llaves públicas al archivo principal de configuración DNS para luego ser utilizadas.

```
[root@localhost Keys]# cat Kjab.net.+008+*.key >> db.jab.net
```

Acá se puede apreciar en el archivo db.jab.com, los datos transferidos desde los archivos .key, previamente.

```
  GNU nano 2.3.1            Fichero: db.jab.net

$TTL 3H
@       IN SOA  www.jab.net.            root.www.jab.net. (
                                        060220182       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.jab.net.
@       IN      A       14.0.0.10
www     IN      A       16.0.0.10
emp2    IN      CNAME   www
; This is a key-signing key, keyid 18214, for jab.net.
; Created: 20191014023709 (Sun Oct 13 20:37:09 2019)
; Publish: 20191014023709 (Sun Oct 13 20:37:09 2019)
; Activate: 20191014023709 (Sun Oct 13 20:37:09 2019)
jab.net. IN DNSKEY 257 3 8 AwEAAdJuLdM0Z7XqfE4T2oNUZQTk8elVYMI9X0Xq6kV58sIOlHXt$
; This is a zone-signing key, keyid 40923, for jab.net.
; Created: 20191014023655 (Sun Oct 13 20:36:55 2019)
; Publish: 20191014023655 (Sun Oct 13 20:36:55 2019)
; Activate: 20191014023655 (Sun Oct 13 20:36:55 2019)
```

Se realizan los mismos pasos para la zona inversa.

```
[root@localhost named]# cat K14.in-addr.arpa.+008+*.private >> db.14
```

Así como en el archivo anterior, se pueden apreciar los datos transferidos en el archivo db.14 (de la zona inversa) desde los archivos .key, previamente.

```
  GNU nano 2.3.1               Fichero: db.14

$TTL 3H
@       IN SOA  www.jab.net.            root.localhost. (
                                        060220182       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS      www.jab.net.
10.0.0.14.in-addr.arpa. IN      PTR     www.jab.net.
; This is a zone-signing key, keyid 21849, for 14.in-addr.arpa.
; Created: 20191014023745 (Sun Oct 13 20:37:45 2019)
; Publish: 20191014023745 (Sun Oct 13 20:37:45 2019)
; Activate: 20191014023745 (Sun Oct 13 20:37:45 2019)
14.in-addr.arpa. IN DNSKEY 256 3 8 AwEAAa1xTf8CS3D6idC5ObdiD+RPDuQ2VQ9q9HgNs/jf$
; This is a key-signing key, keyid 35044, for 14.in-addr.arpa.
; Created: 20191014023756 (Sun Oct 13 20:37:56 2019)
; Publish: 20191014023756 (Sun Oct 13 20:37:56 2019)
; Activate: 20191014023756 (Sun Oct 13 20:37:56 2019)
14.in-addr.arpa. IN DNSKEY 257 3 8 AwEAAe0ISRiiPjAtkrZYXmrV4SJmBkF8CKfxqoSkC7Bt$
```

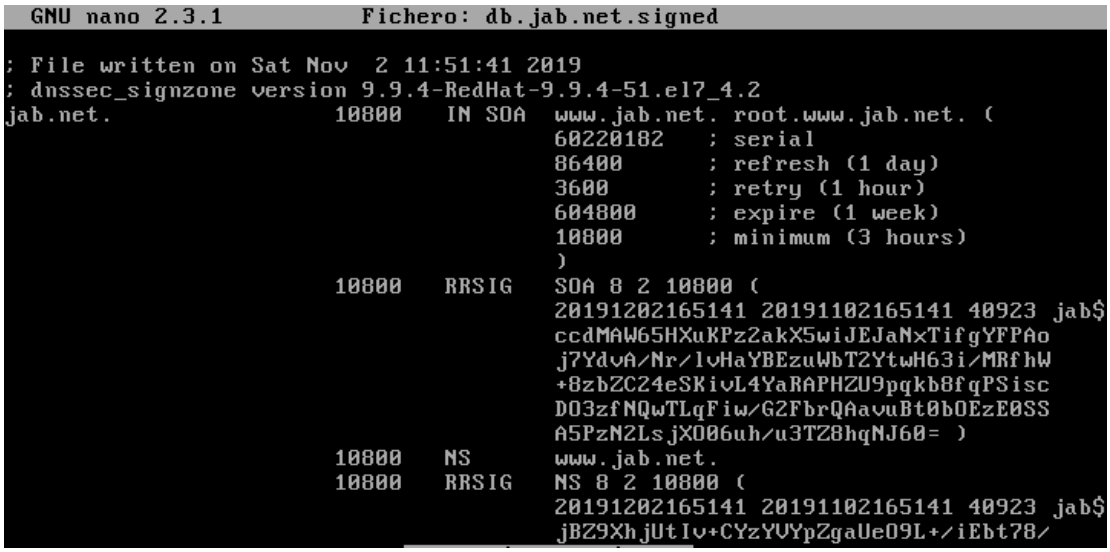
Comando de ejecución para el firmado de zona:

```
[root@localhost named]# dnssec-signzone -t -g -o jab.net db.jab.net Kjab.net.+00
8+*.private
Verifying the zone using the following algorithms: RSASHA256.
Zone fully signed:
Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                      ZSKs: 1 active, 0 stand-by, 0 revoked
db.jab.net.signed
Signatures generated:                       10
Signatures retained:                         0
Signatures dropped:                          0
Signatures successfully verified:            0
Signatures unsuccessfully verified:          0
Signing time in seconds:                 0.005
Signatures per second:                1696.065
Runtime in seconds:                      0.027
[root@localhost named]# _
```

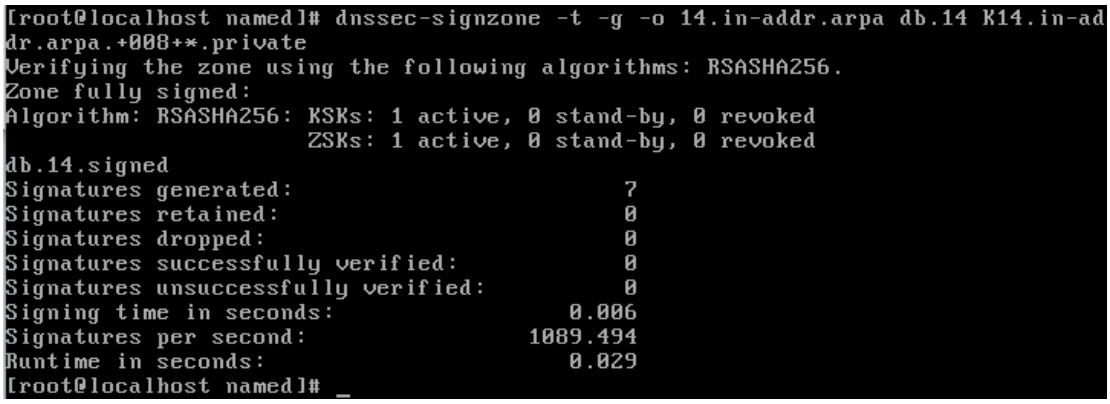
- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t**. TTL
- **-g**.
- **-o jab.net.** Indica el ORIGIN de la zona, es decir el dominio.
- **db.jab.net.** Archivo de configuración de la zona.

- **K jab.net.+008+*.private**. Llaves privadas.

Archivo generado para la zona principal, al ejecutar el comando anterior.



Comando de ejecución para el firmado de la zona inversa:



- **dnssec-sigzone.** Herramienta para el firmado de zona.
- **-t** TTL
- **-g**
- **-o 14.in-addr.arpa.** Indica el ORIGIN de la zona, es decir el dominio.
- **db.14**. Archivo de configuración de la zona.
- **K14.in-addr.arpa.+008+*.private.** Llaves privadas.

Archivo generado para la zona inversa ya cifrada, al ejecutar el comando anterior.

```
 GNU nano 2.3.1          Fichero: db.14.signed

; File written on Sat Nov  2 11:53:23 2019
; dnssec_signzone version 9.9.4-RedHat-9.9.4-51.el7_4.2
14.in-addr.arpa.        10800   IN SOA  www.jab.net. root.localhost. (
                                        60220182   ; serial
                                        86400      ; refresh (1 day)
                                        3600       ; retry (1 hour)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        10800   RRSIG   SOA 8 3 10800 (
                                        20191202165323 20191102165323 21849 14.$
                                        hSqqbOiDP0DBpfC8Ow+gOtw1mIFS1bZNEWTm
                                        /Yf/Hxpu6C+7nyJhwO/efrrGpqZd6NZBdLcR
                                        6K/ul58D05DsawJOymfwmwjTugE6EZvzmMjo
                                        CwbmoEaQp9l21LPp95eEHffkq05xQTz6pPFl
                                        PlsL0M2LoLrd3qcxuQrBo2hieU8= )
                        10800   NS      www.jab.net.
                        10800   RRSIG   NS 8 3 10800 (
                                        20191202165323 20191102165323 21849 14.$
                                        dZsz/jPa8IsKK2WFXPwqz8H2KE/O+vdejk3i
```

Una vez que se tiene la zona firmada, lo que queda es cambiar el archivo de zona actual por el generado anteriormente (“db.nombre_de_zona.signed”), tanto para la zona principal como para la zona inversa.

```
 GNU nano 2.3.1          Fichero: /etc/named.conf

};

zone "jab.net" IN {
        type master;
        file "db.jab.net.signed";
        allow-update { none; };
};

zone "14.in-addr.arpa" IN {
        type master;
        file "db.14.signed";
        allow-update { none; };
};
```

Se reinicia el servicio de named:

```
[root@localhost named]# systemctl restart named
[root@localhost named]# _
```

Se ejecuta lo siguiente para verificar el registro DNSKEY:

```
[root@localhost named]# dig DNSKEY @14.0.0.10 jab.net. +multiline | more
```

- dig es una herramienta de línea de comando para consultar a un servidor de nombres por registros DNS.

Cuando se ha ejecutado el comando anterior, la salida es:

```
; <<>> DiG 9.9.4-RedHat-9.9.4-51.el7_4.2 <<>> DNSKEY @14.0.0.10 jab.net. +multil
ine
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22348
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;jab.net.                       IN DNSKEY

;; ANSWER SECTION:
jab.net.                10800 IN DNSKEY 257 3 8 (
                                AwEAAdJuLdN0Z7XqfE4T2oNUZQTk8elVYMI9X0Xq6kV5
                                8sIOlHXtZxEODFmQPT6AgD3WiwpuSfLyVRySgtJ089x+
                                gLxAmTHcmypaBjMDEy8GEsurDzK8rMWoYjYMzOXZ40tz
                                JthyVPN5yn6OIhJQaAa+JvufrSUriUcrcChhgjy8urzk
                                cs+pfOPESAunAoSv6FsrKNO/XOediU+P8fmTKV6FMNHS
                                plMo8qfCPeQKGvnRWUMBbmyXwSyV67OPUxiVx4kvCR5A
                                +PcbfI6G/ikd8aVYK/pQkeE6QGudLuH+y0/soCdBLdSS
                                Cs8cNJPQzwt5VR30NEtSxKN82dYA41WI0LnI0Ws=
--Más--
```

4.4.11 Conclusión de las extensiones.

DNSSEC no resuelve todos los problemas de seguridad conforme en este escenario virtualizado, sin embargo, merece ser reconocida por todos, para así hacer de internet un lugar más seguro, dado que los delitos cibernéticos son cada vez más difíciles de detectar, todos los propietarios de nombres de dominio deben tener DNSSEC habilitado, además de todas las herramientas en línea disponibles que pueden aumentar el nivel de seguridad en línea.

5 VIDEOS TUTORIALES

5.1 Ataque de envenenamiento ARP (Spooofing) utilizando Ettercap.

En este video tutorial se explica la funcionalidad del escenario virtualizado, de igual manera se expone la configuración y realización de uno de los ataques cuyo método es alterar las direcciones de los servidores DNS que utiliza la potencial víctima, Spoofing.

En el video se observa la realización del ataque desde una máquina que posee el sistema operativo Kali Linux, utilizando la herramienta Ettercap, en donde esta adquiere la petición DNS proveniente de una máquina cliente.

Enlace del video:

https://youtu.be/zBq1InO_MEI

5.2 Implementación de las extensiones DNSSEC en el servidor DNS.

En este video tutorial se detalla de forma precisa la implementación de las extensiones de seguridad DNSSEC, en uno de los servidores DNS presentado en la topología virtual, específicamente el servidor DNS de nombre “Root1”.

En el video se puede apreciar todos los pasos necesarios para lograr dicha implementación, de igual manera, se ve la comprobación de la implementación de las extensiones de seguridad.

Enlace del video:

https://youtu.be/5WdHxNP-ayQ

6 ASPECTOS FINALES

6.1 Conclusiones

- Al exponer los fundamentos de DNSSEC nos dimos cuenta de la importancia de estas extensiones para asegurar el correcto funcionamiento de la infraestructura de DNS.
- Es posible comprobar el correcto funcionamiento de las extensiones de DNSSEC en una topología de red virtualizada corriendo localmente, y su adaptación a un entorno real no requeriría mayor complejidad.
- Los procedimientos implementados para la configuración de las extensiones de DNSSEC en ambientes Linux no son complejas y proporcionan una comprensión completa de los componentes de DNSSEC.

6.2 Recomendaciones

- Ya que muchos servidores DNS en las empresas actuales se configuran en ambiente Windows, se sugiere mejorar este trabajo, añadiendo servidores DNS en ambientes WINDOWS con la configuración respectiva de las extensiones DNSSEC.

- Recomendamos a los proveedores de servicios de internet y a los .ni de cada país, configurar las extensiones DNSSEC en sus servidores DNS locales, para darle seguridad y fortaleza a toda la cadena de consultas.

6.3 Bibliografía

Tanenbaum, Andrew S. (2003). Redes de Computadoras (Cuarta edición).

Symantec. (2014). Tendencias de Seguridad Cibernética en América Latina y el Caribe. https://www.symantec.com/content/es/mx/enterprise/other_resources/b-cyber-security-trends-report-lamc.pdf

IDB. (2016). La seguridad cibernética ¿Estamos preparados en América Latina y el Caribe? https://publications.iadb.org/publications/spanish/document/Ciberseguridad-%C2%BFEstamos-preparados-en-Am%C3%A9rica-Latina-y-el-Caribe.pdf

Guanolique Pereira, César D. (2016). Propuesta de implementación de las extensiones de seguridad DNSSEC en los servidores DNS internos de la Universidad Técnica Particular de Loja.

Altenwald. (2017). ¿Cómo funciona el Servidor de Nombres (DNS)? https://altenwald.org/2017/05/15/como-funciona-el-servidor-de-nombres-dns/

Root Servers. (s.f.) https://root-servers.org/

Comer, Douglas. (1996). Redes Globales de Información con Internet y TCP/IP.

Sánchez, Ernesto. (2012). Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNSSEC). http://sedici.unlp.edu.ar/bitstream/handle/10915/27062/Documento_completo.pdf?sequence=1&isAllowed=y

Armas Gutiérrez, David X. (2017). Análisis de seguridad del sistema DNS (Domain Name System). https://bibdigital.epn.edu.ec/bitstream/15000/17310/1/CD-7804.pdf

Verisign. (s.f.). Desarrollo de las DNSSEC desde sus inicios.
https://www.verisign.com/es_LA/domain-names/dnssec/index.xhtml

Redes Zone. (2018). ¿Qué son los ataques DNS?
https://www.redeszone.net/2018/12/18/ataques-dns-evitarlos/

Albors, Josep. (2017). Ataques al DNS: cómo intentan dirigirte a páginas falsas.
https://www.welivesecurity.com/la-es/2017/02/09/ataques-al-dns/

Verisign. (s.f.). Cómo las DNSSEC brindan el protocolo para una Internet Segura.
https://www.verisign.com/es_LA/domain-names/dnssec/how-dnssec-works/index.xhtml

Martínez E, Victor G. (2013). Configuración del protocolo de enrutamiento RIP (protocolo dinámico)
http://theosnews.com/2013/02/configuracion-del-protocolo-de-enrutamiento-rip-protocolo-dinamico/

Todo sobre Packet Tracer. (2013). Configurar EIGRP.
https://todopacketracer.com/2011/06/19/configurar-eigrp/

Solvetic. (2019). Cómo instalar y configurar servidor DNS en CentOS 7.
https://www.solvetic.com/tutoriales/article/3658-como-instalar-configurar-servidor-dns-centos7/

Desarrollo Web. (2014). Configuración de un servidor web Apache en CentOS.
https://desarrolloweb.com/articulos/configuracion-servidor-web-centos.html

Redes Zone. (2016). Cómo hacer un ataque ARP Poisoning con Kali Linux.
https://www.redeszone.net/2016/11/12/ataque-arp-poisoning-con-kali-linux/

A, Jesin. (2014). How To Setup DNSSEC on an Authoritative BIND DNS Server. https://www.digitalocean.com/community/tutorials/how-to-setup-dnssec-on-an-authoritative-bind-dns-server--2

APNIC. (2018). DNSSEC Tutorial. https://apan.net/meetings/apan46/files/28/28-01-01-01.pdf