Universidad Nacional Autónoma de Nicaragua, UNAN – León

Facultad de Ciencias y Tecnología

Ingeniería en Telemática

Propuesta de prácticas de laboratorio de servicios de red en Debian 8 Kernel 4.9.0 para la carrera de Ingeniería en Telemática de la UNAN-León, en el período comprendido de Agosto a Noviembre 2018.

Tesis para optar al Título de Ingeniero en Telemática

Autor(es):

Br. Alcides José Martínez Zelaya

Br. Juan Carlos Membreño Salmerón

Br. Roderick Randall Narváez Mercado

Tutor: MSC. Denis Espinoza Hernández

León, Nicaragua

Noviembre del 2018

DEDICATORIA

A Dios: Por permitirme tener la fuerza para terminar mi carrera.

A mis padres: Por su esfuerzo en concederme la oportunidad de estudiar y por su constante apoyo a lo largo de mi vida.

A mis hermanos, parientes y amigos: Por sus consejos, paciencia y toda la ayuda que me brindaron para concluir mis estudios.

AGRADECIMIENTO

A mi madre por ser un ejemplo a seguir de trabajo y colaboración con los demás.

A mi papá por ayudarme y apoyarme siempre con sus consejos y su ejemplo de perseverancia, rectitud, integridad y ética.

A mis hermanos por la paciencia que me han tenido.

A mis maestros por compartir conmigo lo que saben y poder transferir sus conocimientos a mi vida.

A Dios por permitirme sonreír nuevamente y tener salud para concluir mis metas.

Índice de contenidos

Índice de ilustraciones

Índice de tablas

1 Introducción

Un servicio de red es un medio por el cual un sistema dispar se comunica con otro, para interactuar y compartir datos entre si [1], la finalidad es proporcionar a los usuarios de los servicios un rendimiento global optimo transparente [2].

Debido al impacto de los servicios de red en el uso masivo de internet, proteger los mismos se volvió un asunto de mucha importancia, el cual requiere no solo el intercambio de información fiable sino también de la confidencialidad entre las partes autorizadas para el intercambio de información [3].

El presente documento tiene como propósito desarrollar propuesta de prácticas de laboratorio de Servicios de Red, para la carrera de Ingeniería en Telemática de la UNAN-León; en el cual se desarrollarán enunciados de prácticas de laboratorio y se abordarán temáticas teórico-prácticas, de tal manera, que los estudiantes adquieran los conocimientos necesarios para ser puestos en práctica durante la carrera.

1.1 Antecedentes

En la UNAN-León se han elaborado tesis relacionadas a este tema de investigación, entre los cuales tenemos:

"Plan Docente para el componente Aplicaciones Telemáticas", elaborado por el MSc. Denis Leopoldo Espinoza Hernández, Julio 2007. Este documento aborda el desarrollo del Plan de Docente del componente Aplicaciones Telemáticas que se imparte en el cuarto año de la carrera de Ingeniería en Telemática de la UNAN-León. Además, se presenta la situación actual de dicho componente, su relación con otros componentes, la metodología y material didáctico de apoyo para impartirla, el sistema de evaluación, la planificación temporal, la presentación del contenido teórico, el desarrollo de las prácticas y la bibliografía necesaria para las mismas.

Es importante resaltar que, aunque el documento presenta prácticas que se abordarán en esta tesis, la distribución y los paquetes se encuentran fuera de uso, las prácticas de laboratorio no poseen una topología o diagrama de red referente al escenario a crear y las prácticas se desarrollan con el uso de una a dos máquinas, el presente documento contiene los servicios mas usados hoy día, actualiza la distribución y los paquetes, presenta un diagrama de red referente a la práctica, propone los servicios en máquinas independientes y define un formato de práctica que incluye preguntas de compresión.

"Propuesta de prácticas de laboratorios de Switching y Routing para la carrera de Ingeniería en Telemática UNAN-León", elaborado por el Br Rudy Otoniel Quiróz Vázquez**,** el Br. Franklin Ernesto Ramírez Medina y el Br. Yoel Francisco Rivera González, en septiembre del 2013. Este documento propone la elaboración de 15 prácticas de laboratorio actualizadas, con la finalidad de abordar temáticas teóricas-prácticas más avanzadas y más ajustadas a la realidad. Dichas prácticas han sido de gran apoyo sobre todo en los componentes curriculares de Comunicación de Datos y Redes Computadores.

“Elaboración de prácticas de Switching, Routing y Servicios de Red con IPv6 para el componente Electiva X correspondiente al plan académico 2011 de la carrera de Ingeniería en Telemática del Departamento de Computación de la UNAN-León”, elaborado por el Br. Delia María Jaime Toruño, Br. Hugo Mariano García Machado y Br. William Francisco Aguilar Zapata, en mayo 2015. Este documento tiene como finalidad exponer las características más importantes del protocolo IPv6, a través del desarrollo de contenido teórico y de propuestas de prácticas de laboratorio guiadas, dirigidas a estudiantes del último año de la carrera Ingeniería en Telemática de la UNAN-León.

“Elaboración de prácticas de laboratorio para el componente Computación en la nube del plan académico 2011 de la carrera de Ingeniería en Telemática de la UNAN-León”, elaborado por el Br. Angel Evelio Maradiaga Leytón, Br. Ervin Ismael Montes Téllez, en agosto 2015. Este documento tiene como propósito desarrollar prácticas de computación en la nube, a través de elementos teóricos y prácticos, dirigidos al componente de computación en la nube del plan académico 2011.

1.2 Planteamiento del problema

El desarrollo de prácticas de laboratorio relacionadas con servicios de red, son esenciales para la formación académica de los estudiantes. Sin embargo, la carencia de documentación y/o bibliografía oficial que brinde una correcta integración de las prácticas y el contenido teórico, dificulta la realización de estas por parte de los estudiantes.

Los aspectos antes mencionados provocan que los docentes, hoy en día no posean un documento de referencia actualizado, que brinde las asignaciones de prácticas relacionadas con Servicios de Red.

Ante las necesidades expuestas, surgen las siguientes interrogantes:

- ¿Qué formato se debe definir para lograr una adecuada comprensión de los Servicios de Red por parte de los estudiantes?
- ¿Qué relación debe existir entre cada una de las prácticas de laboratorio a fin de garantizar una integración entre ellas?
- ¿Cómo debe ser abordado el desarrollo teórico de cada una de las prácticas a fin de garantizar que el estudiante no solo adquiera los conocimientos de configuración del servicio; sino también la comprensión del protocolo que implementa?

1.3 Justificación

Tomando como referencia, los problemas antes expuestos, se da la idea de crear un documento, donde se definirán de manera clara y ordenada conocimientos teóricos y se desarrollarán prácticas de laboratorio referente a servicios de red. Brindando a los estudiantes de la carrera de Ingeniería en Telemática los conocimientos necesarios para enfrentar desafíos reales. Por este motivo hemos decidido: Proponer prácticas de laboratorio que aumenten gradualmente su relación y a fines a algunos escenarios de la vida real.

1.3.1 Originalidad

Anteriormente se han realizado trabajos y documentos que contienen prácticas enfocadas a componentes específicos; sin embargo, el presente documento pretende aportar a UNAN-León una guía completa de prácticas de servicios de red escalables a diversas asignaturas para la carrera de Ingeniería en telemática, que ayudará de manera significativa al proceso de enseñanza-aprendizaje por parte de maestros y estudiantes.

1.3.2 Alcance

Para los docentes: Un documento que les permita asignar prácticas semi-guiadas a los estudiantes.

Para los estudiantes: Comprender los temas y resolver las prácticas propuestas en el documento, obtener conocimientos cruciales para la formación profesional creando un dominio de los conceptos establecidos.

1.3.3 Producto

El documento presenta las características siguientes:

- **Semi-Guiado**: Describirá los puntos principales para realizar cada práctica.
- **Sencillo**: El desarrollo de las prácticas y la documentación, serán elaboradas de forma que se comprenda fácilmente.
- **Secuencial**: Las prácticas se proponen de menor a mayor grado de integración, de tal forma que el estudiante pueda lograr obtener el aprendizaje necesario.

1.3.4 Impacto

El documento permitirá contar con un material de soporte eficaz tanto para el personal docente, como para estudiantes y público en general con mismos fines. Con esto, se contribuye a que el estudiante tenga las herramientas necesarias para enfrentar los retos que presentan organizaciones reales en la actualidad.

1.4 Objetivos

1.4.1 Objetivo General

Elaborar propuesta de prácticas de laboratorio de servicios de red en Debian 8 Kernel 4.9.0 para la carrera de ingeniería en Telemática de la UNAN-LEÓN, en el período comprendido de Agosto 2018 a Noviembre 2018.

1.4.2 Objetivos Específicos

- Definir un formato de práctica que logré una adecuada comprensión de los conocimientos por parte de los estudiantes.
- Establecer relación entre cada una de las prácticas de laboratorio a fin de garantizar integración entre ellas.
- Elaborar un documento con información teórica y práctica, que sirva de base a los profesores y estudiantes para el desarrollo de las prácticas de laboratorios.

2 Marco teórico

Conceptos y términos generales de la capa de aplicación y sus protocolos, con la finalidad de refrescar conocimientos y establecer conexión entre el contenido teórico y el desarrollo de las prácticas, para profundizar en detalles se dejará enlaces directos a los RFC en el apartado Bibliografía.

2.1 Protocolos de la capa de aplicación

2.1.1 Servicios de Red

La finalidad de toda red es que los usuarios puedan compartir recursos por medio servicios que aporten mejor rendimiento a las organizaciones, los servicios de red son configurados en redes locales corporativas para brindar un funcionamiento de la red efectivo y fácil de manejar [2].

Los servicios de red son servicios que corren en la capa de aplicación, capa número 7 del modelo OSI (Open System Interconected) o bien capa 5 de la pila de protocolos TCP/IP (TRANSFER CONTROL PROTOCOL) entre lo más importantes podemos nombrar DHCP, DNS, SMTP, IMAP, HTTP entre otros [4][10].

2.1.2 La capa de aplicación

Encargada de ofrecer a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y de definir los protocolos que se utilizan para el intercambio de datos [5].

Es importante distinguir entre las aplicaciones de red y los protocolos de la capa de aplicación. Un protocolo de la capa de aplicación es únicamente una parte (aunque importante) de la aplicación de red. Como ejemplo tenemos que la aplicación Web, utiliza el protocolo de la capa de aplicación HTTP (Protocolo de Transferencia de Hipertexto) y el correo electrónico utiliza principalmente el protocolo (SMTP).

Como podemos apreciar, a veces el protocolo tiene un nombre diferente de la aplicación, sin embargo, existen otros casos en que se emplea el nombre del protocolo para referirse a la aplicación, un ejemplo claro es DNS. (Servidor de nombres de dominios)

En particular, un protocolo de la capa de aplicación define [6]:

- Tipo de mensajes intercambiados.
- Sintaxis de los distintos mensajes.
- La semántica de los campos.
- Las reglas que determinan cuándo y cómo un proceso envía mensajes y responde a los mensajes.

Es importante mencionar que la mayoría de los protocolos dentro de la capa de aplicación están disponibles en los RFC (Request For Comment), son dominios públicos y su lectura es totalmente libre y gratuita.

2.1.3 DHCP

Dynamic Host Configuration Protocol (DHCP), provee los parámetros de configuración de red a los equipos finales. DHCP es un protocolo construido en el modelo cliente servidor, consta de dos componentes [7]:

- Un protocolo para asignación de parámetros específicos de configuración para equipos de un servidor DHCP a un host.
- Un mecanismo de asignación de direcciones de red.

En el modelo de funcionamiento de DHCP, el servidor es el encargado de asignar las direcciones de red y repartir los parámetros de configuración para que los equipos se configuren de forma dinámica.

- Cliente DHCP es un equipo en una red que usa el protocolo DHCP para obtener sus parámetros de configuración.
- Servidor DHCP es un equipo en una red que envía los parámetros de configuración a los clientes DHCP.
- Agente DHCP (Relay) es un equipo o router en una red que sirve para llevar los mensajes de un cliente DHCP a un servidor DHCP.

2.1.3.1 Mecanismo para asignación de direcciones.

DHCP soporta 3 mecanismos de asignación de direcciones IP:

- Asignación Automática, DHCP asigna una dirección IP permanente para un cliente.
- Asignación Dinámica, DHCP asigna una dirección IP para un cliente por un periodo de tiempo limitado o hasta que el cliente renuncie a dicha dirección IP.
- Asignación Manual, La dirección IP del cliente es asignada por el administrador de la red y DHCP se encarga simplemente de transmitir la dirección asignada al cliente. Una red en particular podría usar uno o más de estos mecanismos dependiendo de las políticas del administrador de red.

De los tres mecanismos, la asignación dinámica es el único método que permite la reutilización de direcciones ya que la dirección no esta unida al equipo cliente y esta solo es asignada por un periodo de tiempo. La asignación dinámica es muy usada para asignar direcciones a equipo que solo se conectara a la red por un tiempo limitado o para asignar un número de direcciones entre un grupo de equipos que no necesitan direcciones permanentes.

DHCP fue diseñado para suministrar los parámetros de configuración definidos a los clientes, una vez que un cliente recibe su configuración este debería poder establecer comunicación con los demás equipos.

Cabe mencionar que DHCP permite la configuración de muchos parámetros que no corresponden al protocolo IP pero son necesarios, es decisión del administrador de la red definir las configuraciones que necesita en una determinada subred.

2.1.3.2 Objetivos de diseño de DHCP.

- Cada cliente podría necesitar descubrir sus apropiados parámetros de configuración e incorporarlos en su configuración local sin necesidad de la intervención de un usuario.
- Una red podría no requerir configuración manual por clientes de forma individual, es decir el administrador de la red no debería tener que ingresar ninguna configuración por cliente.

- DHCP no requiere de un servidor en cada subred, gracias a los agentes DHCP, es posible tener las configuraciones de muchas subredes en un solo servidor.
- Garantizar que una dirección IP no estará en uso por más de un cliente DHCP.

Estos son algunos de los objetivos de diseño del protocolo DHCP.

2.1.3.3 Formato de los mensajes DHCP.

Un mensaje DHCP tiene el siguiente formato, los mensajes de solicitud y respuesta tienen el mismo formato.

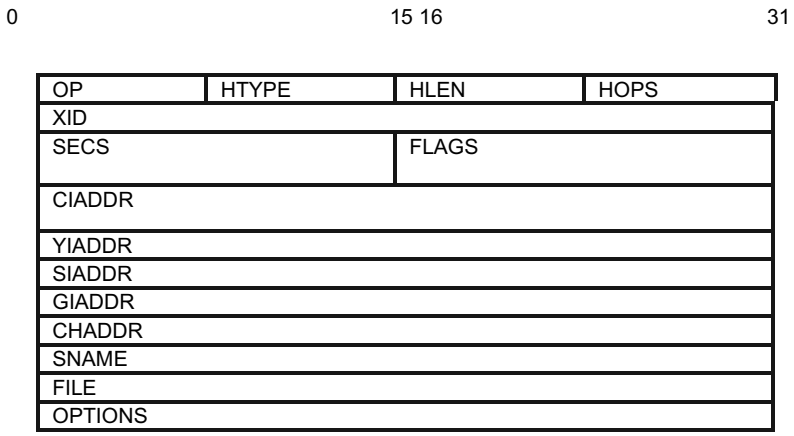

Ilustración 1. Formato de mensaje DHCP

Campo	Bytes	Descripción
OP	1	Código de Opción 1=REQUEST, 2 = REPLY
HTYPE	1	Tipo de dirección e hardware Ej: 1 = Ethernet 10mb
HLEN	1	Longitud de dirección de hardware Ej: Ethernet 10mb = 6
HOPS	1	Establecido a 0 por el cliente usado opcionalmente por los Agentes DHCP cuando reenvían mensajes
XID	4	Identificador de transacción, valor aleatorio elegido por el cliente, usado para asociar mensajes entre el cliente y el servidor
SECS	2	Llenado por el cliente, indica los segundos transcurridos desde que el cliente inició la adquisición de la dirección.
FLAGS	2	Usado para indicar si el mensaje es un broadcast, si es así el bit más a la izquierda es 1 y los demás son 0
CIADDR	4	Dirección IP del cliente solo es agregada si el cliente esta en estado BOUND, RENEW o REBINDING
YIADDR	4	La dirección IP del cliente
SIADDR	4	Dirección IP del próximo servidor para usar en el proceso de inicio
GIADDR	4	Dirección IP del agente usado cuando la solicitud se realiza a través de un agente.
CHADDR	16	Dirección de hardware del cliente.
SNAME	64	Nombre opcional del host servidor acabado en X'00'
FILE		El cliente o bien deja este campo vacío o especifica un nombre genérico, indicando el tipo de fichero de arranque a usar.
OPTIONS		Este parámetro es opcional y de un tamaño variable, revisar el rfc 1533 para una lista completa de opciones.

Ilustración 2. Descripción de los campos de un mensaje DHCP

2.1.3.4 Tipos de Mensajes DHCP.

- **DHCPDISCOVER**: Este mensaje es usado para que el cliente pueda localizar al servidor DHCP.
- **DHCPOFFER**: Enviado por servidor como respuesta de un mensaje DHCPDISCOVER.
- **DHCPREQUEST**: Usado por el cliente para indicar que acepta la configuración ofrecida por el servidor en cuestión, el servidor registra al cliente.
- **DHCPACK**: Enviado por el servidor este mensaje incluye los parámetros de configuración.
- **DHCPNAK**: La dirección IP solicitada pertenece a una subred diferente o está en uso por otro equipo.
- **DHCPDECLINE**: Enviado por el cliente, informa al servidor de que la dirección ofrecida ya está en uso.
- **DHCPRELEASE**: Enviado por el cliente, indica al servido que el cliente renuncia a la dirección IP.
- **DHCPINFORM**: Usado por los equipos en red para solicitar y obtener información de un servidor DHCP para usarla en su configuración local.

2.1.3.5 Asignación de dirección de red mediante DHCP.

Cuando un nuevo cliente DHCP quiere obtener su configuración de red, este debe establecer comunicación con el servidor DHCP, para conseguirlo estos deben seguir el procedimiento de solicitud, este proceso se describe a continuación:

El cliente debe descubrir al servidor, ya que el cliente en este momento no cuenta con una dirección IP, por lo cual realiza un broadcast de DHCPDISCOVER.

El servidor recibe la solicitud, determina la configuración y responde con un mensaje DHCPOFFER.

El cliente recibe 1 o más mensajes DHCPOFFER de más de 1 servidor, el cliente elige a un servidor basándose en los parámetros de configuración ofrecidos, y responde con un mensaje DHCPREQUEST, el servidor elegido es identificado a través del campo XID.

El servidor recibe el mensaje DHCPRESQUEST, verifica si es el servidor seleccionado (lo verifica con el campo XID) si no lo es, supone que el cliente no acepto su oferta, el servidor seleccionado en el mensaje DHCPREQUEST confirma y enlaza al cliente con su almacenamiento persistente y responde con un mensaje DHCPACK que contiene los parámetros de configuración,

En este momento el cliente ya está configurado, el cliente podría realizar un chequeo final de los parámetros obtenidos, por ejemplo, una comprobación con ARP (Address Resolution Protocol) si el cliente detecta que la IP ya está en uso, el cliente puede enviar un mensaje DHCPDEHCLINE al servidor y reiniciar el proceso de asignación.

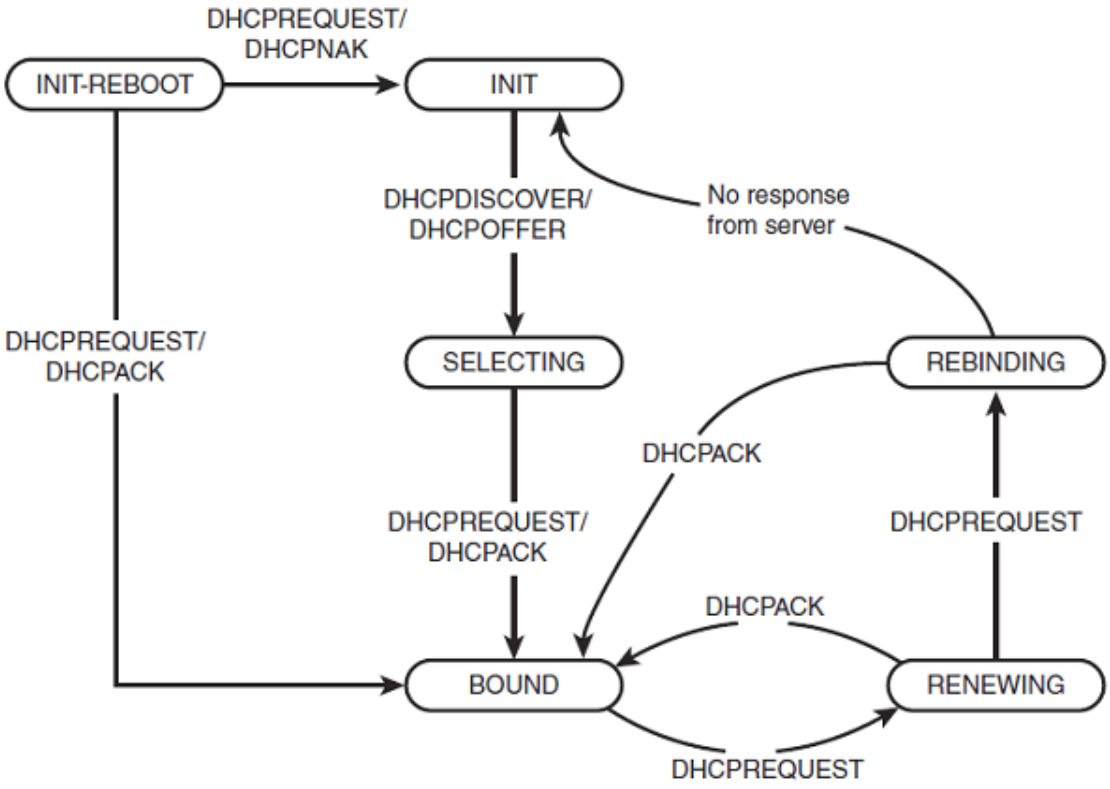

Ilustración 3. Estados del cliente DHCP

2.1.4 HTTP

El protocolo de transferencia de hipertexto (HTTP) es un protocolo de la capa de aplicación del modelo de referencia OSI y esta basado en el modelo cliente/servidor, el uso principal de HTTP es solicitar y transmitir archivos a través de una red especialmente a las aplicaciones web y sus componentes, estos archivos también son conocidos o llamados objetos, aunque en la práctica un sistema de información requiere más funcionalidades que solo transferir archivos por ejemplo actualizaciones de interfaz, incluir formularios, etc. HTTP provee un conjunto de métodos para ser usados para indicar el propósito de la solicitud. [8]

HTTP opera en el esquema solicitud-respuesta (un cliente solicita y un servidor responde), HTTP usa como protocolo de nivel de transporte el protocolo TCP. También hay que mencionar que HTTP es un protocolo sin estado, esto quiere decir que en ningún momento el protocolo maneja información sobre el usuario o posibles solicitudes realizadas con anterioridad.

2.1.4.1 Funcionamiento

Como se mencionó HTTP funciona de forma tal que el cliente realiza una solicitud y el servidor la procesa y responde, este hecho se puede ver cuando accedemos a un sitio web desde nuestro navegador, podríamos intentar entrar al sitio: http://www.google.com/. Esto hace que el navegador le solicite un recurso (objeto) al servidor de Google, pero antes de que eso pase se deben realizar algunas acciones antes de poder realizar la solicitud:

El cliente necesita la dirección IP del servidor de Google para poder realizar la conexión, para esto primero el cliente invoca una rutina la cual usa el protocolo DNS para la traducción del nombre de dominio a la dirección IP correspondiente. Una vez que se obtiene la dirección IP, cliente establece una conexión TCP con el servidor, ya que HTTP usa TCP como protocolo de nivel de transporte.

Cuando se establece la conexión TCP entre los procesos cliente y servidor, el cliente puede enviar la solicitud al servidor, el proceso es el mismo para todas las conexiones, aunque existen casos en los que el cliente almacenara en cache algunos objetos con lo cual la próxima vez que sean solicitados estos no se obtendrán del servidor, sino de la cache del

cliente.[10] Cabe mencionar que HTTP tiene 2 modos de funcionamiento, estos modos indican la forma en cómo se deberá llevar a cabo la comunicación. Estos modos son:

2.1.4.2 Modo No Persistente

Este modo de funcionamiento indica que cada solicitud se debe realizar en una conexión distinta, así si usted solicita un sitio web que consta de un documento HTML, un archivo CSS, un archivo JavaScript y 3 imágenes, el cliente deberá realizar 6 conexiones distintas al servidor, una por cada objeto.

2.1.4.3 Modo Persistente

El modo persistente, es lo inverso del modo no persistente con lo cual a través de una única conexión se envían todos los objetos que se soliciten, entonces con respecto al ejemplo anterior el cliente solo deberá establecer una conexión con el servidor para poder obtener los 6 objetos.

2.1.4.4 Formato de solicitud HTTP

Una solicitud HTTP, es un mensaje que es enviado por el cliente al servidor este mensaje debe corresponder con el siguiente formato de mensaje.

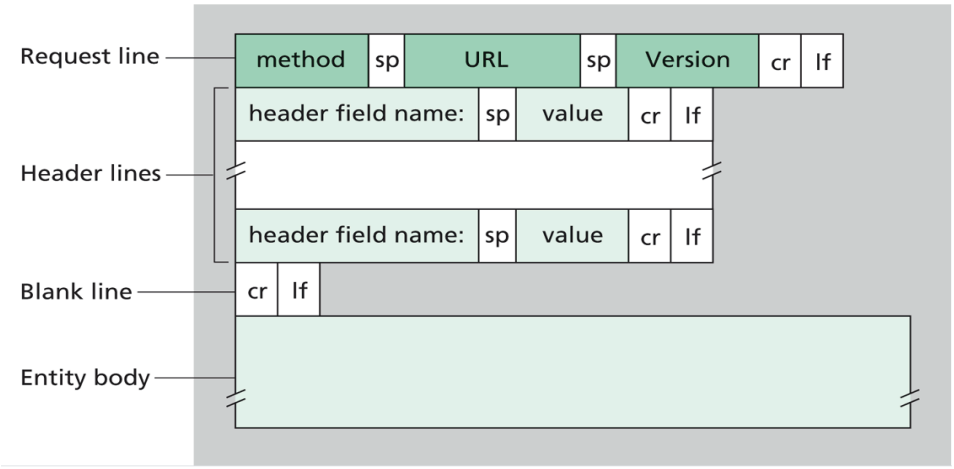

Ilustración 4. Formato de solicitud HTTP

Como se observa en la imagen la solicitud consta de 2 partes: La cabecera y el cuerpo.

Cabecera: La cual a su vez se divide en dos partes:

- **Línea de solicitud:** Esta es la línea que indica el objeto que se quiere solicitar, también define el verbo HTTP que se usa para la solicitud y la versión de HTTP.
- **Líneas de Cabecera**: En esta parte se definen algunas opciones con las cabeceras del protocolo. Estas cabeceras del protocolo se definen según la versión de HTTP que se esté usando al momento de escribir este documento existen 4 versiones de HTTP: 1.0, 1.1, 1.2, 2.0.

2.1.4.5 Verbos HTTP

GET: Es uno de los métodos más usados, ya que este método es el que se usa cuando damos clic a un enlace, cuando escribes un url en la barra de direcciones, etc. El propósito del verbo GET es el solicitar un recurso al servidor, es decir cuando un cliente usa este verbo solo le está solicitando al servidor que le envié una copia de ese objeto, otro aspecto del método GET es que los objetos están identificados por un URI.

- **POST:** Es usado para crear un objeto o recurso, por ejemplo, cuando queremos subir un archivo a un servidor, usamos el verbo post.
- **PUT:** Este verbo es usado para realizar actualizaciones, por ejemplo, actualizar el contenido de un objeto o reemplazarlo.
- **DELETE:** Es lo inverso de POST y como su nombre lo dice, este verbo se usa para indicarle al servidor que queremos borrar un objeto.
- **HEAD:** Funciona igual que GET, pero con la diferencia de que el servidor solo devuelve las cabeceras y no el cuerpo del mensaje.

2.1.4.6 Cuerpo del mensaje:

En esta parte es donde va el cuerpo de la solicitud, si es que la hay. Un ejemplo de la solicitud podría ser:

```
GET / HTTP/1.1
Host: www.google.com
User-Agent: "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Firefox/52.0"

<Cuerpo de la solicitud>
```

Donde se puede apreciar a como se dijo previamente, la primera línea indica el verbo a usar, el verbo es usado para indicar acciones, el objeto que en este caso es "/" y la versión de HTTP que este caso es 1.1, luego se ven otras cabeceras como Host que indica el nombre de dominio del servidor y opcionalmente el número de puerto, también está el campo User-Agent el cual indica el agente de usuario, en este caso un Mozilla/Firefox.

2.1.4.7 Formato de Respuesta HTTP

Al igual que las solicitudes, las respuestas también tienen un formato definido el cual tiene casi la misma estructura que la solicitud, la única diferencia es que el formato de respuesta no cuenta con una línea de solicitud sino con una línea de estatus, esta línea nos brinda información acerca del estado de la solicitud.

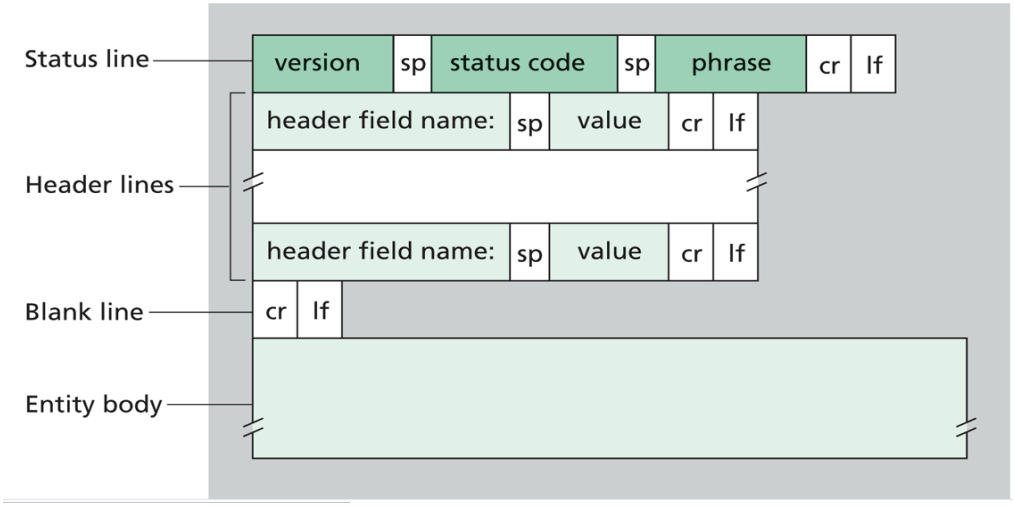

Ilustración 5. Formato de Respuesta HTTP

Ejemplo de respuesta HTTP:

```
HTTP/1.1 200 OK
Content-Type: "text/html; charset=UTF-8"
Content-Length: "7375"
Last-Modified: "Mon, 12 Dec 2016 14:45:00 GMT"
Server: "Apache(Linux/Debian)"

<html>
....
</html>
```

En este caso vemos como en la línea de status aparece un código 200, HTTP usa unas familias de códigos para indicar el estado de la solicitud para este ejemplo el código es 200 y significa que todo salió bien y que el objeto pudo ser rescatado.[10]

2.1.4.8 Códigos de estados HTTP

Al igual que el cliente puede usar un gran número de verbos el servidor también tiene a su disposición un conjunto de números llamados códigos de estado los cuales son usados para indicarle al cliente el estado de su solicitud, estos códigos se agrupan en familia:

- 1XX Respuestas informativas
- 2XX Peticiones correctas
- 3XX Redirecciones
- 4XX Errores del cliente
- 5XX Errores de servidor

2.1.5 DNS

La naturaleza humana nos invita a asociar nombres o etiquetas a diferentes tipos de cosas, animales o personas es por eso que nos resulta sencillo recordar etiquetas cuando tenemos que hacer referencia a algo o alguien, a diferencia de los ordenadores que por su naturaleza programada solo son capaces de reconocer números (A BAJO NIVEL), estos números (Identificadores) son lo que en realidad se utiliza en internet para identificar terminales. Los nombres de terminales (como www.google.com, www.facebook.com, www.yahoo.com) son mnemotécnicos y muy fáciles de recordar por personas. Sin embargo, no proporcionan alguna información y son difíciles de procesar por los ordenadores, por este motivo es que los terminales también son identificados en internet por una dirección IP. [9]

Las direcciones IP (Internet Protocol) son un numero único e irrepetible con el cual se identifica a una computadora, en la cual, leída de izquierda a derecha, se obtiene cada vez información más específica sobre la ubicación de terminal en internet [10].

2.1.5.1 Historia

En la época de los 70´s con el inicio de ARPANET, cada ordenador conectado a la red tenía asignada una dirección numérica por lo que para acceder a aquellos equipos era necesario

tener que recordar la dirección numérica de cada uno, por lo que no existía algún sistema de “nombramiento” [10].

2.1.5.2 El archivo Host y sus problemas

Una vez que fue reconocida la necesidad de establecer una relación entre direcciones-IP y su nombre terminal, se planteó la primera solución que proponía solucionar el problema de “nombramiento” al almacenar en un fichero de texto (HOST) la traducción nombre-IP.

El problema con este fichero no solo consistía en cada administrador de red tenía que emitir por correo los cambios que ocurrieran en su red ni tampoco en descargar y actualizar manualmente el archivo, sino que con el crecimiento significativo de ordenadores conectados a la red el archivo era demasiado grande que se volvía difícil de manejar por lo cual se generaban las siguientes inconsistencias:

- Las instalaciones reguladoras (SRI-NIC) no soportaban tal carga.
- No había un mecanismo eficaz para evitar que aparecieran nombres duplicados.
- Inconsistencia de archivos.

Estos problemas antes mencionados trajeron como consecuencia la creación de un nuevo sistema conocido como Sistema de Nombres de Dominio (Domain Name Service, DNS).

2.1.5.3 Sistema de nombres de dominios

Este sistema debía cumplir con tres objetivos principales [9]:

- Repartir la carga entre varias máquinas, cada una debería mantener información local, pero hacerla accesible globalmente.
- La administración debía ser descentralizada, de tal forma que no se concentre en un solo terminal y evite cuellos de botellas.
- Evitar que tenga nombres duplicados, con lo cual la solución fue, crear un sistema jerárquico.

La primera definición de DNS se encuentra en las RFC´S 882 y 883, las especificaciones actuales se encuentran en RFC 1034 y 1035.

2.1.5.4 Definiciones y descripción

El DNS es un sistema de nomenclatura jerárquica que asocia información terminal con nombres de dominio, su función principal es traducir (resolver) nombres inteligibles para personas en identificadores binarios asociados con equipos conectados a la red, esto con el propósito de localizarlos. [10]

DNS utiliza una base de datos distribuida implementada de forma jerárquica, a su vez es una aplicación y protocolo de la capa de aplicación que permite que los terminales y los servidores se comuniquen para ofrecer un servicio de traducción.

La estructura empleada es cliente-servidor que está construida por:

- Servidores de nombres: Contiene información sobre fragmentos de la base de datos.
- Clientes (resolver): Programas quienes formulan las consultas.

2.1.5.5 Modo de funcionamiento

Se presenta ahora, un resumen a alto nivel de cómo trabaja el DNS.

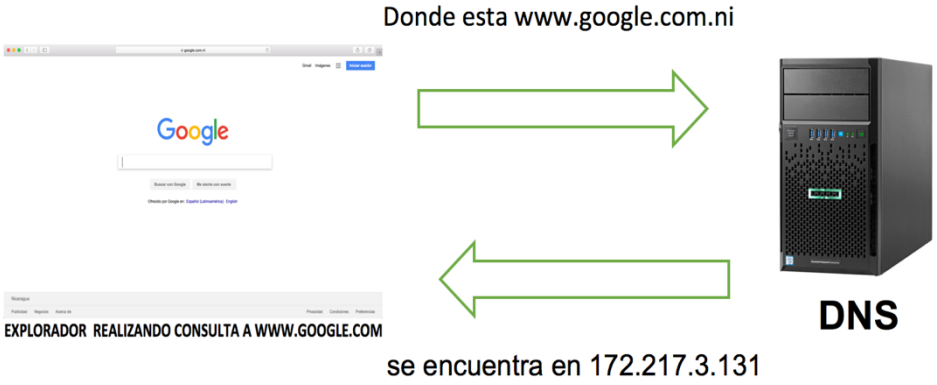

Ilustración 6. Modo de funcionamiento DNS

Suponga que un cliente ejecuta un navegador web tratando de acceder a www.google.com.ni para hacerlo necesita traducir el nombre de terminal a una dirección IP.

El lado cliente del DNS entra en funcionamiento enviando un mensaje de petición al servidor local DNS el cual dará respuesta proporcionando la información deseada al navegador web, este a su vez será capaz de acceder al recurso solicitado.

2.1.5.6 Diseño distribuido

Para lograr la funcionalidad del DNS se creó una basa de datos que pueda tener un diseño descentralizado o distribuido, el diseño centralizado sufre un gran problema de escalabilidad por lo cual se implementó el diseño distribuido que es el usado actualmente.

La escalabilidad es una de las armas fuertes del DNS, para lo cual utiliza un gran número de servidores de nombres organizados de forma jerárquica y distribuida alrededor del mundo, existen cientos de servidores distribuidos globalmente comenzando desde nuestro ISP (Internet Service Provider) hasta los trece servidores raíces ubicados en Norteamérica administrados por doce compañías distintas [9].

2.1.5.7 Resolución

Es la acción de un servidor al recibir una consulta de un resolver, y buscar en sus registros la información correspondiente a un dominio determinado, dependiendo del tipo de solicitud del resolver la consulta puede ser:

Recursiva: Un servidor que recibe una consulta recursiva debe responder con la información pedida o con un código de error. Si en sus registros no se encuentran la información deseada debería consultar a otro servidor.

Iterativa: Un servidor que recibe una consulta iterativa deberá resolver la información solicitada o en caso contrario devolver la dirección de otro servidor al cual consultar.

En la actualidad hacemos uso de una combinación de modelo recursivo e iterativo, representada en alto nivel en el siguiente esquema:

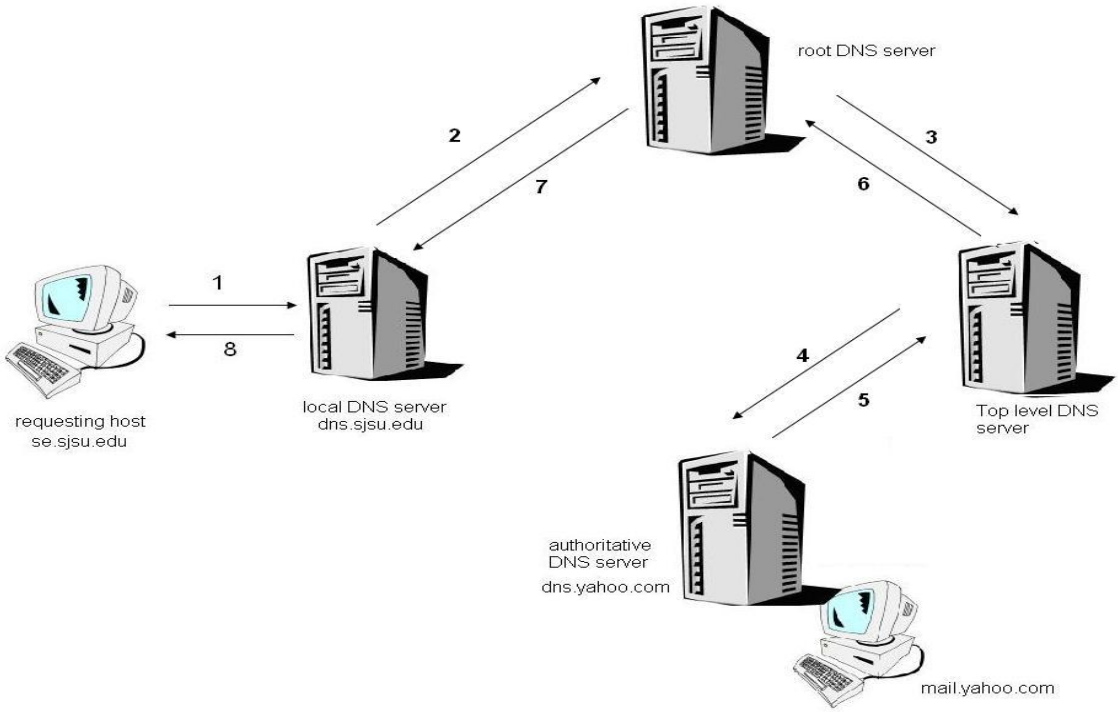

Ilustración 7. Modelo de funcionamiento recursivo e iterativo

Un tipo más de resolución no muy usada por las personas que navegan el internet, pero si por los administradores de red es la resolución inversa. El cual consiste en hallar un nombre a partir de una dirección IP, esta resolución es muy importante para administradores a la hora de interpretar ficheros log y para la autorización de algunos servicios.

2.1.5.8 Caché DNS

Los servidores actuales ampliamente utilizan el cache DNS para acelerar el proceso de resolución de nombres, todos los datos que se utilizan para hallar la dirección IP de un domino en particular, se guardan por un espacio de tiempo, de tal manera que, ante otra consulta hacia el mismo dominio, el servidor no tiene la necesidad de solicitarla de nuevo a los servidores raíces.[10]

Al espacio de tiempo en que un registro es almacenado en caché se le conoce técnicamente como TTL (Time To Live) que puede tener un tiempo máximo razonable de 48 horas pero que depende, del tipo de consistencia del dominio en sí.

2.1.5.9 Mensajes DNS

Los únicos mensajes DNS que se pueden tener son de consulta y respuesta, los cuales tienen el mismo formato que se muestra a continuación [9]:

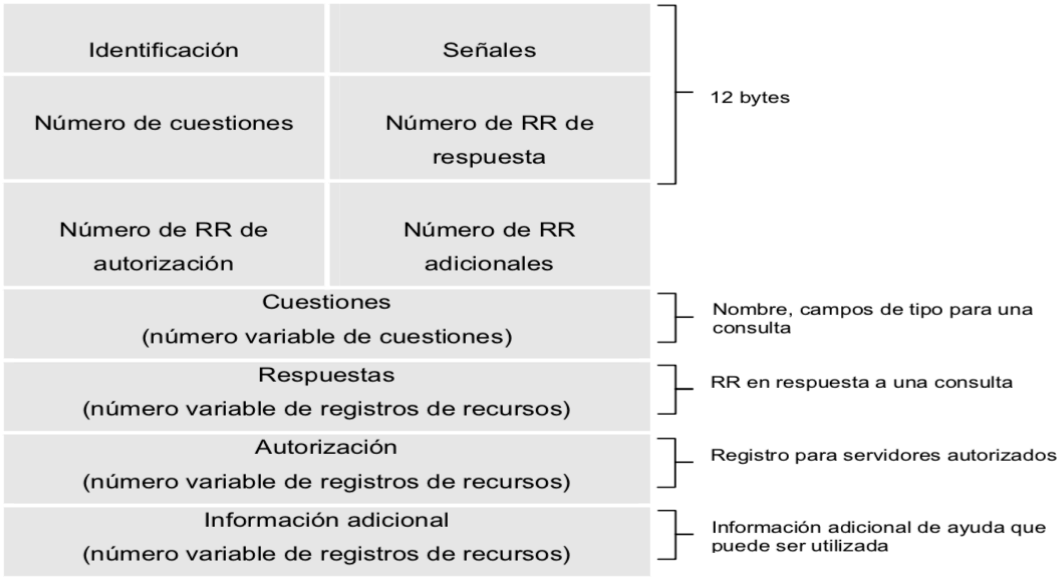

Ilustración 8. Formato consulta/ respuesta DNS

Y se dividen en:

- Sección de cabecera: Identifica la consulta, define el tipo de mensaje, identifica servidores autorizados y define el tipo de resolución.
- Sección de consultas: Contiene información asociada al tipo de consulta que se esta haciendo.
- Sección de respuestas: Contiene los registros de recursos asociados al nombre de dominio solicitado.
- Sección de autorización: Contiene registros de otros servidores de nombres autorizados.
- Sección adicional: Contiene registros de ayuda.

Sin embargo, el tipo de mensaje respuesta puede traer consigo uno o más registros de recursos (RR) asociados a un dominio en particular, los RR se encuentran formados por:

- Nombre: Indica el Dominio al que pertenece el recurso.
- Tiempo de vida (TTL): Consistencia del registro en caché.
- Clase: Define la ubicación del recurso en cuestión.
- Valor: Es el valor del recurso.
- Tipo: Define el tipo de respuesta.

De esta manera se logra la traducción de datos en las bases de datos DNS, logrando los objetivos planteados por los cual se creó.

2.1.6 SMTP

El correo electrónico es una de las aplicaciones de internet más usadas día a día, este protocolo brinda la funcionalidad de poder transferir información de un lugar a otro. [11]

El protocolo SMTP fue creado pensando en que los sistemas que intercambiarían mensajes serian grandes computadores conectados siempre a internet, sim embargo con la aparición de ordenadores personales que presentan conexiones ocasionales y envían correos a destinos remotos , solventar esta situación se volvió un gran asunto, por lo cual se desarrollaron protocolos como POP e IMAP (Protocolos de entrega final) y SMTP que se encarga de transferir mensajes desde los servidores de correo emisores a los servidores destinatarios[10].

2.1.6.1 Arquitectura

Un sistema de correo electrónico es, un sistema que esta compuesto de 2 subsistemas como son:

1) **Agentes de usuario (MUA)**:

   Los MUA es lo que les permite a las personas poder enviar y leer correos.

2) **Agentes de transferencia de mensajes (MTA)**:

   Estos sistemas son los que se encargan de transportar los mensajes del origen al destino, estos agentes de transferencias también son conocidos como **servidores de correo**.

El Agente de usuario, es un programa que proporciona una interfaz gráfica o texto que permite interactuar con el MTA, incluye los medios para poder redactar mensajes y revisar el buzón, etc.

Los agentes de transferencia de mensajes (MTA) son generalmente procesos que corren en un servidor con la intención de estar siempre disponibles y su tarea es la de entregar al destino el mensaje usando el protocolo SMTP.

El protocolo SMTP envía el mensaje sobre la conexión y devuelva el estado de entrega y los errores existentes.

2.1.6.2 Proceso de envió de un mensaje

- El usuario1 que desea enviar un mensaje al usuario2 invoca a su agente de usuario para el correo electrónico, en el que proporciona su correo electrónico, por ejemplo: (user@dominio1.com), redacta el mensaje y le indica que lo envié.
- El agente de usuario (MUA) envía el mensaje al servidor de correo (MTA) del usuario, donde es colocado en la cola de mensajes.
- El servidor de correo de este usuario ve el mensaje en cola e inicia una conexión TCP con el servidor de correo correspondiente del usuario2.
- Una vez realizada la negociación de SMTP, el servidor de usuario1 envía el mensaje al servidor del usuario2, a través de la conexión TCP.
- El servidor de correo de usuario2 recibe el mensaje y lo coloca en el buzón del usuario2.
- El usuario2 invoca a su agente de usuario (MUA) para leer el mensaje cuando el lo desee.

Un aspecto importante de SMTP es que no utiliza servidores de correo intermedios para enviar el correo, la conexión se realiza de extremo a extremo de este modo si el servidor de correo del usuario2 está fuera de servicio, el servidor correo del usuario1 conservara el mensaje y continuara intentando enviarlo nuevamente, de esta forma el mensaje no se almacena en servidores intermedios. [10]

SMTP usa conexiones persistentes, por lo cual si el MTA del cliente tiene más mensajes para enviarle a ese mismo servidor lo hará a través de la misma conexión, antes de cada transmisión los servidores de correo dialogan entre si para poder intercambiarse la información, veremos un ejemplo del establecimiento de conexión de un servidor:

Para ello iniciamos conexión con el equipo servidor con el comando telnet IP 25, denotaremos los mensajes del cliente con C: y del servidor con S:

```
S: 220 Server.localdomain ESMTP Postfix (Debian/GNU)
C: HELO prueba.com.ni
S: 250 server.localdomain
C: MAIL FROM: <usuario emisor@prueba.com.ni>
S: 250 2.1.5 OK
C: RCPT TO: <usuario receptor@server.localdomain>
S: 250 2.1.5 OK
C: DATA
S: 354 End Data with <CR><LF>.<CR><LF>
C: Esto es un mensaje de prueba
C:
S: 250 2.0.0 OK: queued as 26BDFA0
C: QUIT
S: 221 2.0.0 Bye
```

Como se puede ver antes de intercambiar los mensajes hay un proceso de negociación en el que los servidores se presentan, luego el servidor cliente le indica información sobre, quien envía el correo, a que correo está dirigido y luego el cuerpo del mensaje.

2.1.6.3 Comandos SMTP

En SMTP cuando un servidor de correo quiere comunicarse con otro, este le envía comandos para poder dialogar, cuando un servidor SMTP recibe un comando este procesa la información y devuelve una respuesta, en el ejemplo de dialogo y establecimiento de conexión en la sección anterior se muestran algunos comandos de SMTP como son:

- HELO: se encarga de iniciar el dialogo SMTP, tiene como parámetro el nombre del cliente.
- MAIL FROM: indica al servidor el inicio de un mensaje de correo y el remitente del mensaje.
- RCPT TO: indica al servidor el destinatario del mensaje, si tiene múltiples destinatarios se debe separar por coma.

- DATA: indica al servidor que lo que va a continuación es el mensaje de correo que debe llevarse al destinatario.
- QUIT: indica al servidor que el cliente no tiene más operaciones por realizar y que debería cerrar la conexión.

2.1.6.4 Respuestas SMTP

Como se mencionó en la sección anterior, cuando un servidor de correo recibe un comando este genera una respuesta, estas repuestas sirven para garantizar; la sincronización de las consultas y las acciones en el proceso de transferencia del correo, así de este modo el cliente siempre podrá conocer el estado del servidor.

Estructura de una respuesta

Una repuesta SMTP está formada por un numero de 3 dígitos y un texto, cada uno de estos dígitos les permite a los servidores tomar las decisiones adecuadas. Existen cuatro posibles valores para el primer digito de una respuesta:

1. **2xx Respuesta de terminación Positiva**:

Indica que la acción solicitada se ha completado satisfactoriamente y que se puede iniciar una solicitud nueva.

2. **3xx Respuesta intermedia Positiva**:

El comando fue aceptado, pero el servidor se encuentra a la espera de información adicional, el MTA cliente debe enviar otro comando especificando la información requerida.

3. **4xx Terminación Negativa transitoria**:

El comando no fue aceptado y la acción solicitada no se ejecutó.

Sin embargo, la condición de error es temporal, y la acción se puede solicitar nuevamente.

4. **5xx Terminación Negativa permanente**:

El comando no fue aceptado y la acción solicitada no se ejecutó, el MTA cliente no debería repetir exactamente esta solicitud.

El segundo digito de cada repuesta indica la codificación en categorías específicas, las cuales son:

1. **x0x Sintaxis**:

Estas respuestas se refieren a errores de sintaxis, comandos correctos sintácticamente que no entran en ninguna categoría funcional o comandos no implementados.

2. **x1x Información**:

Estas son respuestas a solicitudes como estatus o ayuda.

3. **x2x Conexiones**:

Son respuestas que hacen referencia al canal de transmisión

4. **x3x, x4x** Sin especificar

5. **x5x Sistema de correo**:

Estas respuestas indican el estatus del sistema de correo receptor, la transferencia solicitada o cualquier otra acción del sistema de correo.

2.1.7 IMAP

Con el nuevo orden mundial de internet en el “boom”, el correo electrónico se volvió más y más popular, acogido por una gran cantidad de usuarios alrededor de mundo, sim embargo este gran acogimiento del correo electrónico por parte de los usuarios, comenzó a hacer que se necesitara nuevas características y funcionalidades, es por eso que, el primer protocolo de entrega final optimo (en esos momentos) POP (Post Office Protocol) dio lugar a un nuevo protocolo de entrega final denominado IMAP (Protocolo de acceso a mensajes de internet), que se definió en el RFC 2060 y su última versión la encontramos en el RFC 3501. [12] Entre las nuevas funcionalidades que IMAP incluía se encuentran las siguientes:

- Descarga de mensaje bajo demanda
- Soporte de estado de mensaje en el servidor
- Soporte de acceso simultaneo a un mismo destinatario.
- Soporte de búsqueda de mensajes por parte del servidor bajo ciertos criterios.
- Soporte a extensiones del protocolo.

Independientemente de si hacemos uso de sus versiones finales POP3 o IMAP4, hay que dejar claro que ambos utilizan SMTP para poder enviar el mensaje final de un origen a un destino, pero sin duda IMAP se adapta más a las demandas de los usuarios de hoy en día al permitir tener accesos más rápidos, desde múltiples dispositivos siempre y cuando constemos con una conexión a internet. [10]

2.2 Vulnerabilidades

2.2.1 Historia de las vulnerabilidades

En los primeros años, los ataques involucraban poca sofisticación técnica. Los ataques internos se basaban en utilizar los permisos para alterar la información. Los externos se basaban en acceder a la red simplemente averiguando una clave válida. A través de los años se han desarrollado formas cada vez más sofisticadas de ataque para explotar vulnerabilidades en el diseño, configuración y operación de los sistemas.

Esto permitió a los nuevos atacantes tomar control de sistemas completos, produciendo verdaderos desastres que en muchos casos llevaron a la desaparición de aquellas organizaciones o empresas con altísimo grado de dependencia tecnológica (bancos, servicios automáticos, etc.) [13]

Estos nuevos métodos de ataque han sido automatizados, por lo que en muchos casos sólo se necesita un conocimiento técnico básico para realizarlos. El aprendiz de intruso, script-kiddie o ankle biter, o aprendiz de hacker, lamer o wannabee, tiene acceso hoy en día a numerosos programas y scripts (exploits) que se aprovechan de las vulnerabilidades, disponibles desde numerosas fuentes underground, como hacker newsgroups, mailing-lists y web sites, donde además encuentra todas las instrucciones para ejecutar ataques con las herramientas disponibles.

Bruce Schneier criptógrafo Norte Americano y especialista en privacidad y seguridad de redes, en numerosos artículos, ha definido y clasificado las generaciones de ataques en la red existentes a lo largo del tiempo:

1. **La primera generación**: Ataque físico, ataques que se centraban en los componentes electrónicos: ordenadores y cables. El objetivo de los protocolos distribuidos y de la

redundancia, es la tolerancia frente a un punto único de fallo, problemas para los que actualmente se conoce la solución.

2. **La segunda generación**: Ataque sintáctico, las pasadas décadas se han caracterizado por ataques contra la lógica operativa de los ordenadores y las redes, es decir, pretenden explotar las vulnerabilidades de los programas, de los algoritmos de cifrado y de los protocolos, así como permitir la denegación del servicio prestado. En este caso se conoce el problema, y se está trabajando en encontrar soluciones cada vez más eficaces.

3. **La tercera generación**: Ataque semántico, se basan en la manera en que los humanos asocian significado a un contenido. El hecho es que en la sociedad actual la gente tiende a creerse todo lo que lee (medios informativos, libros, la Web...). El inicio de este tipo de ataques surgió con la colocación de información falsa en boletines informativos o e-mails, por ejemplo, para beneficiarse de las inversiones dentro de la bolsa financiera. También pueden llevarse a cabo modificando información caducada. Esta generación de ataques se lleva a su extremo si se modifica el contenido de los datos de los programas de ordenador, que son incapaces de cotejar o sospechar de su veracidad, como por ejemplo la manipulación del sistema de control de tráfico aéreo, el control de un coche inteligente, la base de datos de los libros más vendidos o de índices bursátiles como el NASDAQ. Lo más curioso es que estos ataques han existido fuera del entorno informático desde hace muchos años como estadísticas manipuladas, falsos rumores..., pero es la tecnología la que potencia su difusión. Su solución pasará no sólo por el análisis matemático y técnico, sino también por el humano.

La conclusión tras el análisis de las vulnerabilidades desde un punto de vista operacional es que para evitarlas pueden definirse las tareas a realizar dentro de un sistema de seguridad en tres etapas: [18]

- Prevención: implementada por dispositivos como los firewalls.
- Detección: a través de sistemas como los IDS (Sistemas de Detención de Intrusos).
- Respuesta: las acciones a tomar deben ser dirigidas por la parte humana, típicamente los administradores de la red.

2.2.2 Vulnerabilidades genéricas

Las vulnerabilidades pretenden describir las debilidades y los métodos más comunes que se utilizan para perpetrar ataques a la seguridad de la familia de protocolos TCP/IP (confidencialidad, integridad y disponibilidad de la información). [13]

Los ataques pueden estar motivados por diversos objetivos, incluyendo fraude, extorsión, robo de información confidencial, venganza, acceso no autorizado a un sistema, anulación de un servicio o simplemente el desafío de penetrar un sistema.

Éstos pueden provenir principalmente de dos fuentes:

- **Usuarios autentificados**: al menos a parte de la red, como por ejemplo empleados internos o colaboradores externos con acceso a sistemas dentro de la red de la empresa. También denominados insiders.
- **Atacantes externos a la ubicación física de la organización**: accediendo remotamente. También denominados outsiders.

Los métodos de ataque descritos se han dividido en categorías que pueden estar relacionadas entre sí, ya que el uso de un método permite o facilita el uso de otros, en ocasiones, complementarios. Un ejemplo de ataque podría ser la realización del análisis de un sistema, mediante fingerprinting, tras el cual es posible explotar una vulnerabilidad como un buffer-overflow de un servicio TCP/IP, enviando paquetes que parecen válidos mediante IP spoofing, dentro de los métodos no se han incluido ataques de alto nivel, como por ejemplo la distribución y ejecución de virus a través del correo electrónico (protocolo SMTP), ya que afectan a vulnerabilidades particulares de las aplicaciones y los lenguajes de programación soportados por éstas. [18]

En numerosas ocasiones se ha empleado inicialmente el término inglés para nombrar la vulnerabilidad, ya que es como se conoce comúnmente, para posteriormente asociarle su posible traducción al español.

Las vulnerabilidades pueden clasificarse según dos criterios:

1. **Número de paquetes a emplear en el ataque**:

- Atomic: se requiere un único paquete para llevarla a cabo.

- Composite: son necesarios múltiples paquetes.

2. **Información necesaria para llevar a cabo el ataque**:

- Context: se requiere únicamente información de la cabecera del protocolo.
- Content: es necesario también el campo de datos o payload

2.3 Seguridad

2.3.1 Importancia de la seguridad

Debido a que el uso de Internet se encuentra en aumento [10], cada vez más compañías migran sus servicios a través de la red, por lo tanto, es fundamental saber qué recursos de la compañía necesitan protección para así controlar el acceso al sistema y el uso de los datos. [14]

Generalmente, los sistemas de información incluyen todos los datos de una compañía y los recursos de software que permiten a una compañía almacenar y hacer circular estos datos. Los sistemas de información son fundamentales para las compañías y deben ser protegidos, la seguridad informática consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto.

La seguridad es un tema amplio que cubre una multitud de aspectos. De forma breve, se define, como la encargada de garantizar que personas no autorizadas no puedan leer, o peor aún, modificar mensajes dirigidos a otros destinatarios.

La Seguridad en redes tiene el objetivo de mantener el intercambio de información libre de riesgo y proteger los recursos informáticos de los usuarios y las organizaciones. Generalmente, se encuentra amenazada por riesgos que van de la mano con el aumento del uso de Internet en las Instituciones de todos los ámbitos. De esta forma, la Seguridad en redes es la clave para conseguir la confianza de los visitantes web y está avalada por Autoridades de Confianza. [15]

2.3.2 Objetivos principales de la seguridad

Los problemas de seguridad de las redes se pueden dividir en términos generales en cuatro áreas: confidencialidad, autentificación, no repudio e integridad.

- **La confidencialidad**, consiste en mantener la información fuera del alcance de usuarios no autorizados, solo el emisor y el receptor deseado deberán ver el contenido de los mensajes, es necesario que los mensajes sean cifrados de alguna manera, de modo que, si alguien los intercepta, este no podrá ver el contenido.
- **La autentificación,** se encarga de determinar con quién se está hablando, tanto el emisor como el receptor podrán autenticar la identidad del otro.
- **El no repudio,** se encarga de las firmas: ¿cómo comprobar que su cliente en realidad hizo un pedido electrónico "tal vez argumente que él nunca realizó ningún pedido".
- **Integridad,** tiene que ver con la forma en que podemos estar seguros de que un mensaje recibido realmente fue el que se envió, y no algo que un adversario malicioso modificó en el camino.

2.3.3 Protocolo SSL

El protocolo SSL, (Secure Socket Layer), es el predecesor del protocolo TLS (Transport Layer Security). Se trata de protocolos criptográficos que proporcionan privacidad e integridad en la comunicación entre dos puntos en una red de comunicación. Esto garantiza que la información transmitida por dicha red no pueda ser interceptada ni modificada por elementos no autorizados, garantizando de esta forma que sólo los emisores y los receptores legítimos sean los que tengan acceso a la comunicación de manera íntegra. [16]

Considerando un modelo OSI, el protocolo SSL se utiliza entre la capa de aplicación y la capa de transporte. Uno de sus usos más extendidos, es el que se realiza junto al protocolo HTTP, dando lugar al HTTPS o versión segura de HTTP. Se utiliza para la transferencia de hipertexto (Sitios web) de manera segura. De esta forma se consigue que la información transmitida entre un sitio web y un usuario (en ambos sentidos), sea segura, especialmente importante cuando se trata de información sensible: datos confidenciales, contraseñas, información bancaria, imágenes personales, etc.[10]

2.3.3.1 Funcionamiento de SSL

Cuando algún usuario visita un sitio web seguro, el certificado SSL proporciona información de identificación del servidor web y establece una conexión cifrada. Este proceso ocurre, instantáneamente, en fracciones de segundo. Mientras tanto, entre el navegador y el servidor web se da la siguiente secuencia: [17]

- El navegador intenta conectarse a un sitio web con SSL y solicita la identificación del servidor.
- El servidor envía al navegador una copia de su certificado SSL.
- El navegador comprueba si es posible confiar en el certificado SSL y una vez confirmado envía un mensaje al servidor.
- El servidor emite un acuse de recibo, firmado digitalmente, para iniciar una sesión SSL cifrada.
- De esta forma, los datos encriptados se comparten entre navegador y servidor certificado.

Un certificado SSL opera como una credencial en la industria electrónica. De esta manera, realiza la identificación de un dominio específico y un servidor web. La validez de esta credencial depende de la confianza en la Autoridad de Certificación que la emitió. A su vez, las Autoridades de certificación tienen métodos para verificar la información proporcionada por las personas y organizaciones que desean adquirir un Certificado de Seguridad. [10]

3 Diseño Metodológico

La realización de este trabajo de tesis se realizó en diversas etapas, que muestran la creación de un tipo de investigación aplicada que pretende ser desarrollado en los componentes de la carrera que tengan similitud con servicios de red.

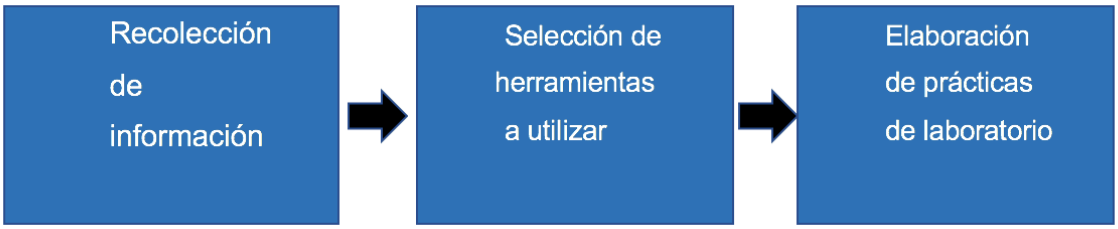

Ilustración 9. Etapas de la investigación

3.1 Recolección de información

En la primera parte del trabajo, se realizó un previo análisis sobre los servicios de red mas conocidos, con el fin de conocer los aspectos que tienen un mayor grado de importancia, organizando dicha información según el nivel de relevancia.

3.2 Selección de herramientas a utilizar

En la segunda parte se seleccionaron las plataformas y/o herramientas que se necesitan para el desarrollo de las prácticas, seleccionando las siguientes:

- VirtualBox versión 5.1.28
- Distribución Debian 8 kernel 4.9.0
- ISC-DHCP Server versión 4.3.5-3
- Bind9 versión 1:9.10.3.dfsg.P4-12.3
- Apache versión 2.4.25-3
- Dovecot versión 1:2.2.22
- Postfix versión 3.1.0-3
- Wireshark versión 2.6.3

3.3 Elaboración de prácticas de laboratorio

En este punto la información es organizada de modo que se logre una integración entre las prácticas. Para ello se ha definido el siguiente formato:

TÍTULO: Nombre de la práctica

OBJETIVOS: Presenta la visión general de lo que se espera lograr con el desarrollo de la práctica y aspectos específicos, punto de énfasis para los estudiantes.

INTRODUCCIÓN: Contiene aspectos generales introductorios a la práctica.

REQUERIMIENTOS

- **HARDWARE:** Detalla las características de la computadora que se usara en realización de la práctica.
- **SOFTWARE:** Especifica el simulador o entorno en que se desarrollará la práctica.

TOPOLOGÍA: Se presentará una imagen que muestre el diagrama de red y equipos correspondientes a la práctica.

COMANDOS DE AYUDA: Se presentará un cuadro mostrando los comandos más importantes a tener en cuenta para la realización de la práctica.

ENUNCIADO DE LA PRÁCTICA: Se definirá de forma concreta el enunciado de la práctica a realizar.

DURACIÓN DE LA PRÁCTICA: Tiempo estimado en sesiones presenciales y no presenciales para dar solución a cada práctica propuesta.

MEJORAS A IMPLEMENTAR: Contiene propuestas adicionales a incorporar a la práctica.

PREGUNTAS DE ANÁLISIS: Se presentará 3 preguntas de análisis por cada práctica que abordará tanto aspectos teóricos como prácticos.

REFERENCIAS BIBLIOGRÁFICAS: Contiene enlaces directos con información de ayuda adicional tanto en aspectos teóricos como de configuración relacionados a la práctica.

4 Desarrollo Práctico

Debido a que el trabajo presentado tiene relación con diversos componentes, el contenido del mismo deberá ser adaptado para trabajos y tareas de laboratorio trabajando de la mano con la evolución de la asignatura, las prácticas de laboratorio propuestas tienen relación con los siguientes componentes:

- Administración de Servicios de Red, se instalan, configuran y supervisan servicios de la capa de aplicación.
- Adimistracion de Servidores, se enfatiza en la administracion y mantenimiento de equipos dedicados al almacenamiento de datos.
- Seguridad de Redes, expone politicas y prácticas encargadas de prevenir y proteger el acceso no autorizado.

Programación

- Práctica 1: Configuración de servidor DHCP.
- Práctica 2: Configuración de servidor DNS.
- Práctica 3: Configuración de servidor HTTP.
- Práctica 4: Configuración de servidores de correo electrónico.
- Práctica 5: Captura de datos HTTP con Wireshark.
- Práctica 6: Observación de tramas HTTP persistentes y no persistentes.

Evaluación

La manera de evaluar quedara a criterio del docente, es común usar una evaluación parcial debido a que el documento es teórico-práctico por lo cual se recomienda dividir de la siguiente manera: 50% para los elementos teóricos y 50% para los elementos prácticos.

PRÁCTICA 1: Configuración de servidor DHCP

OBJETIVO GENERAL

- Asignar parámetros de configuración dinámicamente y aprender la configuración básica de un servidor DHCP corriendo bajo una distribución GNU/Linux (Debian)

OBJETIVOS ESPECÍFICOS

- Configurar la asignación de IP manual por dirección MAC
- Configurar pool de direcciones IP

INTRODUCCIÓN

En esta práctica se estudiará el funcionamiento del protocolo DHCP. El estudiante será capaz de simular y configurar una red física que incluya un servidor DHCP que corra bajo una distribución GNU/Linux (Debian)

DHCP: es un protocolo de red que usa el modelo cliente/servidor, el servidor DHCP posee una lista de direcciones IP dinámicas y las asigna a los clientes conforme éstas quedan disponibles, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se le ha asignado después. Así los clientes de una red IP pueden conseguir sus parámetros de configuración automáticamente. Este protocolo se publicó en octubre de 1993, y su implementación actual está en la RFC 2131, para DHCPv6 se publica el RFC 3315.

ISC-DHCP-SERVER: es un servidor el cual es una implementación libre del protocolo DHCP. Evita al administrador de red desplazarse hasta el lugar donde se encuentra ubicado el equipo del cliente en cuestión para proceder a su configuración IP y garantizando que no se cometerá un error por parte del administrador de red a la hora de configurar la IP en el equipo del cliente manualmente.

REQUERIMIENTOS

Para la realización de la práctica: configuración de un servidor DHCP, el ordenador que se destinará para realizar esta práctica debe contar con los siguientes requisitos;

HARDWARE

- Procesador mínimo de 1.5 GHz ciclos de reloj.
- Memoria RAM de 4 GB.

SOFTWARE

- VirtualBox versión 5.1.28
- Máquinas virtuales (Debian 8 kernel 4.9.0, recomendable)
- Paquete isc-dhcp-server versión 4.3.5-3

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico.

TOPOLOGÍA

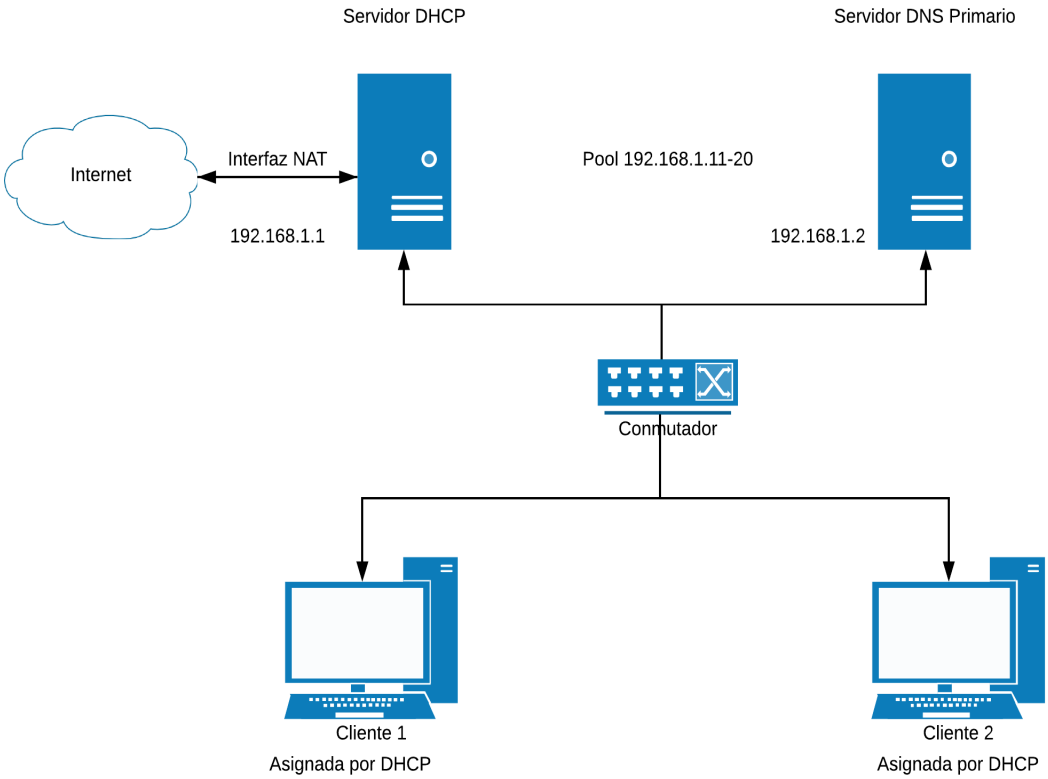

Ilustración 10. Topologia de la práctica: Configuración de servidor DHCP

COMANDOS DE AYUDA

Tabla 1.Comandos de ayuda, práctica: Configuración de servidor DHCP

Comando	Descripción
`service isc-dhcp-server start/stop/restart` `/etc/init.d/isc-dhcp-server start|stop|restart`	Arranca, detiene o reinicia el servidor DHCP.
`apt-get install isc-dhcp-server`	Instala el servidor DHCP (isc-dhcp-server)
`nano /etc/dhcp/dhcpd.conf`	Edita archivo de configuración del servidor DHCP.
`nano /etc/default/isc-dhcp-server`	Edita archivo de configuración, que establece las opciones para ejecutar el servidor DHCP.
`nano /etc/network/interfaces`	Archivo de configuración de interfaces de red.
`service isc-dhcp-server status` `/etc/init.d/isc-dhcp-server status`	Muestra el estado en el que se encuentra el servidor DHCP.
`/etc/sysctl.conf`	Activa el enrutamiento en una maquina Linux, se modifica la directiva net.ipv4.ip_forward = 1.
`ifconfig`	Muestra información acerca las interfaces de red.
`iptables –t nat –A POSTROUTING –s 'IP' –o eth0 –j MASQUERADE`	Aplica NAT sobrecargado en la interfaz indicada por el argumento -o para la dirección de red definida por el argumento -s.

Parámetros en el archivo /etc/dhcp/dhcpd.conf y un parametro del archivo /etc/default/isc-dhcp-server (INTERFACES)

Tabla 2.Parámetros de ayuda, práctica: Configuración de servidor DHCP

Directiva	Descripción
`subnet`	Se utiliza para indicar un segmento de red, especificando la dirección de la subred y su máscara de red.
`range`	Especifica el rango o pool de direcciones IP que se brindara a los clientes cuando estos la soliciten.
`option domain-name-servers`	Indica los servidores de nombres de dominio que serán usados para el segmento de red
`default-lease-time`	Se utiliza para indicar el tiempo de concesión de una IP cuando el cliente en su solicitud no indica ningún tiempo. Tiempo en segundos.
`max-lease-time`	Se utiliza para indicar el máximo tiempo de concesión de una IP, si un cliente solicitara una concesión por encima de este tiempo, se le asignaría el máximo. Tiempo en segundos.
`host`	Definir configuración para un host especifico.
`hardware`	Usado en conjunto con la directiva host, especifica a un equipo a través de su dirección física.
`fixed-address`	Usado en conjunto con la directiva host, define la asignación manual de la dirección IP.
`INTERFACES`	Directiva localizada en /etc/default/isc-dhcp-server indica las interfaces de red por las cuales el servidor DHCP asignara parámetros de configuración.

Ejemplo de /etc/dhcp/dhcpd.conf

```
subnet 10.20.10.0 netmask 255.255.255.0
{
  range 10.20.10.20 10.20.10.30;
  option domain-name-servers 10.20.10.2;
  max-lease-time 7200;
  host Servidor
  {
```

```
    hardware ethernet **:**:**:**:**:**;
    fixed-address 10.20.10.2;
  }
}
```

Ejemplo de /etc/default/isc-dhcp-server

```
INTERFACES="eth0 eth1 eth2 wlan0
```

Datos de los dispositivos

Tabla 3.Datos de los dispositivos, práctica: Configuración de servidor DHCP

Nombre de equipo	**IP**
Servidor_DHCP	192.168.1.1 (FIJA)
Servidor_DHCP(interfaz puente)	Configurar modo puente desde virtualbox
Servidor_DNS	192.168.1.2 (FIJA)
Pool dinámico del servidor DHCP	192.168.1.11 a 192.168.1.20
Cliente1	Obtener IP mediante DHCP
Cliente2	Obtener IP mediante DHCP

ENUNCIADO

Asignar IP de manera dinámica y fija por medio del servidor DHCP a los siguientes equipos;

- Servidor_DNS
- Cliente1
- Cliente2

Las direcciones se facilitan en el recuadro (Datos de los dispositivos). En el dispositivo Servidor_DHCP se configurará la IP de manera estática en su interfaz de red; y se configurará la segunda interfaz en modo NAT y activaremos el enrutamiento, para que este funcione como Gateway a internet.

En el dispositivo Servidor_DNS se configurará su interfaz en modo DHCP, que reciba del servidor DHCP la dirección fija asignada a este equipo. identificándose ante el servidor por su dirección MAC.

En el dispositivo Cliente1 y Cliente2 se configurará su interfaz en modo DHCP, obtendrá un IP del pool configurado en el servidor DHCP.

TIEMPO ESTIMADO DE SOLUCIÓN

- 4 horas.

MEJORAS A IMPLEMENTAR

- Configurar al pool de direcciones, que el tiempo de concesión de las IP, tengan como tiempo máximo de concesión 30 segundos.
- Apagar la maquina Servidor_DHCP. Hacer una prueba de ping, de Cliente1 a Cliente2.

PREGUNTAS DE ANÁLISIS

- ¿Qué sucedería si en la red existe más de un servidor DHCP?
- ¿Bajo qué criterio el cliente decide de que servidor obtener su configuración?
- ¿Puede forzar el cliente DHCP al servidor a que le asigne una dirección determinada?

REFERENCIAS BIBLIOGRÁFICAS

R. Woundy, Dynamic Host Configuration Protocol, RFC 4388 [online]. Disponible en: https://tools.ietf.org/html/rfc4388

Alberto Molina, NAT con iptables [online]. Disponible en: https://albertomolina.wordpress.com/2009/01/09/nat-con-iptables/

ISC, Archivo de Configuración DHCPD [online]. Disponible en: https://www.isc.org/wp-content/uploads/2017/08/dhcp41conf.html

PRÁCTICA 2: Configuración de servidor DNS

OBJETIVO GENERAL

- Aprender a configurar un servidor DNS

OBJETIVOS ESPECÍFICOS

- Configurar correctamente los dominios
- Explicar parámetros de configuración para BIND9.

INTRODUCCIÓN

En esta práctica estudiaremos el funcionamiento del protocolo DNS. El estudiante será capaz de simular y configurar una red física que incluya un servidor DNS que corra bajo una distribución GNU/Linux basándose en la topología de la práctica anterior de DHCP.

DNS: por sus siglas en inglés, Domain Name System. Su función más importante es "traducir" nombres inteligibles (Nombres de dominiospara las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos.

BIND9: por sus siglas en ingles Berkeley Internet Name Domain, es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un estándar.

REQUERIMIENTOS

Para la realización de la práctica: configuración de servidor DNS, el ordenador que se destinará para realizar esta práctica debe contar con los siguientes requisitos:

HARDWARE

- Procesador mínimo de 1.5 GHz ciclos de reloj.
- Memoria RAM de 4 GB.

SOFTWARE

- VirtualBox versión 5.1.28
- Máquinas virtuales (Debian 8 kernel 4.9.0, recomendable)
- Paquete Bind9 versión 1:9.10.3.dfsg.P4-12.3

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico.

TOPOLOGÍA

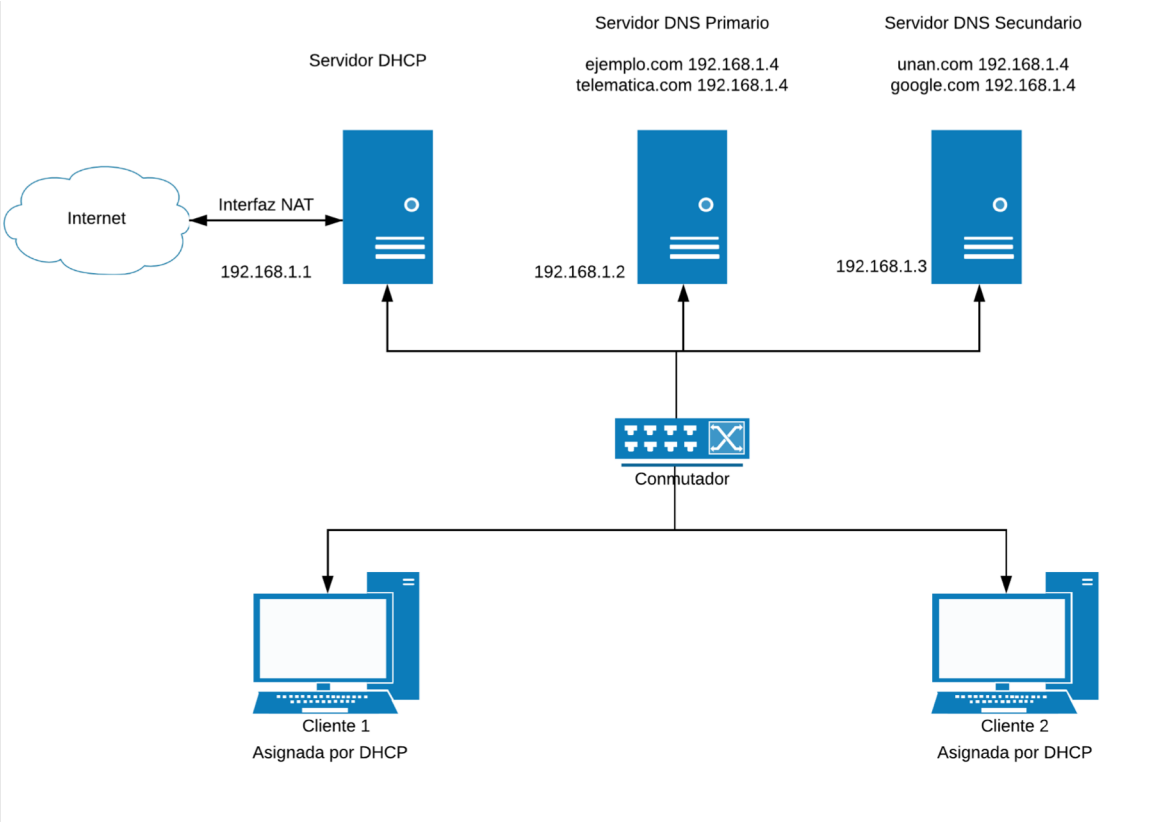

Ilustración 11. Topología de la práctica: Configuración de servidor DNS

COMANDOS DE AYUDA

Tabla 4. Comando de ayuda, práctica: Configuración de servidor DNS

Comando	Descripción
`service bind9 start/stop/restart` `/etc/init.d/bind9 start|stop|restart`	Arranca, detiene o reinicia el servidor bind9.
`apt-get install bind9`	Instala el servidor DNS (bind9)
`nano /etc/bind/named.conf.local`	Edita archivo de configuración del servidor DNS (bind9).
`nano /etc/bind/named.conf.options`	Edita archivo de configuración, que establece las opciones para ejecutar el servidor DNS (bind9).
`nano /etc/network/interfaces`	Edita archivo de configuración, que establece los valores de las interfaces de red, ya sean cableadas e inalámbricas.
`service bind9 status` `/etc/init.d/bind9 status`	Muestra el estado en el que se encuentra el servido DNS.
`ifconfig`	Muestra información acerca las interfaces de red.

Directivas en el archivo /etc/bind/named.conf.local

Tabla 5. Directivas del archivo, práctica: Configuración de servidor DNS

Directiva	Descripción
`zone`	Define una nueva zona.
`type`	Define el tipo de zona, ya sea maestro o esclavo (master o slave)
`file`	Indica cual es el archivo que contiene la configuración de dicha zona.
`masters`	Indica que será el servidor DNS maestro que contiene la traducción de dicha zona.
`slave`	Indica que hará una petición para aprender la zona. Al servidor maestro descrito, en la directiva masters.
`allow-transfer`	Lista de direcciones IP, a las que se le permite copiar archivos de zonas desde el servidor master o slave.

Ejemplo /etc/bind/named.conf.local para el servidor primario master

```
zone "ejemplo.com"{
   type master;
   file "/etc/bind/db.ejemplo.com";
   allow-transfer {192.168.1.3;};
}
```

Ejemplo /etc/bind/named.conf.local para el servidor secundario slave

```
zone "ejemplo.com" {
   type slave;
   file "/etc/bind/db.ejemplo.com";
   masters {192.168.1.2};
}
```

Ejemplo /etc/bind/db.ejemplo.com

```
$TTL  604800
@     IN    SOA   ns1.ejemplo.com. root.ejemplo.com. (
                        2         ; Serial
                   604800         ; Refresh
                    86400         ; Retry
                  2419200         ; Expire
                   604800)   ; Negative Cache TTL
;
IN     NS   ns1.ejemplo.com.
ns1    IN   A     127.0.0.1
www    IN   A     127.0.0.1
ftp    IN   A     127.0.0.1
```

Configuración del servidor DNS primario

Tabla 6. Servidor primario, práctica: Configuración de servidor DNS

Nombre de dominio	IP
ejemplo.com	192.168.1.4
telematica.com	192.168.1.4
unan.com	192.168.1.4
google.com	192.168.1.4

ENUNCIADO

En esta práctica se pretende, configurar dos servidores DNS, los cuales obtendrán IP por medio del servidor DHCP, el servidor DNS primario obtendrá la IP 192.168.1.2 y servidor DNS secundario obtendrá la IP 192.168.1.3.

El servidor DNS primario se configurará con respecto al recuadro de configuración anteriormente descrito, se configurará como maestro en todas las zonas, permitiendo la transferencia de zonas hacia el servidor DNS secundario. El servidor DNS secundario se configurará para que "aprenda" solamente las zonas configuradas en el servidor primario ejemplo.com y telematica.com.

El servidor DNS primario responderá las peticiones de los clientes usando el comando nslookup o dig, posteriormente se apagará o desconectará la interfaz del servidor DNS primario, para que el servidor DNS secundario atienda las peticiones de los clientes. Compruebe que solo responderán las zonas ejemplo.com y telematica.com

TIEMPO ESTIMADO DE SOLUCIÓN

- 8 horas.

MEJORAS A IMPLEMENTAR

- Configurar al servidor DNS primario un acl, que permita peticiones solamente del Cliente1, las acls se configuran en el archivo /etc/named.conf, ejemplo de una acl:

```
acl-pass{
    192.168.1.3/32;
};
acl-block{
   192.168.4/32;
}
options{
   allow-query{acl-pass;};
   blackhole{acl-block;}
}
```

- Configurar las trasferencias de zonas seguras del servidor DNS primario al secundario.

PREGUNTAS DE ANÁLISIS

- ¿Explique el comportamiento de DNS cuando un mensaje es igual o mayor a 512 Bytes? Por ejemplo, cuando se realiza la transferencia de zonas entre servidores.
- ¿Explique y mencione los tipos de servidores DNS?
- ¿Explique por qué motivo se implementa un acl en un servidor DNS?

REFERENCIAS BIBLIOGRÁFICAS

P. Mockapetris, System Domain Names, RFC 1035 [online]. Disponible en: https://tools.ietf.org/html/rfc1035

ISC, Configuración de Referencias BIND9 [online]. Disponible en: https://ftp.isc.org/isc/bind9/9.9.7rc1/doc/arm/Bv9ARM.ch06.html#id2574035

Eduardo Lara, Guia Práctica Sistemas de Nombres de Dominios [online]. Disponible en: http://personals.ac.upc.edu/elara/documentacion/INTERNET%20-%20UD6%20-%20DNS.pdf

James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.a. Ed. Pearson, 2017, pp. 104-115.

PRÁCTICA 3: Configuración de servidor HTTP

OBJETIVO GENERAL

- Configurar de manera básica un servidor HTTP con sus respectivos hosts virtuales, corriendo bajo una distribución GNU/Linux (Debian).

OBJETIVOS ESPECÍFICOS

- Configurar hosts virtuales.

INTRODUCCIÓN

Para esta práctica se estudiará el funcionamiento del protocolo HTTP. El estudiante será capaz de simular y configurar un servidor HTTP que corra bajo una distribución GNU/Linux basándose en la topología de la práctica anterior de DNS.

El Protocolo de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol o HTTP) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web, la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software en la web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores, para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente, esto les permite a las aplicaciones web instituir la noción de sesión, y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

Servidor web

La definición más sencilla de servidor web; es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, baners, etc), estos servidores web utilizan el protocolo HTTP.

Apache

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento.

REQUERIMIENTOS

Para la configuración del servidor Apache, el computador que se destinará para realizar esta práctica debe contar con los siguientes requisitos:

HARDWARE

- Procesador mínimo de velocidad de 1.5 GHz
- Memoria RAM de 4 GB.

SOFTWARE

- VirtualBox versión 5.1.28
- Máquinas virtuales (Debian 8 kernel 4.9.0, recomendable)
- Paquete apache2 versión 2.4.25-3
- Paquete php5 version 5.6.38
- Librería ibapache2-mod-php5 version 5.6.38

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico. Si se desea hacer peticiones al servidor web desde un navegador, puede dejar las máquinas virtuales de los clientes en modo gráfico.

TOPOLOGÍA

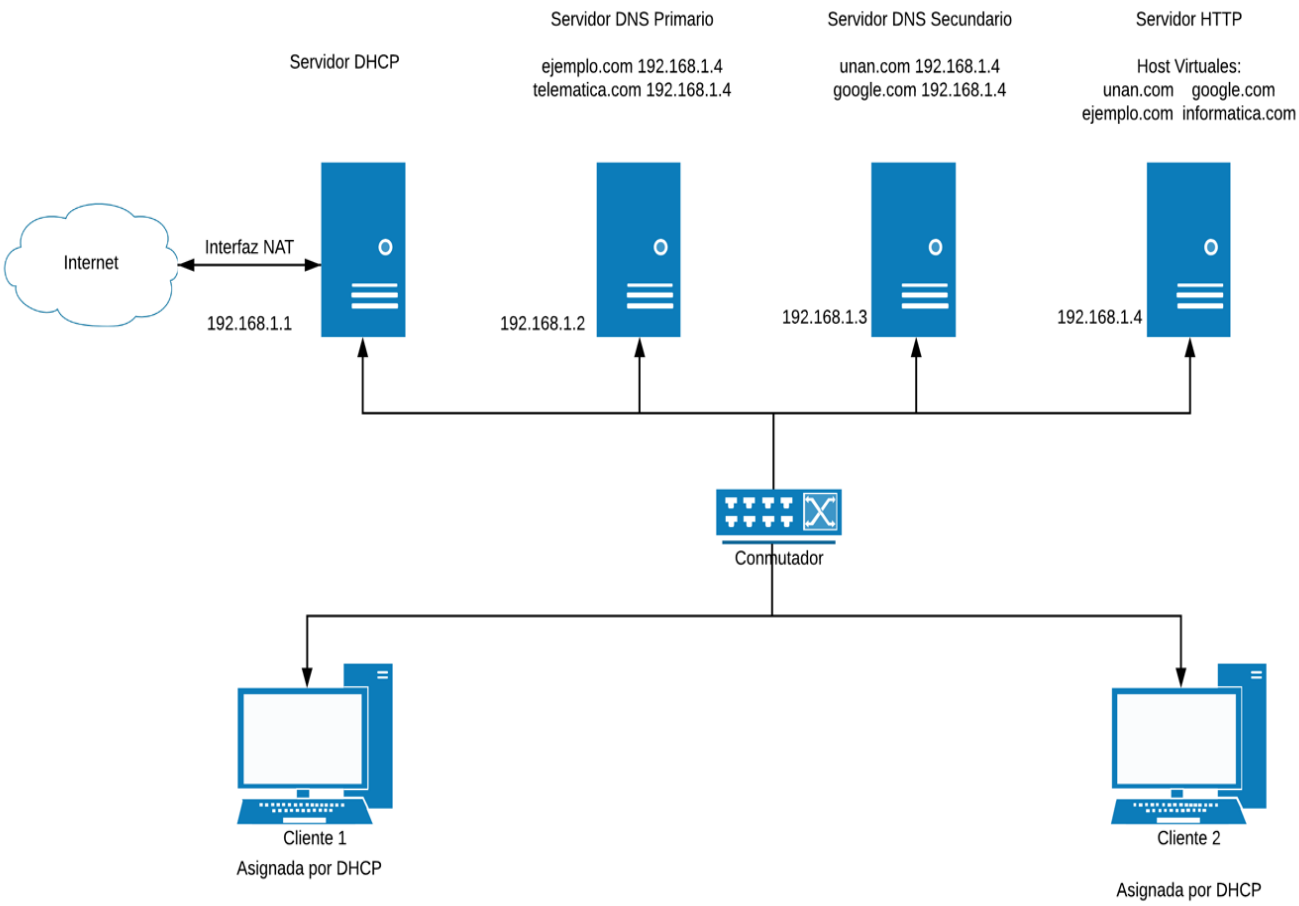

Ilustración 12. Topología de la práctica: Configuración de servidor HTTP

COMANDOS DE AYUDA

Tabla 7. Comandos de ayuda, práctica: Configuración de servidor HTTP

Comando	Descripción
`service apache2 start/stop/restart` `/etc/init.d/apache2 start|stop|restart`	Arranca, detiene o reinicia el servidor apache.
`apt-get install apache2`	Instala el servidor apache
`nano /etc/apache2/apache2.conf`	Archivo de configuración de apache
`nano /etc/apache2/ports.conf`	Indica en que puertos TCP estará escuchando el

	servidor apache.
`service apache2 status` `/etc/init.d/apache2 status`	Muestra el estado en el que se encuentra el servidor apache.
`ifconfig`	Muestra información acerca las interfaces de red.
`nano /etc/network/interfaces`	Archivo de configuración de interfaces de red.
`a2ensite`	Activa un virtual host.

Directivas en el archivo /etc/apache2/sites-available/www.ejemplo.com.conf

Tabla 8. Directivas de archivo, práctica: Configuración de servidor HTTP

Directiva	Descripción
`<VirtualHost>`	Etiqueta de definición para un virtual host
`serverAdmin`	Especifica la dirección de correo del administrador del servidor
`documentRoot`	Especifica el archivo web que se retornara para el respectivo host virtual.
`serverName`	Especifica el nombre de dominio, al cual el host virtual responderá.
`directoryIndex`	Es la página servida por defecto por el servidor cuando un usuario solicita el recurso principal (/) de un dominio.

Ejemplo de la configuración de directivas para crear un host virtual

```
<VirtualHost *:80>
    ServerAdmin ejemploAdmin@ejemplo.com
    DocumentRoot /var/www/html/ejemplo
    DirectoryIndex index.html
    ServerName www.ejemplo.com
</VirtualHost>
```

ENUNCIADO

En esta práctica se pretende, configurar un servidor HTTP el cual obtendrá IP por medio del servidor DHCP, obtendrá la IP 192.168.1.4.

Hosts virtuales a configurar

Tabla 9. Hosts virtuales, práctica: Configuración de servidor HTTP

Nombre de dominio
ejemplo.com
telematica.com
unan.com
google.com

El servidor HTTP, se configurará con respecto a los recuadros de configuración anteriormente descritos. El servidor HTTP responderá las peticiones de los clientes (Cliente1 y Cliente2), con respecto a los dominios configurados en los hosts virtuales definidos en la práctica. Se debe de instalar mysql-server versión 5.5.9999, phpmyadmin versión 5.6.38 (incluir libapache2-mod-php5 versión 5.6.38) y wordpress, para comprobar que todo esté configurado correctamente.

Acciones a realizar por cada host virtual:

Tabla 10, Página por defecto por Host virtual, práctica: Configuración de servidor HTTP

Host virtual	**Página de inicio por defecto**
ejemplo.com	PHPMyAdmin
telematica.com	index.php de apache.
unan.com	Crear una pequeña página web con el escudo de la universidad y un párrafo, “A la libertad de nuestra Universidad”.
google.com	Instancia de WordPress

TIEMPO ESTIMADO DE SOLUCIÓN

- 8 horas.

MEJORAS A IMPLEMENTAR

- Habilitar los flags HTTPOnly & Secure para permitir el uso de cookies al servidor http.
- Explique que beneficios nos aporta el uso de cookies.

PREGUNTAS DE ANÁLISIS

- Mencione los modelos de gestión de conexiones de HTTP, Detalle las ventajas y desventajas de cada uno de ellos.
- ¿Por qué el protocolo HTTP es sin estado?
- ¿Como se relaciona un VirtualHost con una solicitud HTTP?

REFERENCIAS BIBLIOGRÁFICAS

T.Berners-Lee, R. Fielding. Hypertext Transfer Protocol, RFC 2616 [online]. Disponible en: https://www.ietf.org/rfc/rfc2616.txt

Apache, Archivos de Configuración HTTP [online]. Disponible en: https://httpd.apache.org/docs/2.4/configuring.html

James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.a. Ed. Pearson, 2017, pp. 81-95.

PRÁCTICA 4: Configuración de servidores de correo electrónico

OBJETIVO GENERAL

- Configurar servidor SMTP y IMAP

OBJETIVOS ESPECÍFICOS

- Configurar correctamente los servidores de transporte y acceso, de correo electrónico.

INTRODUCCIÓN

El protocolo para transferencia simple de correo (en inglés Simple Mail Transfer Protocol o **SMTP**), es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (teléfonos móviles, impresoras, etc).

Fue definido inicialmente en agosto de 1982 por el RFC 821 (para la transferencia) y el RFC 822 (para el mensaje). Son estándares oficiales de Internet que fueron reemplazados respectivamente por el RFC 2821 y el RFC 2822, que a su vez lo fueron por el RFC 5321 y el RFC 5322

El protocolo de acceso a mensajes de Internet (en inglés Internet Message Access Protocol o **IMAP**), es un protocolo de la capa de aplicación que permite el acceso a mensajes almacenados en un servidor de correo en Internet, mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet, es posible especificar carpetas del lado del servidor, por otro lado, es un protocolo complejo.

Dovecot es un servidor de correo electrónico de código abierto IMAP y POP3 para sistemas Linux / UNIX, escrito principalmente con la seguridad en mente, dovecot es una excelente opción para instalaciones pequeñas y grandes, es rápido, fácil de configurar, no requiere una administración especial y usa muy poca memoria.

Postfix es un servidor de correo de software libre / código abierto, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado sendmail. Anteriormente conocido como VMailer e IBM Secure Mailer, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente. Postfix es el agente de transporte por omisión en diversas distribuciones de Linux.

Postfix es un Agente de transferencia de correo (MTA), es decir, un software que envía y recibe correos electrónicos desde y hacia otras computadoras en la red usando el protocolo simple de transferencia de correo (SMTP). Desde el punto de vista de un cliente de correo electrónico, POP / IMAP son los protocolos utilizados para recibir mensajes, y SMTP se usa para enviar. Sin embargo, no es cierto que "POP / IMAP = recibir" y "SMTP = enviar": los servidores de correo electrónico usan SMTP para intercambiar mensajes entre ellos, es decir, tanto el envío como la recepción. Lo que es correcto es que:

POP / IMAP son utilizados por un cliente para leer mensajes de un servidor de correo electrónico; SMTP se usa para intercambiar correos electrónicos entre computadoras.

REQUERIMIENTOS

Para la realización de la práctica se necesitaron los siguientes requisitos:

HARDWARE

- Procesador mínimo de velocidad de 1.5 GHz
- Memoria RAM de 4 GB.

SOFTWARE

- VirtualBox versión 5.1.28
- Máquinas virtuales (Debian 8 kernel 4.9.0, recomendable)
- Paquete postfix versión 3.1.0-3
- Paquete dovecot-common versión 1:2.2.22
- Paquete dovecot-imapd versión 1:2.2.22

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico. Si se desea hacer peticiones al servidor web desde un navegador, puede dejar las máquinas virtuales de los clientes en modo gráfico.

TOPOLOGÍA

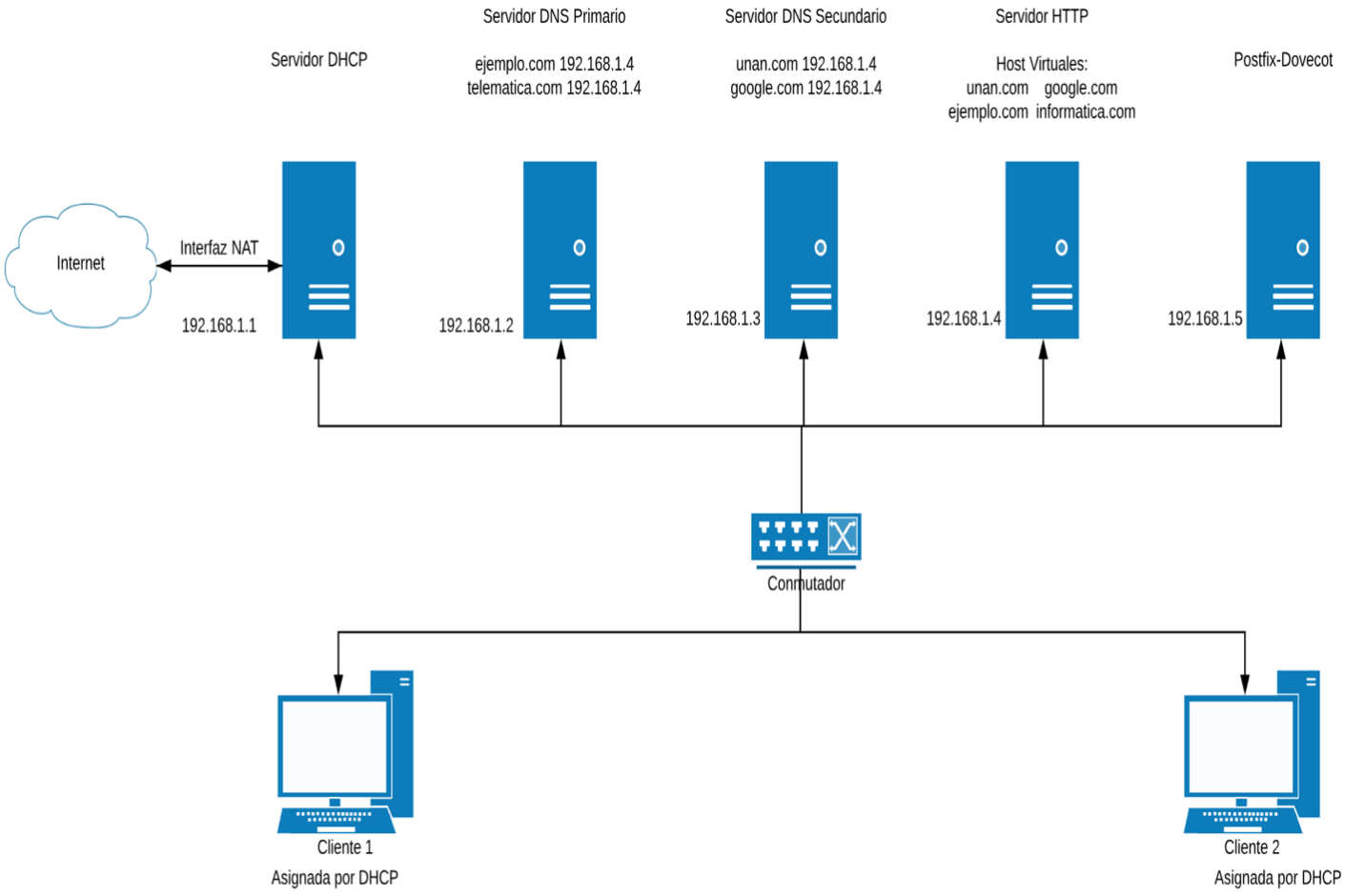

Ilustración 13. Topología de la práctica: servidores de correo electrónico.

COMANDOS DE AYUDA

Tabla 11. Comando de ayuda, práctica: servidores de correo electrónico.

Comando	Descripción
`service postfix start/stop/restart /etc/init.d/postfix start/stop/restart`	Arranca, detiene o reinicia el servidor postfix.
`service dovecot start/stop/restart /etc/init.d/dovecot start/stop/restart`	Arranca, detiene o reinicia el servidor dovecot.
`apt-get install postfix dovecot-common dovecot-imapd`	Instala el servidor postfix y dovecot
`sudo postconf -e "directiva = parametro"`	Agrega configuración al archivo global del servidor postfix (/etc/postfix/main.cf)

Tabla 12. Directivas en el archivo /etc/postfix/main.cf

Directiva	Descripción
`mydomain`	Establece el dominio que se tomara, como referencia al servidor SMTP.
`mynetworks`	Indican las IP desde las que pueden enviarse mensajes.
`mydestination`	Especifica que dominios entregar localmente, en vez de enviarlo a otras maquinas
`inet_interfaces`	Indica las interfaces de red, en las que el postfix recepciónará los mensajes.
`home_mailbox`	Indica el directorio del usuario del sistema donde se almacenarán los mensajes.

Tabla 13 Directivas en el archivo /etc/dovecot/dovecot.conf

Directiva	Descripción
`protocols`	Indica que protocolo será utilizado (POP3 IMAP)
`auth default`	Tiene varios argumentos, entre ellos los más destacados: mechanisms, socket listent. socket listen, recibe client y master como parámetros, estos reciben como parámetros el modo de acceso que tendrán al directorio de autenticación, con que usuario y con qué grupo se abrirá.

Tabla 14. Directivas en el archivo /etc/dovecot/conf.d/10-mail.conf

Directiva	Descripción
mail_location	Indica el directorio en el cual serán almacenados los mensajes.

Tabla 15. Configuración en el DNS primario

Dominio	Descripción
mail.unan.com	Este dominio ya está configurado, se configuro en la práctica de servicio de DNS, se le agregara mail.unan.com al dominio existente, con la IP 192.168.1.5

Ejemplo de configuración de directivas para el archivo /etc/postfix/main.conf

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
readme_directory = no
compatibility_level = 2
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database=btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database=btree:${data_directory}/smtp_scache
smtpd_relay_restrictions=permit_mynetworks    permit_sasl_authenticated
defer_unauth_destination
myhostname = telematica.ejemplo.com
mydomain=ejemplo.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $mydomain
mydestination  =  ejemplo.com,  $myhostname,  telematica.ejemplo.com,
localhost.telematica.com, localhost
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
home_mailbox = Maildir/
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = loopback-only
default_transport = error
relay_transport = error
inet_protocols = all
```

Ejemplo de configuración de directivas para el archivo /etc/dovecot/dovecot.conf

```
!include_try /usr/share/dovecot/protocols.d/*.protocol
listen = *
dict
{
  #quota = mysql:/etc/dovecot/dovecot-dict-sql.conf.ext
  #expire = sqlite:/etc/dovecot/dovecot-dict-sql.conf.ext
}
!include conf.d/*.conf
!include_try local.conf
protocols = "imap"
auth default
{
  mechanisms = plain login
  socket listen
  {
    client
    {
       path = /var/spool/postfix/private/auth-client
       mode = 0660
       user = postfix
       group = postfix
    }
  }
}
```

Ejemplo de configuración de directivas para el archivo /etc/dovecot/conf.d/10-auth.conf

```
disable_plaintext_auth = no
auth_mechanisms = plain login
```

Ejemplo de configuración de directivas para el archivo /etc/dovecot/conf.d/10-mail.conf

```
mail_location = maildir:~/Maildir
mail_location = mbox:~/mail:INBOX=/var/mail/%u
namespace inbox {inbox = yes}
```

ENUNCIADO

En esta práctica se pretende, configurar un servicio de correo electrónico conformado por dos servidores Postfix y Dovecot, el cual obtendrá IP por medio del servidor DHCP, obtendrá la IP 192.168.1.5.

El servidor Postfix se configurará para recibir peticiones mediante el dominio indicado en el apartado anterior. También se configurará para que trabaje con los usuarios del sistema usando como cliente de correo Thunderbird.

Configurar el cliente web Roundcube para enviar y recibir emails, para lo cual deberá configurar un host virtual en apache2, una zona con bind9, por último, configurar los permisos respectivos y crear un usuario en MySQL para que el cliente web Rouncube funcione adecuadamente.

El servidor Dovecot se configurará con el protocolo de transporte IMAP.

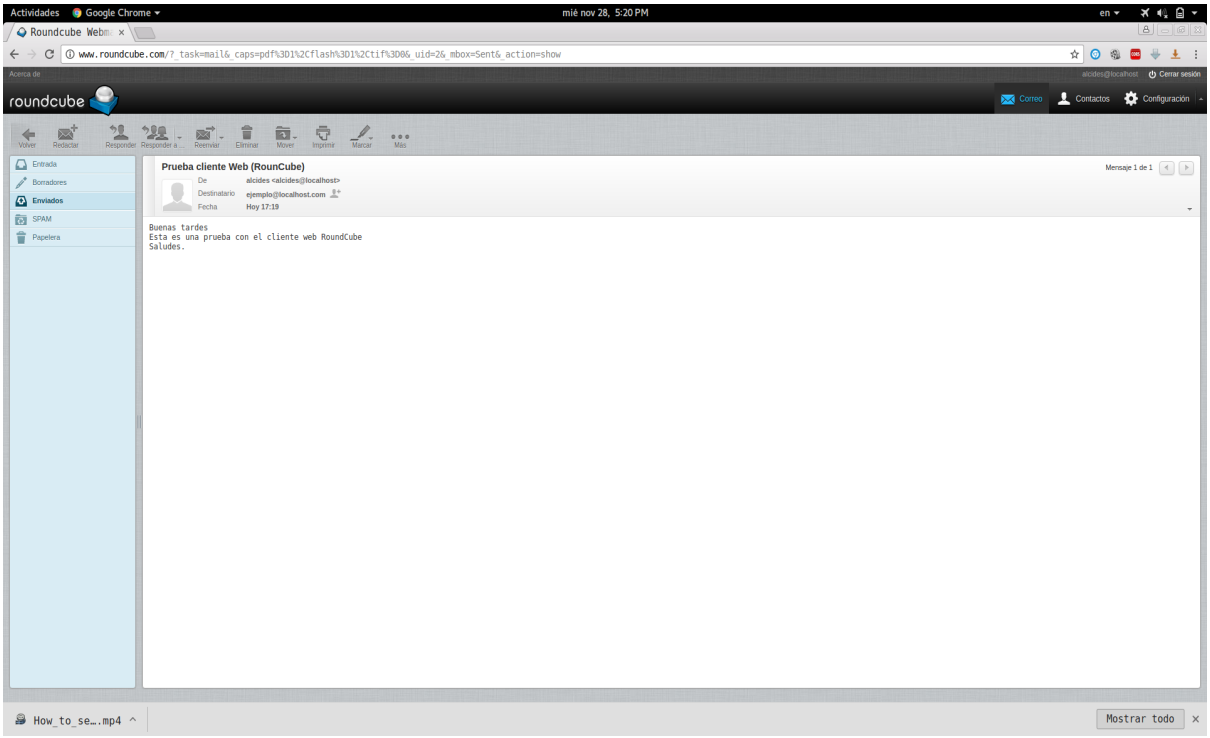

Ilustración 14. Envió de correo por Roundcube, práctica 4.

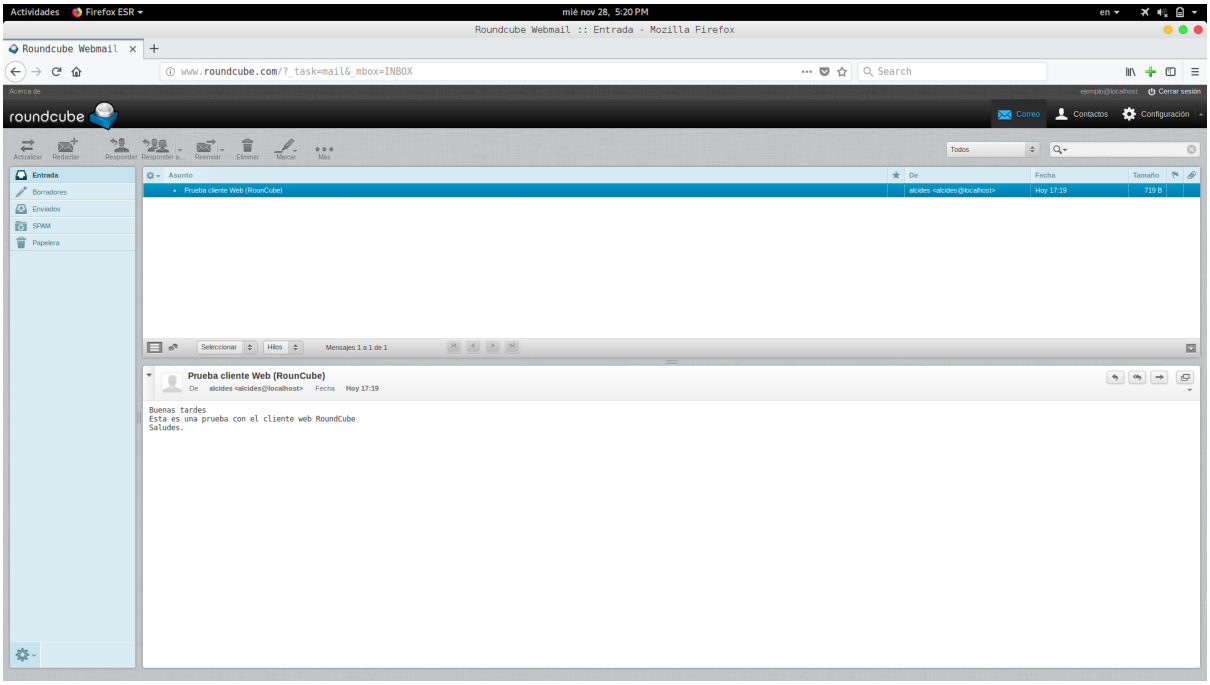

Ilustración 15. Recepción de correo por Roundcube, práctica 4.

TIEMPO ESTIMADO DE SOLUCIÓN

- 8 horas.

MEJORAS A IMPLEMENTAR

- Configurar al servidor Dovecot con el protocolo de transporte POP3

PREGUNTAS DE ANÁLISIS

- ¿Qué puertos predeterminados utilizan los protocolos IMAP, SMTP Y POP3?
- ¿Explique cuál es la diferencia de los protocolos IMAP y POP3?
- ¿Mencione 2 ventajas y desventajas de IMAP y POP3?

REFERENCIAS BIBLIOGRÁFICAS

J. Klensin, Simple Mail Transfer Protocol, RFC 5321 [online]. Disponible en: https://tools.ietf.org/html/rfc5321

A. Gulbrandsen,. Internet Message Access Protocol, RFC 6851 [online]. Disponible en: https://tools.ietf.org/html/rfc6851

Charlie Sanchez, Configuración de Posfix y Dovecot [online]. Disponible en: https://www.charliejsanchez.com/2018/02/06/instalar-configurar-postfix-dovecot-parte-1/

James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.a. Ed. Pearson, 2017, pp. 96-103.

PRÁCTICA 5: Captura de datos HTTP con Wireshark

OBJETIVO GENERAL

- Observar datos o mensajes HTTP

OBJETIVOS ESPECÍFICOS

- Ver la vulnerabilidad que posee el protocolo HTTP
- Aplicar HTTPS a todos los hosts virtuales.

INTRODUCCIÓN

El protocolo HTTP es un protocolo sin estado, muy utilizado en la actualidad, pero el protocolo HTTP no fue diseñado inicialmente para ser seguro es por eso que habilitaremos el modo seguro del protocolo HTTP el cual es HTTPS, ya que es uno de los protocolos más usados de internet,se abordará en esta práctica la importancia de la seguridad sobre el protocolo.

Wireshark es un analizador de protocolos open-source que actualmente está disponible para plataformas Windows y Unix. Su principal objetivo es el análisis de tráfico, pero además es una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red. Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos soportados actualmente y todo ello por medio de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados.

REQUERIMIENTOS

Para la práctica: captura de datos HTTP con Wireshark, el ordenador que se destinará para realizar esta práctica debe contar con los siguientes requisitos:

HARDWARE

- Procesador mínimo de velocidad de 1.5 GHz
- Memoria RAM de 4 GB.

SOFTWARE

- VirtualBox versión 5.1.28
- 6 máquinas virtuales (Debian 8 kernel 4.9.0, recomendable)
- Wireshark versión 2.6.3

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico. Si se desea hacer peticiones al servidor web desde un navegador, puede dejar las máquinas virtuales de los clientes en modo gráfico y detener los servicios de Postfix y Dovecot, ya que no se utilizarán.

TOPOLOGÍA

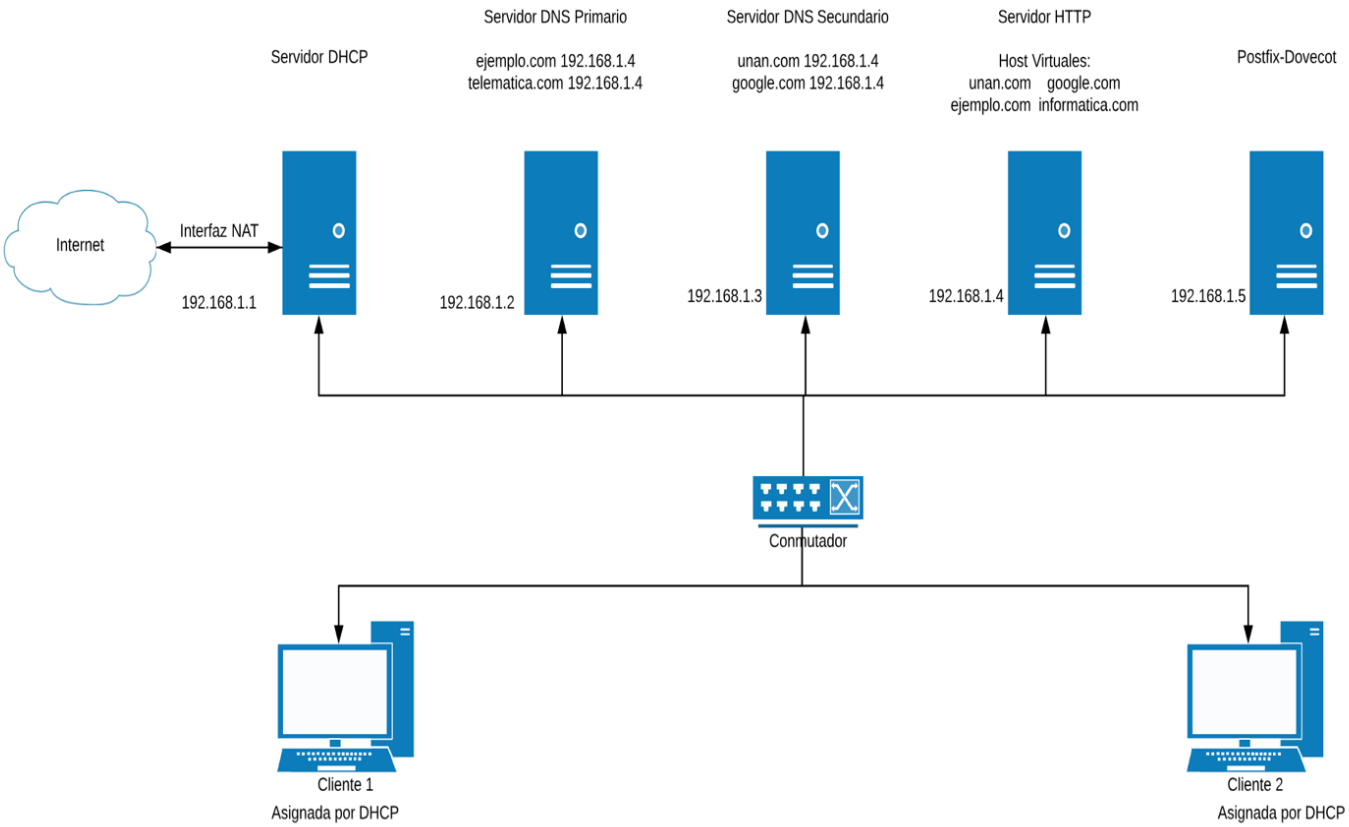

Ilustración 16. Topología de la práctica: Captura de datos HTTP con Wireshark

COMANDOS DE AYUDA

Tabla 16. Comandos de ayuda, práctica: Captura de datos HTTP con Wireshark

Comando	Descripción
`sudo wireshark`	Ejecuta wireshark con permisos de administrador.
`a2enmod ssl`	Habilita modulo SSL(HTTPS)
`a2ensite default-ssl.conf`	Habilita la configuración SSL por defecto (HTTPS)

En la configuración de los hosts virtuales se necesita agregar las siguientes directivas:

Tabla 17. Directivas para el host virtual, práctica número 5.

Directiva	Descripción
`SSLEngine on`	Activa el motor de servidor web seguro
`SSLCertificateFile /etc/apache2/ssl/apache.crt`	Localiza el certificado SSL
`SSLCertificateKeyFile/etc/apache2 /ssl/apache.key`	Localiza la clave para el cetificado SSL

Código de los archivos de la página web del host virtual de apache2 (www.ejemplo.com)

Archivo index.html

```
<!DOCTYPE html><html><head>
<meta charset="utf-8">
<title>Ejemplo</title></head><body>
    <form action="accion.php" method="post">
      <input type="text" name="usuario" value="" placeholder="usuario">
      <input type="password" name="password" value="" placeholder="password">
      <input type="submit" name="" value="enviar">
    </form>
  </body>
</html>
```

Archivo accion.php

```
<?php
  if($_POST['usuario'] =="ejemplo" && $_POST['password'] =="123" )
    echo "credenciales correctas...";
  else
    echo "credenciales incorrectas...";
?>
```

ENUNCIADO

En esta práctica se pretende, instalar Wireshark en la máquina Cliente1. Se modificará la página web (index.html) del dominio www.ejemplo.com se construirá un form que reciba un usuario y una contraseña que se enviaran a un archivo llamado accion.php, que validará si las credenciales (usuario= ejemplo, password= 123) son correctas. Los códigos de los archivos del host virtual (www.ejemplo.com) se encuentran definidos en el recuadro anterior.

Se escuchará en todas las interfaces de la máquina virtual Cliente1 con Wireshark y se filtrarán los paquetes, con el siguiente filtro “http.request.method == POST”, desde el navegador entrar a la página web www.ejemplo.com. y enviar las credenciales, que deseen.

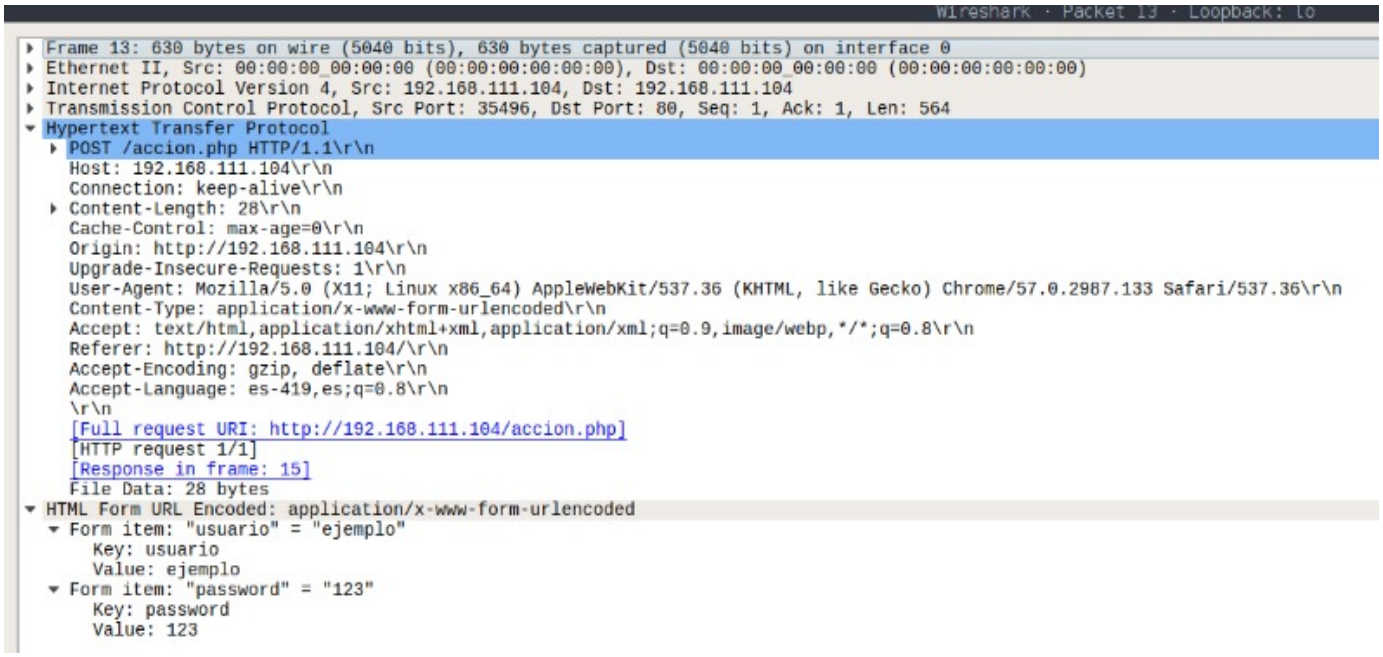

Ilustración 17. Envió de credenciales por HTTP, práctica 5.

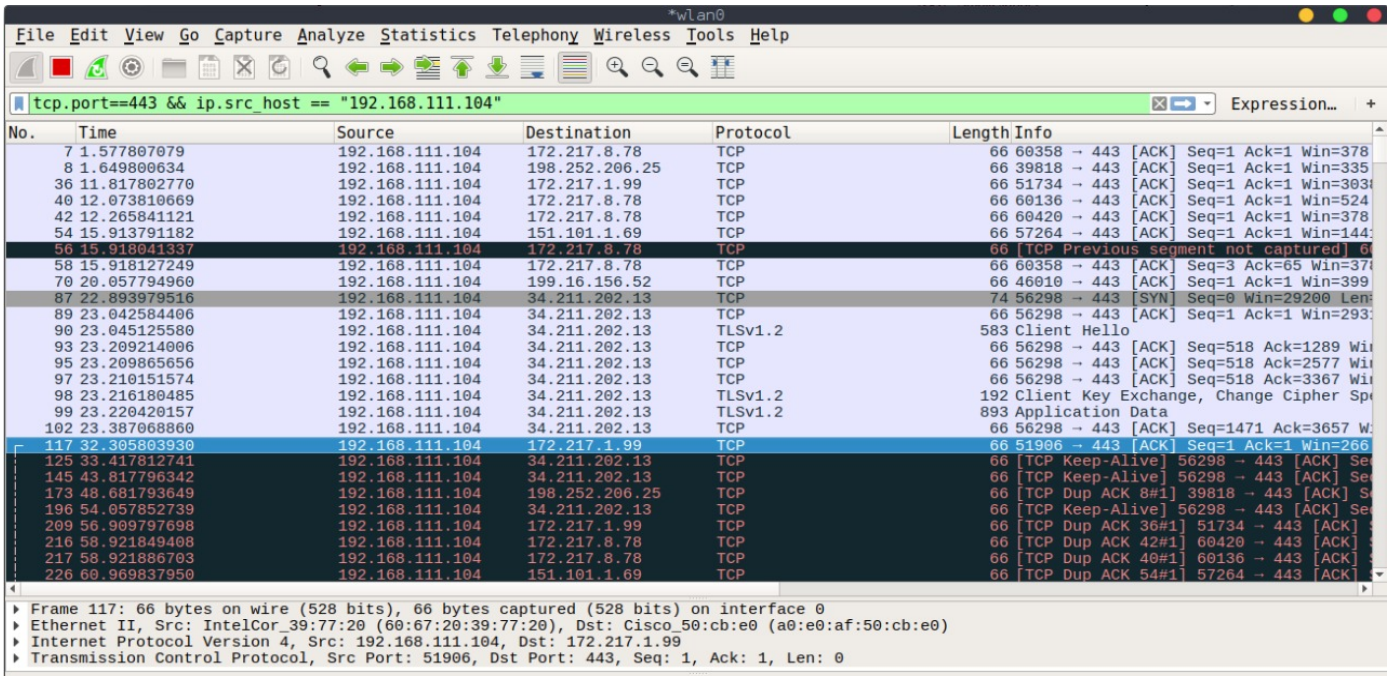

Ilustración 18. Envió de credenciales por HTTPS, práctica 5.

TIEMPO ESTIMADO DE SOLUCIÓN

- 4 horas.

MEJORAS A IMPLEMENTAR

- Configurar el host virtual con HTTPS (URL)
- Repetir el envío de las credenciales, aplicar el filtro tcp.port == 443 en wireshark.

PREGUNTAS DE ANÁLISIS

- ¿Cuál es la función de la cabecera Host en el mensaje de solicitud?
- Conociendo el esquema de una solicitud HTTP, ¿Qué pasaría si un cliente intencionalmente no envía el carácter LRLF que indica el fin de la cabecera?
- ¿Por qué es importante aplicar HTTPS a las páginas web? Justifique su respuesta

REFERENCIAS BIBLIOGRÁFICAS

INCIBE, Análisis de Trafico con Wireshark [online]. Disponible en: https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf

James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.a. Ed. Pearson, 2017, pp. 523-528.

Hostdime, Como Crear un Certificado SSL [online]. Disponible en: http://blog.hostdime.com.co/como-crear-un-certificado-ssl-en-apache-para-ubuntu-14-04/

A. Freier, The Secure Sockets Layer, RFC 6101 [online]. Disponible en: https://tools.ietf.org/html/rfc6101

ITT, Protocolos de Seguridad en la Capa de Transporte [online]. Disponible en: https://www.iit.comillas.edu/palacios/seguridad_dr/tema4_ssl.pdf

PRÁCTICA 6: Observación de tramas HTTP persistentes y no persistentes

OBJETIVO GENERAL

- Observar tramas HTTP.

OBJETIVOS ESPECÍFICOS

- Definir las diferencias entre una conexión HTTP tradicional y HTTP persistente.
- Ventajas y desventajas de conexiones HTTP persistentes.

INTRODUCCIÓN

El protocolo HTTP puede utilizar conexiones persistentes y no persistentes, por ejemplo, sí pedimos una página web a un servidor y la página consta de un HTML y varios objetos, en una conexión persistente solo se hará una conexión TCP, mientras que en una conexión no persistente se utilizarán múltiples conexiones TCP, una por cada objeto solicitado.

Estas conexiones pueden ser paralelas para mejorar el rendimiento, por lo que un navegador puede realizar por peticiones al mismo tiempo en vez de ir realizando una conexión tras otra (en serie), que habitualmente alargaría el tiempo de conexión. Utilizando conexiones persistentes el servidor mantiene abierta una conexión TCP para que las siguientes peticiones y respuestas se transmitan por esa conexión.

Apache2 es uno de los servidores web más utilizados, el cual implementa las conexiones persistentes por medio de la directiva KeepAlive, por defecto estas vienen en estado On, con esta directiva en modo On y las directivas KeepAliveTimeOut (cantidad de tiempo en segundos que el servidor espera la petición subsiguiente), MaxKeepAliveRequest (número de peticiones permitida en una conexión persistente). Estas directivas pueden ser encontradas en el archivo /etc/apache2/apache.conf.

REQUERIMIENTOS

Para la práctica: observación de tramas HTTP persistentes, el ordenador que se destinará para realizar esta práctica debe contar con los siguientes requisitos:

HARDWARE

- Procesador mínimo de velocidad de 1.5GHz
- Memoria RAM de 4 GB o mas

SOFTWARE

- Apache2 version 2.4.25-3
- Navegador Web Chrome versión 70.0.3538.110

Para desarrollar la práctica sin ningún problema, se recomienda iniciar las máquinas virtuales en modo texto, ya que no consumirán tantos recursos, a como lo harían en modo gráfico. Si se desea hacer peticiones al servidor web desde un navegador, puede dejar las máquinas virtuales de los clientes en modo gráfico y detener los servicios de Postfix y Dovecot, ya que no se utilizarán. Instalar el nuevo servidor apache en el mismo servidor donde está instalado el bind9 para que la práctica se torne más ligera, para los computadores.

TOPOLOGÍA

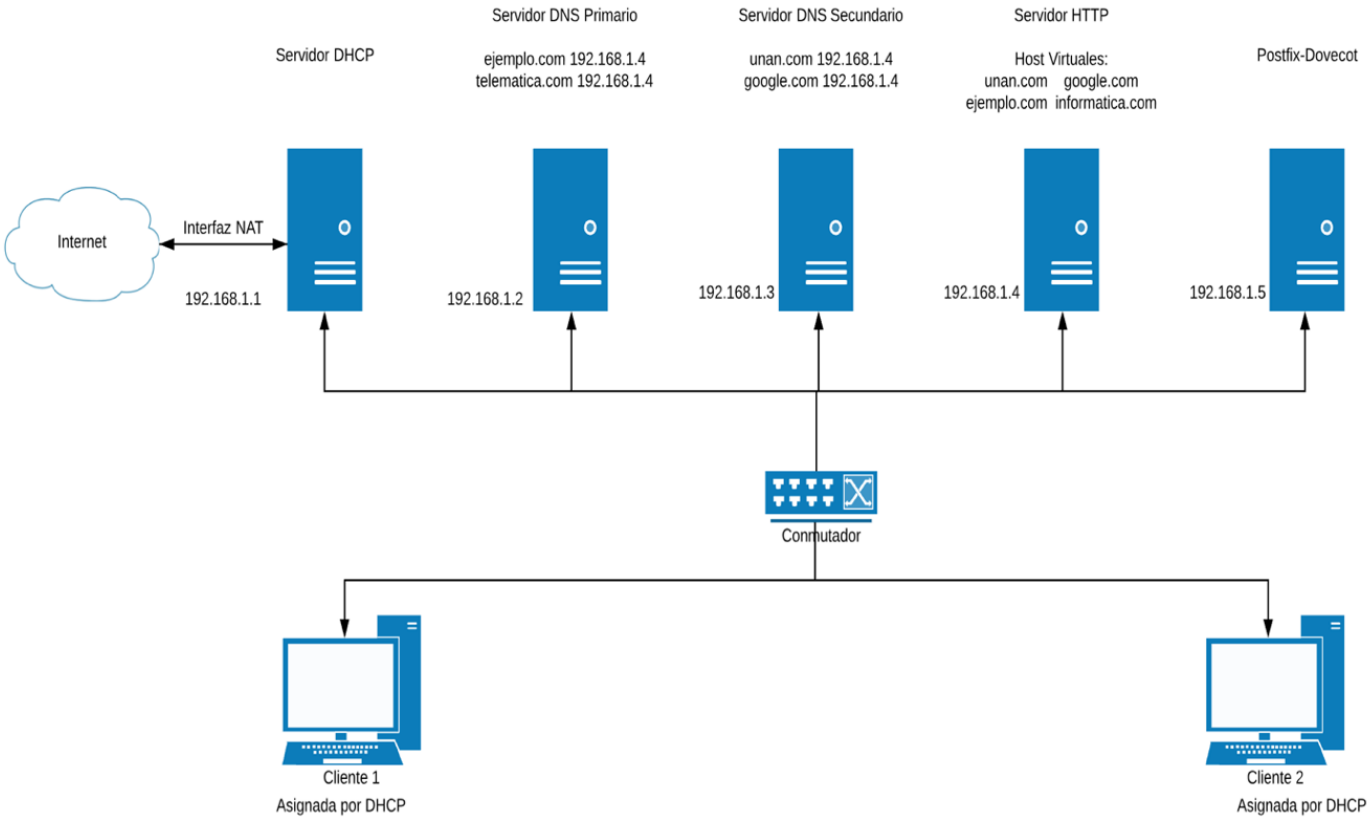

Ilustración 19. Topología de la práctica 6.

COMANDOS DE AYUDA

Tabla 18. Comandos de ayuda, práctica 6

Comando	Descripción
`sudo apt-get install apache2`	Instala apache2 en nuestro ordenador.
`http.request.method == "GET" || http.response.code == 200`	Filtro de Wireshark para ver las tramas HTTP con el método GET o las tramas HTTP que tiene una respuesta 200(OK)
`sudo wireshark`	Ejecuta wireshark con nivel privilegiado.

ENUNCIADO

Para esta práctica se adjuntará una página web básica que contenga 10 imágenes. Esta se configurará en vez de la página que trae apache2 por defecto en el archivo de configuración /va/www/html/

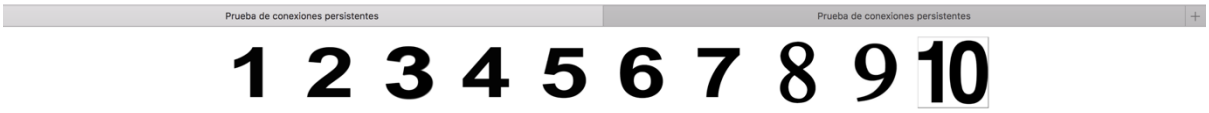

Ilustración 20. Página de prueba, práctica 6.

Para cada servidor web, crear un host virtual para la página que se les compartirá, y crear una zona en el servicio de DNS. Se usará dos servidores HTTP (apache2), los cuales se configurarán, uno con la directiva KeepAlive con estado On y el otro con la directiva KeepAlive con estado Off. Abrir el Wireshark en el cliente, escuchar en todas las interfaces, aplicar el filtro descrito en la tabla 18. Hacer la petición desde un navegador web (Chrome o Firefox) al servidor web que tiene configurada la directiva KeepAlive con el estado On, observar la estructura de las cabeceras HTTP. Repita el mismo procedimiento, pero ahora haciendo la petición web al servidor que tiene configurada la directiva KeepAlive con estado Off. Observe la estructura de las cabeceras HTTP.

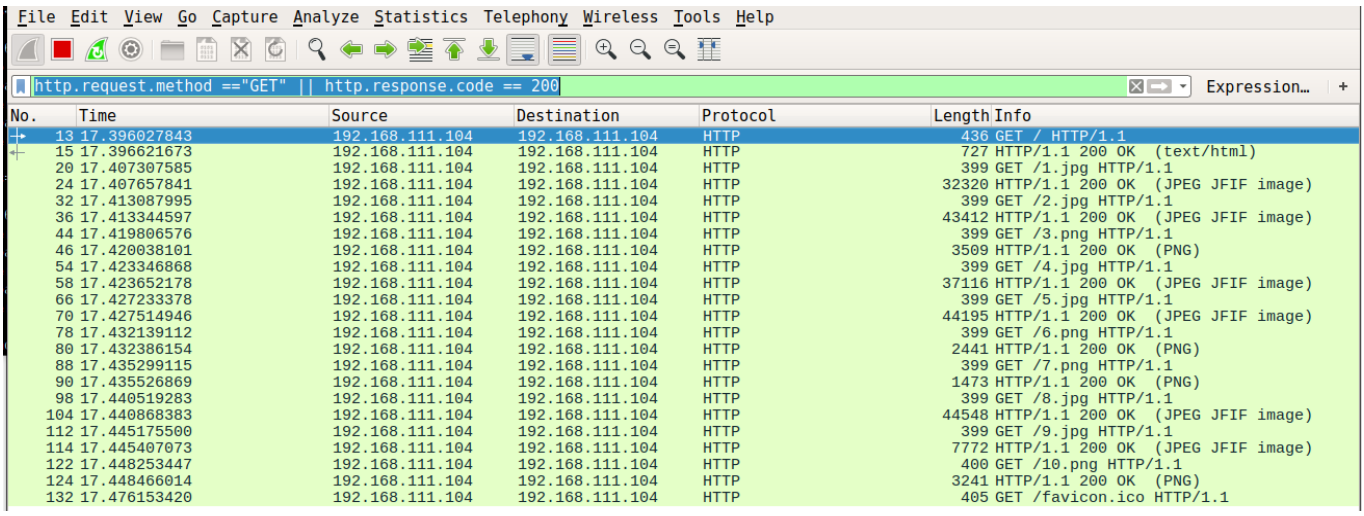

Ilustración 21. Conexiones no persistentes, práctica 6

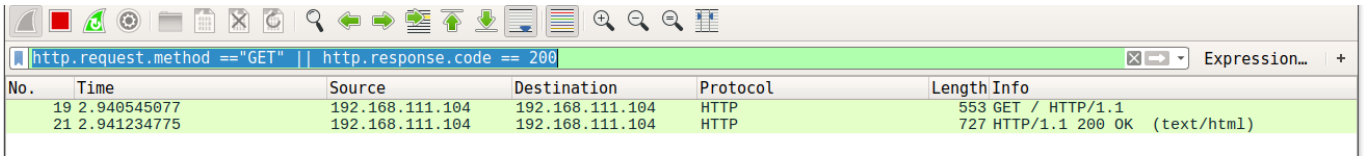

Ilustración 22. Conexiones persistentes, práctica 6.

TIEMPO ESTIMADO DE SOLUCIÓN

- 4 horas.

MEJORAS A IMPLEMENTAR

- Utilizar la version 2.0 de HTTP para que utilize conexiones persistentes con procesamiento en cadena en paralelo en lugar de en serie.

PREGUNTAS DE ANÁLISIS

- ¿Cómo identificar una trama HTTP persistente con respecto a una no persistente?
- ¿Qué indica la cabecera Connection: keep-alive?
- ¿Cuáles son las ventajas de usar HTTP persistente?

REFERENCIAS BIBLIOGRÁFICAS

James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.a. Ed. Pearson, 2017, pp. 83-85.

UTFSM, HTTP/2.0 [online]. Disponible en: http://profesores.elo.utfsm.cl/~agv/elo322/1s16/projects/reports/HTTP-2.0.pdf

Vozidea, KeepAlive en servidores Apache [online]. Disponible en: https://www.vozidea.com/optimizando-keepalive-en-servidores-apache

5 Conclusiones

Con la culminación de este trabajo de tesis, consideramos que hemos logrado llevar a cabo los objetivos propuestos, llegando a las siguientes conclusiones:

- La definición de un apropiado formato de práctica proporcionará a los estudiantes una adecuada compresión los conocimientos y un abordaje apropiado de los mismos.
- La integración entre las prácticas facilitará a los estudiantes la correcta solución de estas.
- La integración de temas teóricos y ejercicios prácticos le proporciona a los docentes y estudiantes una base consistente para el desarrollo de las prácticas.

Con los elementos antes mencionados se puede afirmar que logramos desarrollar un documento sencillo y objetivo que resuelve las necesidades planteadas en la definición del problema del presente trabajo.

6 Recomendaciones

Las recomendaciones descritas a continuación se fundamentan en una futura actualización del documento.

- Los temas que presenta este documento son de gran trascendencia para la evolución del aprendizaje de los alumnos de la carrera de Ingeniería en Telemática en lo que corresponde al ámbito de servicios de red, no obstante, deben renovarse continuamente con temas de relevancia, como los mencionados a continuación :
    - Configuración firewalls usando iptables.
    - Implementación de servidores SSH para conexiones seguras.
    - Implementar calidad de servicios.
    - Configurar servicios de logs.

- Elaborar las prácticas en el sistema operativo de Microsoft Windows 10 para extender el uso y aplicación en otras plataformas.
- Elaborar guia práctica para docentes y estudiantes de forma independiente.
- Se recomienda mantener la relación de las prácticas dado que están distribuidas a fin de garantizar integración entre ellas.

7 Bibliografía

[1]. Unece, Servicios de red [online]. 2012 Disponible en: http://tfig.unece.org/SP/contents/web-services.html

[2]. Uma, Servicios de red [online]. Disponible en: http://vgg.uma.es/redes/servicio.html

[3]. Andrew S. Tanenbaum. Redes de computadoras, 5ta. Ed. Pearson, 2012, pp. 657-748.

[4]. William Stallings, Comunicaciones y Redes de computadores, 7ta. Ed. Pearson, 2008.

[5]. Carlos Valdivia Miranda. Sistemas informáticos y redes locales, 1ra. Ed. Paraninfo, 2014, pp. 120-125.

[6]. Unavarra, Aplicaciones [online]. Disponible en : https://www.tlm.unavarra.es/pluginfile.php/25632/mod_resource/content/1/cap6-1-Aplicaciones_HTTP.pdf

[7]. R. Droms,. Dynamic Host Configuration Protocol, Bucknell University, RFC 2131, Marzo de 1997.

[8]. T.Berners-Lee, R. Fielding. Hypertext Transfer Protocol, RFC 2616, Junio de 1999.

[9]. P. Mockapetris,. Domain Names - Implementation and Specification, RFC 1035, Noviembre de 1987.

[10]. James F. Kurose, Keith W. Ross,. Redes de computadoras: Un enfoque descendente, 7.ª. Ed. Pearson, 2017.

[11]. DRAFT STANDARD,. Simple Mail Transfer Protocol, RFC 5321, Octubre de 2008.

[12]. A. Gulbrandsen,. Internet Message Access Protocol, RFC 6851, Enero de 2013.

[13]. Raúl Siles Peláez. Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados, 1ra. 2012.

[14]. Ccm, Introducción a la seguridad informática [online]. Disponible en: https://es.ccm.net/contents/622-introduccion-a-la-seguridad-informatica

[15]. Certsuperior, Seguridad en Redes [online]. Disponible en: https://www.certsuperior.com/SeguridadenRedes.aspx

[16]. Redalia, Protocolo SSL [online]. Disponible en: https://www.redalia.es/ssl/protocolo-ssl/

[17]. Certsuperior, SSL como funciona [online]. Disponible en: https://www.certsuperior.com/SSLComoFunciona.aspx

[18]. Alvaro Gómez Vieites. Tipos de ataques e intrusos en las redes informáticas [online]. Disponible en: https://repository.javeriana.edu.co/bitstream/handle/10554/12715/ZaccaroValverdeJorgeAndres2013.pdf;sequence=1

Anexos

Anexo 1: Instalación y configuración de un certificado SSL

Para la creación de los certificados en la práctica número 5 del presente documento, se brindará un enlace externo que muestra la forma de instalar y configurar certificados SSL, que se utilizan en la práctica 5 para recalcar la importancia de usar conexiones cifradas que garantizan la integridad de la información; tal manual se encuentra disponible en: http://blog.hostdime.com.co/como-crear-un-certificado-ssl-en-apache-para-ubuntu-14-04/

Anexo 2: Automatización de prácticas mediante scripts

Se han realizado 4 shell scripts que se suministrarán a los docentes que lo requieran para configurar automáticamente los servicios de la práctica 1 hasta la práctica 4, con el objetivo que, si algún estudiante presenta un determinado caso y no ha logrado configurar una práctica, el script automatizará el proceso de configuración y le permitirá abordar la siguiente.

El paquete que se suministrará al docente lo encontrará disponible en el siguiente enlace, https://mega.nz/#!DKwDzAKD!vm8gYTrnaQkYNSkI12wIgetLy6okv9LO_s250mDFbIY ,una vez descargado utilizará la contraseña ScriptsServiciosDocentes2018 para acceder al paquete docente, en el cual encontrará lo siguiente:

- Manual de uso docente, contiene informacion sobre requerimientos y ejecución de los scripts.
- Script DHCP, automatizará el proceso de configuración de la práctica 1.
- Script DNS, automatizará el proceso de configuración de la práctica 2.
- Script HTTP, automatizará el proceso de configuración de la práctica 3.
- Script EMAIL, automatizará el proceso de configuración de la práctica 4.