# UNIVERSIDAD NACIONAL
# “HERMILIO VALDIZÁN”

**FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS**

**ESCUELA ACADEMICA PROFESIONAL INGENIERIA DE SISTEMAS**

---

**Proyecto de tesis:**

# “APLICACIÓN DE UN FIREWALL CON IPTABLES EN LA EMPRESA CONEXIÓN LINUX SAC.”

---

**TESIS PARA OPTAR EL TITULO DE: Ingeniero de Sistemas**

**TESISTA: Joseph Frank Veliz Castañeda**

**ASESOR: Ing. Luis Meza Ordoñez**

**Huánuco – Perú**

**2016**

**DEDICATORIA**

A Dios por guiarme en mis pasos, a mi madre que es mi alma gemela, a mi padre, mis hermanas por su apoyo incondicional y a Patricia Luz Veliz Castañeda que desde el cielo me cuida, a ti hermana con mucho cariño.

## AGRADECIMIENTO

A Dios por todo lo que me brinda, a mi madre, a mi padre, mis hermanas, mis docentes de la EAP Ing. de Sistemas, mi asesor, y a Patricia Luz Veliz Castañeda, que desde el cielo me ayuda.

## RESUMEN

En la empresa Conexión Linux SAC se ha detectado eminentes riesgos como los ataques de agentes externos a la configuración de red, esto sucedió en dos oportunidades observado por el gerente de dicha empresa; como también se observó lentitud cada cierto tiempo de la red LAN. Por lo cual se debería minimizar la vulnerabilidad y la lentitud, usando como herramienta “iptables” en la construcción de un firewall perimetral para el servidor master.

Para ello se desea implementar un firewall con iptables, creando un script en el sistema operativo, que ayudará a controlar los ingresos y salidas de paquetes a la red LAN como también a minimizar la lentitud.

El script se dividirá en dos partes, un script donde irán las variables, y el segundo script donde ira todo el código completo del firewall, en este último script el firewall tendrá tres partes, la primera parte contendrá todo el código necesario para un firewall, la segunda parte serán las reglas duras, y por ultimo las reglas blandas. En las reglas duras estará todos los filtros de acuerdo a las políticas de seguridad planteada por la empresa, mientras que en las reglas blandas se encontrarán todos los filtros adicionales que la empresa lo requiera algún momento determinado.

Una vez realizado la construcción del script se procederá a realizar las pruebas de software, midiendo las variables de vulnerabilidad y lentitud, usando las siguientes herramientas: wireshark, iperf, nmap, bmw-ng y arp –a.

Concluyendo, se verifico que existía vulnerabilidad en algunos puertos y que de acuerdo a las políticas de seguridad solo tendrán acceso dos usuarios, usando como medio de accesibilidad el protocolo SSH, y en cuanto a la lentitud bloquear el broadcast, el IPV6 y las reglas predeterminadas del firewall que hacen que la red LAN colapse cada cierto tiempo.

## SUMMARY

In the company (Connection Linux SAC) has been detected eminent risks and attacks outside the network configuration agents, this happened on two occasions observed by the manager of the company. Therefore it should minimize vulnerability and slow, using as "iptables" tool in building a perimeter firewall for the master server.

To do this you want to deploy a firewall with iptables, creating a script in the operating system, which will help control inputs and outputs packet to the LAN as well as minimize slowly.

The script is divided into two parts, a script where will the variables, and the second script where anger all the full code of the firewall, in the latter script the firewall will have three parts: the first part will contain all the necessary code for a firewall, the second part will be tough rules, and finally the soft rules. In the harsh rules will be all filters according to security policies raised by the company, while the soft rules every additional filters will be found that the company requires any given time.

Once completed the construction of the script will proceed to perform software testing, measuring variables vulnerability and slowly, using the following tools: wireshark, iperf, nmap, bmw-ng and arp -a.

In conclusion, it was found that there was vulnerability in some ports and according to security policies only have access two users using as a means of accessibility the SSH protocol, and as for the slowness block broadcast, the IPV6 and rules default firewall that make the LAN collapse from time to time.

# INTRODUCCIÓN

Las vulnerabilidades hacen referencia que existe debilidad en un sistema, permitiendo a un atacante violar la confidencialidad, integridad y disponibilidad. Entre las diferentes maneras que un servidor quede vulnerado, el más común son los puertos abiertos, mientras más puertos se encuentren abiertos, más formas hay para que alguien se conecte.

En la empresa Conexión Linux SAC el servidor master fue vulnerado en dos oportunidades con ataques de agentes externos a la configuración de red, y también se observó lentitud cada cierto tiempo de la red LAN. Por lo que se debería minimizar la vulnerabilidad y la lentitud usando como herramienta “iptables” en la construcción del firewall. Para ello se realizó los siguientes capítulos en el desarrollo de la tesis:

En el capítulo uno, se formuló el problema, se platearon los objetivos, se definió las variables, se justificó el uso de iptables en el firewall, se asignó la viabilidad de la tesis, y por último se definieron las limitaciones del firewall tanto a nivel de software como a nivel de hardware.

En el capítulo dos, se citó seis antecedentes relacionados al tema del firewall y de seguridad a la red, seguidamente se planteó las bases teóricas relacionadas al firewall como son: firewall, redes de datos, modelo OSI, modelo TCP/IP, transmisión de datos, iptables, seguridad; no obstante se definió los términos empleado en el marco teórico.

En el capítulo tres, se definió el nivel y tipo de investigación que tendrá la tesis, como también la población y muestra de la investigación.

En el capítulo cuatro, se describió a la empresa Conexión Linux SAC, como misión, visión, contacto, equipo de trabajo y laboratorios; como también se hizo un análisis situacional de la red.

En el capítulo cinco, se diseñó las políticas y reglas de seguridad de la empresa, lo requerimientos tanto de la solución como del script; y sus particularidades, diseño y construcción del script.

En el capítulo seis, se presentó la aplicación con sus dos archivos script del firewall para luego pasar a las pruebas pertinentes, donde se abarco en dos partes, uno relacionado a la vulnerabilidad y lentitud y el segundo para medir las variables definidas en el capítulo 1. Finalmente se dio a conocer las tres versiones del AllInOne, quedando la versión 3 como la versión final.

Con el desarrollo de la presente investigación se concluyó que había puertos que eran vulnerables al servidor master, lo cual se desarrolló un diseño de acuerdo a las políticas de seguridad de la empresa, se determinó usar medios de accesibilidad seguros, y por último se logró bloquear algunas reglas como broadcast, etc. Con la finalidad de tener un tráfico limpio y garantizado al funcionamiento de la red.

# Índice

# CAPÍTULO 1

# 1

# EL PROBLEMA DE LA INVESTIGACIÓN

## 1.1. Descripción del problema

La empresa Conexión Linux SAC cuenta con una estructura de red interna, donde el servidor master fue vulnerado en dos oportunidades, dejando sin servicio de internet en la red LAN y los DMZ totalmente corrompidos. La primera vez que fue vulnerado detuvieron el squid y manipularon la configuración, por otra parte saturaron los DMZ bombardeando con un sinfín de peticiones desde el Servidor Master. En la segunda oportunidad eliminaron la configuración de los DNS y cambiaron el puerto de enlace de los DMZ, quedando totalmente aislados.

Pero el problema no solo era la vulnerabilidad a tratar, si no la lentitud en la red LAN y los DMZ. Los trabajadores se quejaban que cada cierto tiempo, había una lentitud en responder las peticiones solicitadas, ya sea a los DMZ o a la INTERNET, lo que causaba molestias por parte de los usuarios.

Se entendía entonces que el problema era seguridad informática en redes, que la seguridad empleada y la velocidad de transmisión de datos en el servidor master debían mejorar, y se tendría que administrar en forma segura desde la WAN y LAN.

Si bien es cierto que el Servidor Master cuenta con un puente de conexión, este no está estructurado ni desarrollado, en otras palabras no tiene un script.sh donde se pueda administrar y asegurar las peticiones tanto de la WAN como de la LAN (firewall perimetral), lo único que cuenta es con los puertos abiertos a los DMZ.

## 1.2. Formulación del problema

En la estructura de la red LAN de la empresa Conexión Linux SAC se ha detectado eminentes riesgos como los ataques de agentes externos a la configuración de red; por lo que se debería minimizar la vulnerabilidad y la lentitud, usando como herramienta “iptables” en la construcción de un firewall perimetral para el servidor master.

### 1.2.1. Formulación del problema general

¿Cómo minimizar la vulnerabilidad y lentitud de la red LAN, utilizando un firewall con iptables en la empresa Conexión Linux SAC?

### 1.2.2. Formulación del problema especifico

1. ¿Qué puertos y qué ip son vulnerables a los DMZ y al servidor master?

2. ¿Cuál sería la política de seguridad en el nivel de usuario y medios de acceso al servidor master?

3. ¿De qué manera se debe optimizar el tráfico de red y asegurar su transmisión de datos?

## 1.3. Objetivos

### 1.3.1. Objetivo general

Implementar un firewall con iptables para reducir la vulnerabilidad y lentitud de la red LAN en la empresa Conexión Linux SAC.

### 1.3.2. Objetivos específicos

1. Realizar un testeo para verificar los puertos y los ip vulnerables en la red LAN.

2. Identificar e implementar políticas de seguridad para los niveles de usuario que existen, así como también los medios de accesibilidad desde la LAN y WAN.

3. Implementar un algoritmo que optimice la velocidad de transmisión y la integridad de los paquetes de datos.

## 1.4. Variables

### 1.4.1. Variable independiente

Firewall con iptables.

#### A. Dimensiones/Indicadores

1. Enrutador con Capacidades de Filtrado.
2. Servidor de Control a Nivel Circuito.

### 1.4.2. Variable dependiente

Vulnerabilidad y lentitud.

#### A. Dimensiones/Indicadores

1. Escaneo.
2. Obtención de Acceso.
3. Manteniendo el Acceso.
4. Encubrimiento de rastros.

1. Capacidades de transporte de la red LAN.
2. Ancho de banda del ISP.
3. Velocidad y capacidad del servidor de red (capacidad del servidor).
4. Demanda de usuarios en un momento dado.
5. Cantidad de tráfico que compita en la red (en todos los niveles de la red).
6. Capacidades de la computadora del usuario final.

## 1.5. Justificación e importancia

Existen muchas formas de implementar un firewall desde los más caros, como security cisco, endian, antivirus karspersky, entre otros, hasta los que son libres, como iptables, zentyal, ipchain, etc.

Sin embargo al tratarse de una PYME, no cuenta con presupuesto para filtros de costos altos, optando por implementar software libre, y que mejor que GNU/Linux Centos 7.0, para implementar si de seguridad se trata.

Si hablamos de software libre, es hablar de libertad de implementar código a medida, esto no tiene que envidiar nada a los de licencia, es casi lo mismo o aún mejor, ya que se trata de un software personalizado y hecho a medida, todo dependerá del buen uso del programador y del administrador de red.

Iptables es una herramienta de GNU/Linux, no es la única, existen muchos, pero a diferencia de los demás iptables es más seguro y robusto, tiene muchas opciones para trabajar, y te permite programar en caliente sin necesidad de detener el servicio.

Tal vez es una de las ventajas de GNU/Linux Centos 7.0, ya que existe distribuciones especiales para redes, como Red Hat (Si se está dispuesto a pagar por el soporte), Centos (Que es libre pero va a cuenta propia enfrentar desafíos del sistema, cuya responsabilidad es netamente del administrador de red), entre otras muchas distribuciones que se deja a criterio y a gusto del programador.

La importancia radica, que si no tenemos un firewall administrable en una red empresarial, no podremos tener el control de lo que sale e ingresa en nuestra red, por lo tanto es importante la implementación de un firewall.

## 1.6. Viabilidad

### 1.6.1. Espacio / Instalaciones

Todo el proyecto se realizó en los laboratorios de Conexión Linux S.A.C., con domicilio en la calle Emilio Atheus 256, San Miguel – Lima, 2° piso. Supervisado cada avance por el Mg. Alex Segura.

### 1.6.2. Instrumentos de medición.

Los instrumentos de medición son:

a) Wireshark.
b) Bwm-ng.
c) Iperf.
d) Arp –a.
e) Nmap.

### 1.6.3. Equipos, materiales e instrumentos

#### a) Equipos y materiales

##### 1. Servidor

- ✓ Procesador Core i7 4770k Lga1150.
- ✓ 16 Gb de memoria RAM DDR3 (Kingston).
- ✓ 240 Gb de disco duro solido (Kingston).
- ✓ Placa madre Asus Z97-A.
- ✓ 2 tarjetas de red DLink (1Gb / seg. Dual port).
- ✓ Lectora de DVD Samsung.

**2. Administrador**

- ✓ Laptop Lenovo Z470.
- ✓ Procesador Core i5.
- ✓ 16 Gb de memoria RAM (Kingston).
- ✓ 240 Gb de disco duro solido (Kingston).

**3. Sistema Operativo**

- ✓ Centos 7.0.

**b) Instrumentos**

- ✓ Wireshark.
- ✓ Winscp.
- ✓ Putty.
- ✓ SSH.
- ✓ Nmap.
- ✓ Tail –f /var/log/messege.
- ✓ Tail –f /var/log/squid.
- ✓ Bwm-ng.
- ✓ Arp –a.
- ✓ Iperf.
- ✓ Packet tracer.

### 1.6.4. Cooperación de otras personas

La cooperación del Mg. Alex Segura, en cuanto a pruebas y errores, las revisiones se realizaban cada procedimiento terminado.

### 1.6.5. Recursos financieros

#### a) Servidor

- ✓ Placa madre Asus Z97-A: 690 Soles.
- ✓ 16 Gb de memoria RAM DDR3: 414 Soles.
- ✓ Procesador Core i7 4770k Lga1150: 1190 Soles.
- ✓ 240 Gb de disco duro solido (Kingston): 420 Soles
- ✓ 2 tarjetas de red DLink (1Gb / seg. Dual port): 420 Soles.
- ✓ Lectora de DVD Samsung: 100 Soles.

#### b) Administrador

- ✓ Laptop Lenovo Z470: 1800 Soles.

#### c) Costos

Financiado por Conexión Linux SAC.

## 1.7. Limitaciones

El firewall construido con iptables tiene las siguientes limitaciones:

### 1.7.1. Nivel software:

- ✓ El firewall no contiene entorno gráfico para el usuario administrador.
- ✓ El firewall no se usa como un Servidor de Control de Aplicaciones (Proxy).
- ✓ El firewall no puede protegerse contra los ataques de la “Ingeniería Social”
- ✓ El firewall no puede protegerse contra los ataques posibles a la red interna por virus informáticos a través de archivos y software.
- ✓ El firewall no protege de los fallos de seguridad, de los servicios y protocolos de los cuales se permita el tráfico de red.

### 1.7.2. Nivel Hardware:

- ✓ Slot PCI, no cuenta con slot suficientes para emergencia, tiene 3 slot disponibles.
- ✓ Limitaciones del funcionamiento, al no tratarse de una infraestructura de un servidor, se sobrecalienta y tiene que ser reiniciado cada cierto tiempo (se sugiere cada semana).
- ✓ Mantenimiento de limpieza, al no ser un hardware de un servidor, se tiene que limpiar constantemente los cooler de la PC, para evitar sobrecalentamientos.

# CAPÍTULO 2

# 2

# MARCO TEÓRICO

## 2.1. Antecedentes

**2.1.1. Tesis:** "DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA DE SERVICIOS DE RED Y RESGUARDO DE SERVIDORES LINUX A TRAVÉS DE OPEN SOURCE EN LA EMPRESA PROTECO COASIN S.A.", presentado por Luis Guillermo León Bustamante, el 2012, que sirvió para optar el título de Ingeniero de Sistemas en Telecomunicaciones en la Universidad Internacional Sek (Quito-Ecuador), refiere: "La solución a los problemas detectados en la infraestructura de la red actual y que mediante la implementación de servicios basados en Linux han mejorado el rendimiento y la administración."[1]

**2.1.2. Tesis: "**EVALUACIÓN DE LAS VULNERABILIDADES QUE PRESENTAN LOS FIREWALLS EN LA EMPRESA DATASOLUTION S.A.", presentado por Erika Lidia Montaño Rivas, el 2011, que sirvió para optar el título de Ingeniero en Sistemas Computacionales en la Universidad de Guayaquil (Guayaquil-Ecuador), refiere: "Al estudio de las vulnerabilidades de los firewalls en la empresa Data Solution orientándose a los firewall Linux y como software libre ayudan a la protección de las redes de las empresas." [2]

---

[1]http://repositorio.uisek.edu.ec/jspui/bitstream/123456789/534/1/TESIS%20FINAL%20LUIS%20GUILLERMO%20LE%C3%93N%20BUSTAMANTE.pdf [Citado el: 02 de diciembre del 2015, 9:08 am.]

[2]http://repositorio.ug.edu.ec/bitstream/redug/6743/1/Tesis%20Completa%20-342-2011.pdf [Citado el: 02 de diciembre del 2015, 10:14 am.]

**2.1.3. Tesis:** “VIRTUALIZACIÓN DE UNA RED LAN CON SERVIDORES DE CÓDIGO ABIERTO PARA EVALUAR LOS NIVELES DE SEGURIDAD”, presentado por Cesar Libardo Rosado Muñoz, el 2014, que sirvió para optar el grado de Magíster en Telecomunicaciones en la Universidad Católica de Santiago de Guayaquil (Guayaquil-Ecuador), refiere: “A una virtualización de una red conteniendo una Zona Desmilitarizada (DMZ), donde se ha implementado un Cortafuego mediante IPtables al cual se le aplican cadenas para traslado de direcciones y filtrado de paquetes a conveniencia según un esquema determinado.” [3]

**2.1.4. Tesis:** “SEGURIDAD EN REDES DE DATOS”, presentado por Luis Alberto Orellana Benavides y Rafael Cristóbal Hernández Vásquez, el 2003, que sirvió para optar el grado de Ingeniero en Electrónica en la Universidad Don Bosco en El Salvador, refiere: “Al crecimiento de redes y la cantidad de información disponible, que lleva a la seguridad de redes de datos mediante diferentes modelos de seguridad.” [4]

---

[3]http://repositorio.ucsg.edu.ec/bitstream/123456789/1697/1/T-UCSG-POS-MTEL-14.pdf [Citado el: 03 de diciembre del 2015, 3:11 pm.]

[4]http://rd.udb.edu.sv:8080/jspui/bitstream/11715/281/1/033380_tesis.pdf [Citado el: 03 de diciembre del 2015, 3:32 pm.]

**2.1.5. Tesis:** “DISEÑO E IMPLEMENTACIÓN DE UNA RED LAN Y WLAN CON SISTEMA DE CONTROL DE ACCESO MEDIANTE SERVIDORES AAA”, presentado por Nuttsy Aurora Lazo García, el 2012, que sirvió para optar el grado de Título en Ingeniería de las Telecomunicaciones en la Universidad Católica del Perú (Lima - Perú), refiere: “Al culminar con la implementación del presente proyecto se pudo concluir que, gracias al servidor RADIUS, un usuario inalámbrico puede autenticarse e ingresar a la red; asimismo, el servidor TACACS+, teniendo como base el nivel de privilegio del usuario, permite a este ingresar o no a los equipos de red para realizar configuraciones en los equipos.”[5]

**2.1.6. Tesis:** “DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR SDN/OPENFLOW PARA UNA RED DE CAMPUS ACADÉMICA”, presentado por Gabriel Josías Cuba Espinoza y Juan Manuel Augusto Becerra Ávila, el 2015, que sirvió para optar el grado de Título en Ingeniero de las Telecomunicaciones en la Universidad Católica del Perú (Lima - Perú), refiere: “El paradigma de redes SDN permite quitar el Plano de Control de los equipos de red y centralizarlo en un elemento llamado Controlador, el cual tiene conocimiento de toda la red, por lo que hace posible un mejor uso de los recursos de esta, haciendo que sea flexible y escalable en el Plano de Datos, y disminuyendo así los gastos de OPEX.”[6]

---

[5]http://tesis.pucp.edu.pe/repositorio/handle/123456789/1445 [Citado el: 08 de marzo del 2016, 4:07 pm.]

[6]http://tesis.pucp.edu.pe/repositorio/handle/123456789/7149 [Citado el: 11 de abril del 2016, 2:38 pm.]

## 2.2. Bases teóricas del firewall

### 2.2.1. Concepto general de firewall

Un firewall es un sistema (o grupo de sistemas) que fortalece a las políticas de seguridad entre una red interna segura y una red no confiable como lo puede ser Internet.

Normalmente, los firewall son vistos como un elemento de protección entre Internet y una red privada. Sin embargo, de forma genérica, un firewall se considera como una forma de dividir a las redes en una o más redes seguras, de aquellas que no lo son.

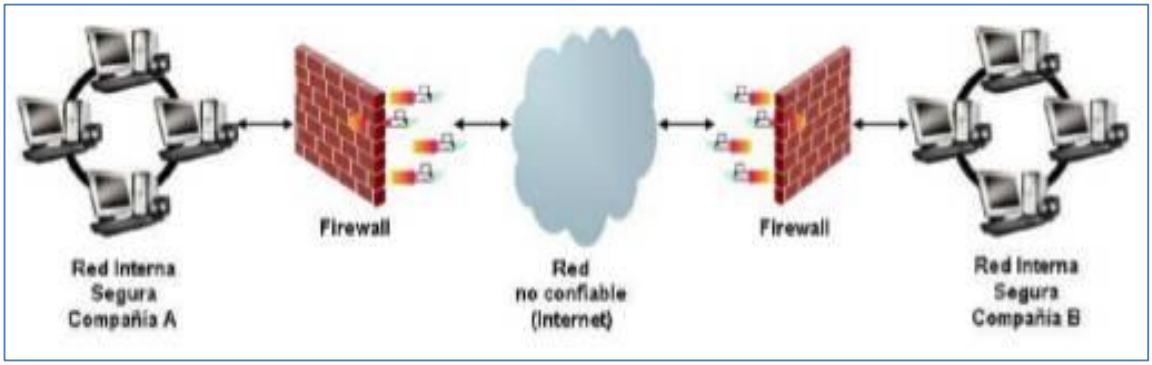



Figura 1.0 División de las redes por uno o varios firewalls

Un firewall puede ser una PC, un enrutador, un servidor o una combinación de estos, simplemente son los dispositivos que determinan cual información o servicios pueden ser accesibles desde el exterior de la red segura.

Generalmente un firewall es instalado en un punto entre la red interna (segura) y la red externa (no confiable), también conocido como "choke point".[7]

---

[7]Concepto General de Firewall. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 04 de enero del 2016, 10:51 am.]

### 2.2.2. Componentes de un sistema firewall

Como ya se mencionó, un firewall puede ser una PC, un enrutador, un servidor o la combinación de los mismos. Por lo que depende del tipo de solución, un firewall puede tener uno o más de los siguientes componentes:

- Enrutador con Capacidades de Filtrado
- Servidor de Control a Nivel Circuito
- Servidor de Control de Aplicaciones (Proxy)

Cada uno de estos componentes tienen diferentes funciones, sin embargo, estos componentes deben ser usados en conjunto para construir una solución efectiva de firewall. [8]

### 2.2.3. Enrutador con capacidades de filtrado

Normalmente el filtrado de paquetes está relacionado con las funciones de enrutamiento donde los paquetes serán enrutados o no, de acuerdo a las reglas de filtrado.

Cuando un enrutador con capacidades de filtrado recibe un paquete, el enrutador obtiene cierta información del encabezado del mismo y de acuerdo a las reglas de filtrado toma decisiones, que finalizaran en el reenvío de paquete o en que este sea descartado (Ver Figura 2.0).

---

[8]Componentes de un sistema firewall. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 10 de enero del 2016, 9:13 am.]

Los datos útiles para este proceso son:

- ✓ Dirección IP origen
- ✓ Dirección IP destino
- ✓ Puerto TCP/UDP origen
- ✓ Puerto TCP/UDP destino
- ✓ Tipo de mensaje ICMP
- ✓ Información del protocolo de encapsulamiento (TCP, UDP, ICMP o IP túnel)

Las reglas para el filtrado de paquetes deben estar basadas en las políticas de seguridad, establecidas para proteger de posibles ataques a la información de la organización. [9]

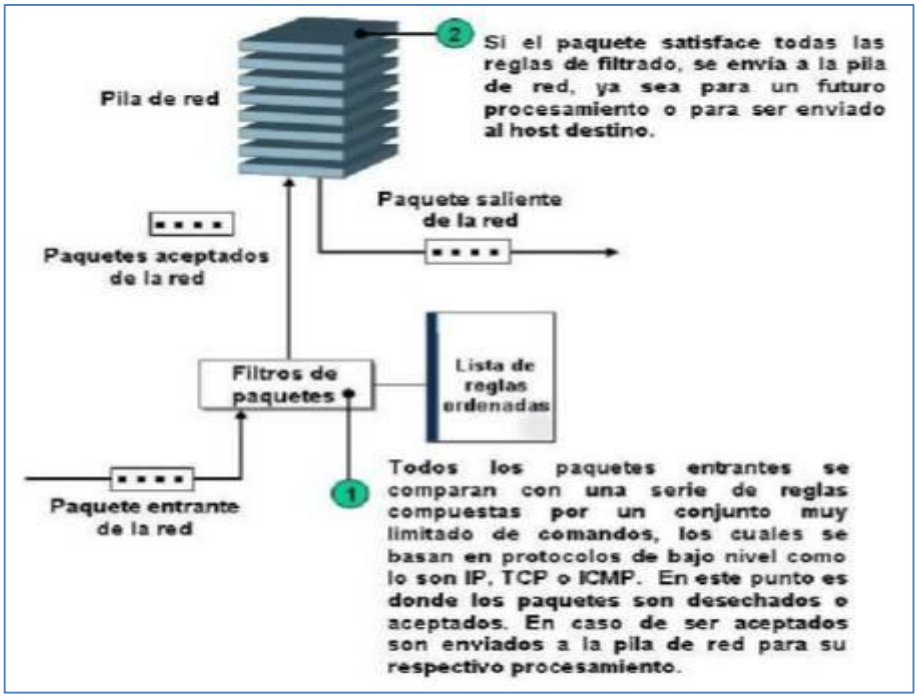



Figura 2.0 Arquitectura Básica de un Enrutador con Capacidades de Filtrado

---

[9]Enrutador con capacidades de filtrado. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 10 de enero del 2016, 4:06 pm.]

### 2.2.4. Tipos de filtrado de un enrutador

Los tipos más comunes de filtrado llevados a cabo por un enrutador son:

- Filtrado por nivel de servicio. Puesto que la mayoría de las aplicaciones utilizan los llamados "puertos bien conocidos", es posible permitir o negar servicios usando este dato para llevar a cabo el filtrado.

- Filtrado por origen o destino. Un enrutador a través de las reglas de filtrado de paquetes puede permitir o negar el reenvío de un paquete basándose en la información origen o destino del encabezado.

- Filtros avanzados. Actualmente existen diferentes tipos de ataques que pueden burlar los filtros más comunes, por lo que se tienen que establecer diferentes tipos de filtros que se basen en otro tipo de información como los campos del encabezado IP, el número de fragmentos, la longitud del paquete, etc. [10]

### 2.2.5. Servidor de control a nivel de circuito

Es una aplicación que permite validar un paquete, para reconocer si es una solicitud de conexión o un mensaje de datos perteneciente a una conexión ya establecida, entre un par de protocolos de transporte.

Para validar una sesión, un servidor de control a nivel circuito examina cada establecimiento de conexión a nivel transporte, para asegurarse que lo siguió legítimamente a través del llamado triple saludo o handshake.

---

[10]Tipos de filtrado de un enrutador. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 12 de enero del 2016, 3:09 am.]

El único protocolo de capa de transporte que utiliza el triple saludo es TCP. Los paquetes de datos no son enviados hasta que este proceso sea completado.

El firewall mantiene una tabla de conexiones validas, que incluye el estatus completo de las sesiones y la secuencia del envío de la información, y sólo permite que los paquetes de red sean reenviados cuando la información de dicho paquete concuerda con la información de la tabla de circuitos.

Una vez que la conexión es terminada, la entrada correspondiente es borrada de la tabla, y el circuito virtual entre las capas de transporte de los dispositivos que se estaban comunicando es cerrada.

Cuando una conexión se establece, el servidor de control a nivel circuito normalmente guarda en su tabla la siguiente información:

- Un identificador único por conexión, que es usado para propósitos de seguimiento.
- El estado de la conexión: handshake, establecida o cerrada.
- La información de secuencia de los paquetes.
- La dirección IP origen.
- La dirección IP destino.
- La interfaz física por la cual el paquete fue recibido.
- La interfaz física por la cual el paquete será reenviado.

Conforme a estos datos, el firewall puede determinar si el destinatario y el receptor son válidos para dicha sesión, sí se está intentando realizar un ataque spoofing detectando si el número de secuencia de la sesión es válido para la misma o no, así como si quien está intentado iniciar la sesión lo tiene permitido.

Una clara desventaja es que sólo es válido para sesiones TCP.[11]

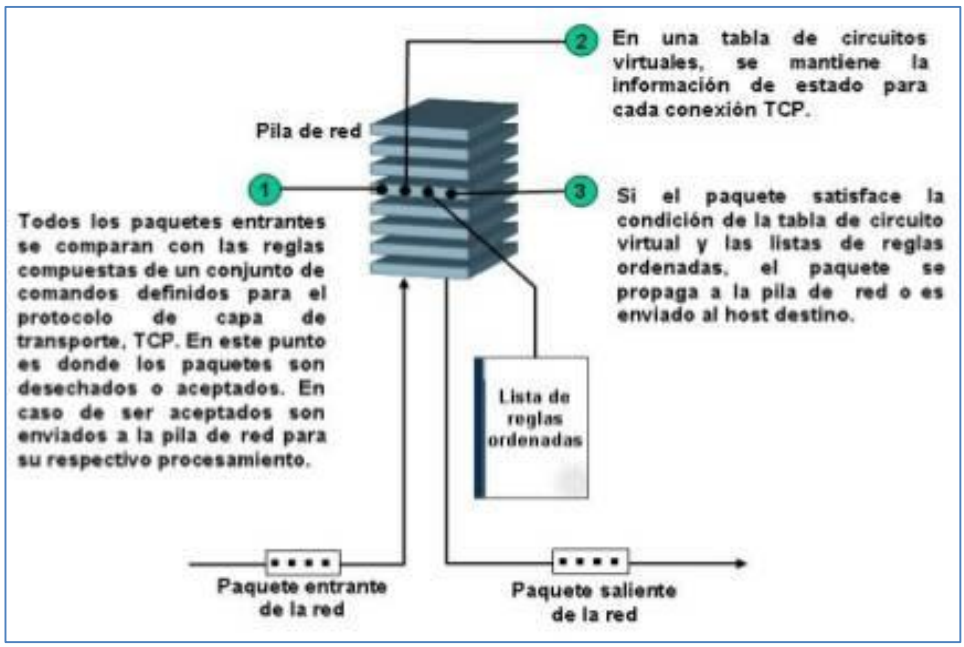



Figura 3.0 Arquitectura de un servidor de control a nivel circuito

### 2.2.6. Servidor de control de aplicaciones (Proxy)

Muchas veces referido como Proxy, el servidor de control de aplicaciones provee un control de alto nivel entre el tráfico de dos redes, en las que el contenido de un servicio en particular puede ser monitoreado y filtrado de acuerdo a las políticas de seguridad establecidas.

El Proxy actúa como el servidor real para el cliente y como cliente para el servidor destino real, es decir, se establece a través del Proxy una conexión virtual entre el cliente real y el servidor real.

---

[11]Servidor de control a nivel circuito. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 12 de enero del 2016, 5:06 pm.]

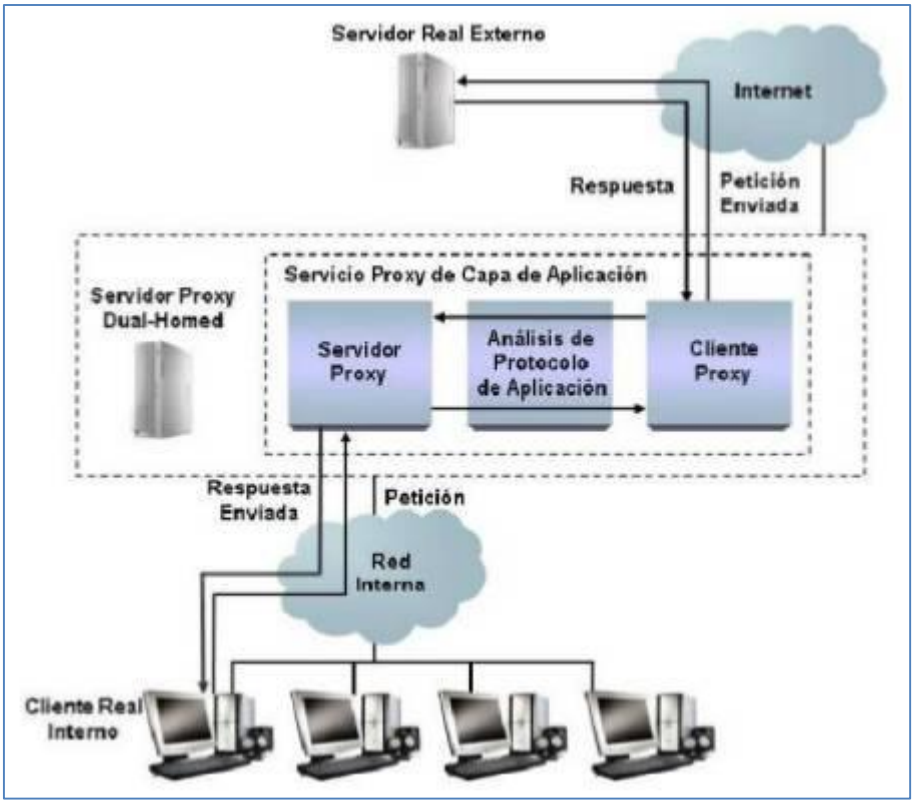



Figura 4.0 Servidor Proxy

El Proxy es capaz de monitorear y filtrar cualquier tipo de datos. Por ejemplo, un servidor FTP puede ser accesible desde el exterior, sin embargo, en orden de protegerlo de cualquier posible ataque, se puede colocar un servidor Proxy que niegue los comandos PUT y MPUT.

Un servidor Proxy es un servidor corriendo una aplicación servidor (demonio) específico, donde este dispositivo es accesible tanto desde la red segura como de la no segura. El propósito de este dispositivo es controlar el intercambio de datos entre dos redes a nivel de la capa de aplicación.

Usando un servidor Proxy, es posible hacer un filtrado específico para que las conexiones destinadas al protocolo de aplicación que maneja el servidor Proxy sólo sean entregadas al mismo.

Es importante señalar que los dispositivos clientes deben permitir la configuración para soportar una conexión a través de un Proxy, sin embargo, a pesar de este extra, las ventajas de control a nivel aplicación son superiores, ya que no sólo se puede llevar a cabo el filtrado, sino que el servidor Proxy también puede funcionar como un dispositivo de autenticación independiente de la propia autenticación solicitada por el servidor real.[12]

### 2.2.7. Ejemplos de configuraciones del firewall

Un firewall consiste en uno o más elementos de software que corren en uno o más dispositivos. Estos dispositivos pueden ser sistemas de cómputo general o dispositivos más especializados como un enrutador. [13]

Existen cuatro ejemplos importantes de configuraciones de sistemas de firewall:

- Firewall por Filtrado de Paquetes.
- Firewall Dual-Homed.
- Firewall por Host de Bastión.
- Firewall por Subred.

---

[12]Servidor de control de aplicaciones (Proxy). http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 12 de enero del 2016, 9:12 am.]

[13]Ejemplos de configuraciones del firewall. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 12 de enero del 2016, 10:00 am.]

### 2.2.8. Firewall por filtrado de paquetes

Es comúnmente usado ya que se puede considerar como un firewall que no implica una inversión extra. En este tipo de firewall, se utiliza el mismo enrutador colocado entre dos redes: la red segura interna y la red insegura.

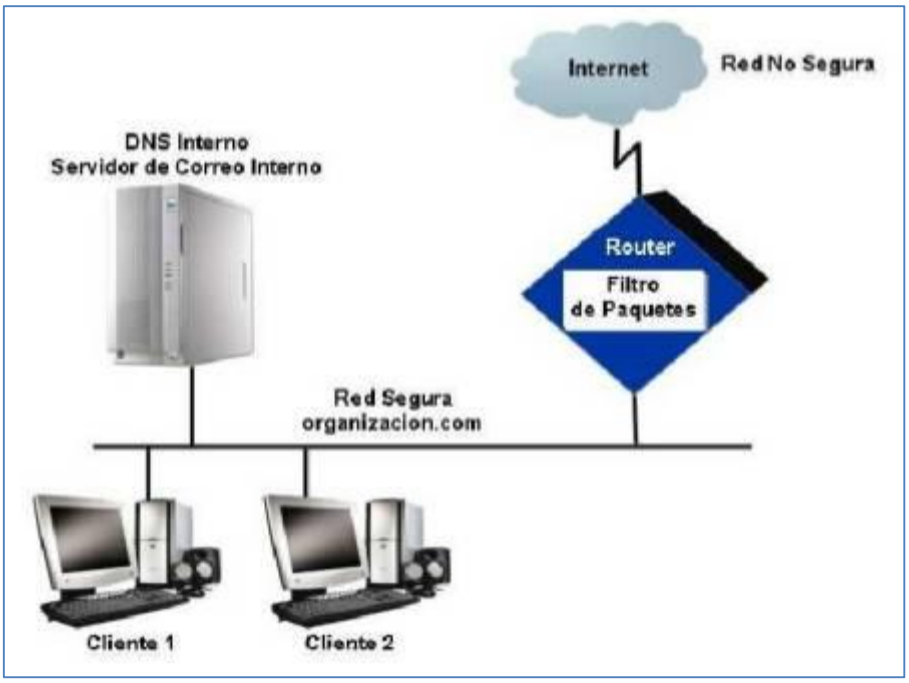



Figura 5.0 Firewall por filtrado de paquetes

Un enrutador con capacidades de filtrado se configura con las reglas de filtrado para permitir o negar el tráfico entre las redes que está interconectando. Se configura normalmente al enrutador con capacidades de filtrado para que niegue todo el tráfico que no está explícitamente permitido.

Aunque con este tipo de firewall se previenen algunos ataques potenciales, sí las reglas de filtrado no son bien configuradas se pueden dejar puertas abiertas que sean aprovechadas maliciosamente.[14]

---

[14]Firewall por filtrado de paquetes. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivo [Citado el: 12 de enero del 2016, 9:06 am.]

### 2.2.9. Firewall Dual-Homed

Este tipo de dispositivo cuenta con al menos dos tarjetas de red y por lo tanto dos direcciones IP. Normalmente se conecta una de estas interfaces al enrutador y la otra a la red LAN obligando a que todo el tráfico IP entre estas dos interfaces pase por el firewall y por lo tanto sea analizado por el mismo. Las funciones de reenvío son deshabilitadas para que solo los paquetes que cumplan con las condiciones de filtrado sean transmitidas hacia su destino (ver fig. 6.0)

Comparándolo con un firewall por filtrado de paquetes, un firewall dual-homed se asegura que todo el tráfico proveniente de o para cualquier servicio desconocido sea bloqueado pudiendo analizar hasta nivel de aplicación. Un firewall dual-homed también implementa el principio de que todo lo que no está específicamente permitido es negado.[15]

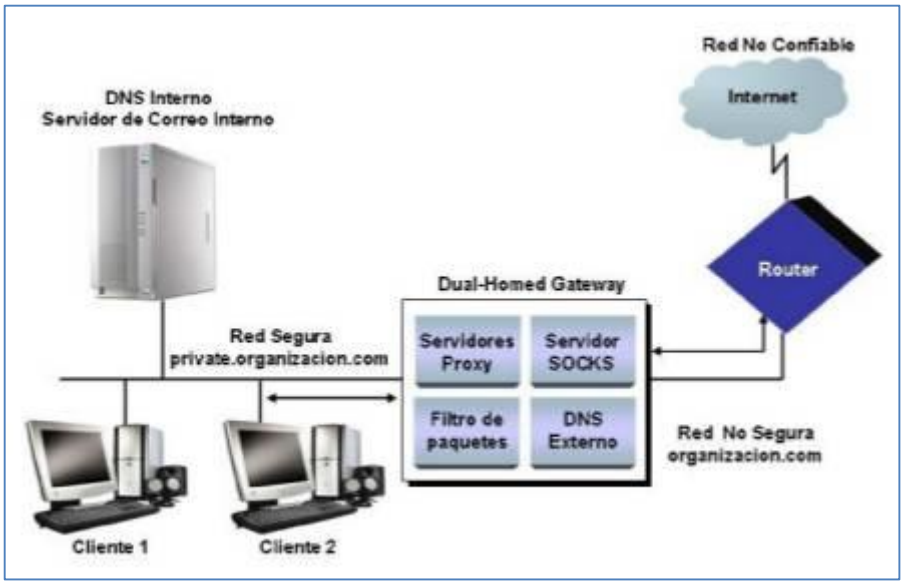



Figura 6.0 Firewall Dual-Homed

[15]Firewall Dual-Homed. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 16 de enero del 2016, 1:00 pm.]

### 2.2.10. Firewall por subred, Zona Desmilitarizada (DMZ)

Este tipo de sistema de firewall se compone de dos enrutadores con capacidades de filtrado y un host de bastión. Una DMZ provee un alto nivel de seguridad.

Se crea una zona de seguridad media o DMZ entre la red externa o insegura y la red interna o segura, donde el enrutador conectado directamente a la zona insegura sólo permite conexiones desde el exterior al host de bastión y el enrutador interno sólo permite conexiones desde el interior al mismo host.

El host de bastión es anunciado hacia el exterior con una dirección IP válida (ya sea NAT o no) y es la única dirección que conocen los usuarios externos, por lo que tendrían que romper tres barreras para poder tener acceso a la red interna.[16]

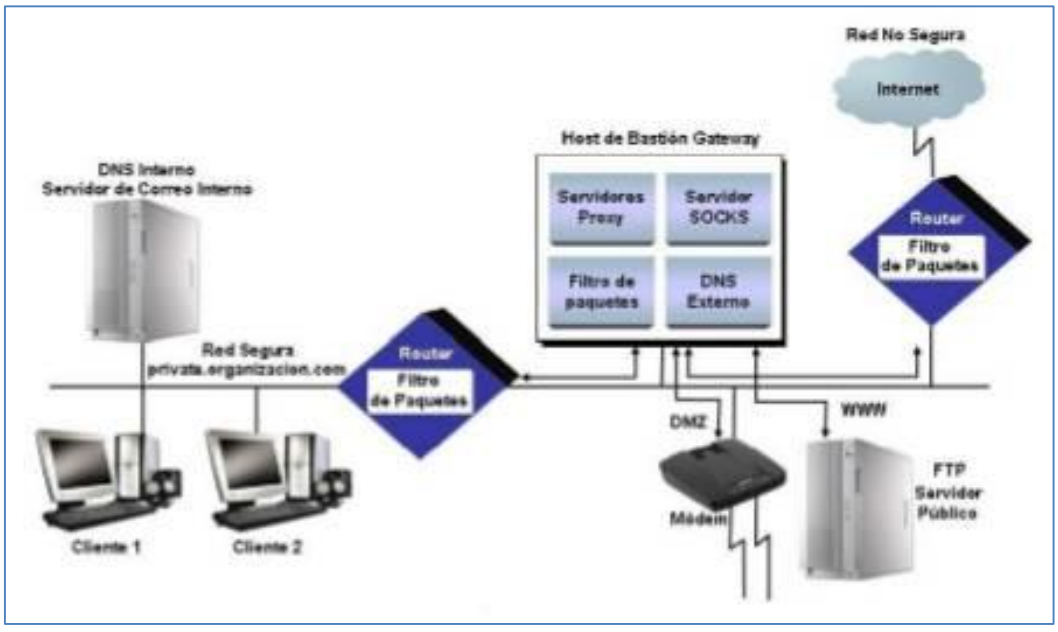



Figura 7.0 Firewall por subred, Zona Desmilitarizada (DMZ)

---

[16]Firewall por subred, Zona Desmilitarizada. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall [Citado el: 16 de enero del 2016, 3:21 pm.]

## 2.3. Bases teóricas de las redes de datos

### 2.3.1. Concepto general de las redes de datos

Dentro del mundo de las redes se manejan conceptos que permiten comprender el proceso de interacción entre los distintos dispositivos que comprenden una red. Enseguida se describen algunos conceptos que frecuentemente se encuentran relacionados con las redes de datos.[17]

### 2.3.2. Dirección IP

Cuando un equipo de cómputo desea enviar un paquete a otro equipo, dicho paquete deberá contener la dirección destino y origen, ya que sin esto no sería posible que el paquete llegue a su destino. La dirección IP (Internet Protocol – Protocolo de internet) contiene la información necesaria para enrutar un paquete a través de la red TCP/IP (Transmission Control Protocol/Internet Protocol – Protocolo de control de transmisión/ Protocolo de Internet).

La dirección IPv4 es representada por 32 bits, dicha dirección comúnmente es representada utilizando notación decimal, se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal máximo de cada octeto es 255, reservando este número para envió broadcast.

Los campos que componen a una dirección IP son identificador de red (Network ID) e identificador de host (host ID), el número de red de una dirección IP identifica la red a la cual pertenece dicho dispositivo. El host ID de una dirección IP identifica el dispositivo específico de una red.

---

[17]Concepto general de las redes de datos. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 20 de enero del 2016, 3:00 pm.]

Existen tres clases de direcciones IP A, B, C, que una organización puede recibir de parte del Registro Americano de Números de Internet (ARIN) o ISP de la organización (Ver tabla 1.0).[18]

Tabla 1.0. Direcciones de IP

| Tipo | Dirección menor | Dirección más alta | Máscara de Red | Numero de host por red |
|---|---|---|---|---|
| A | 0.0.0.0 | 126.0.0.0 | 255.0.0.0 | 16,777,214 |
| B | 128.0.0.0 | 191.255.0.0 | 255.255.0.0 | 65,534 |
| C | 192.0.0.0 | 223.255.255.0 | 255.255.255.0 | 254 |
| D | 224.0.0.0 | 239.255.255.255 | No aplica | No aplica |
| E | 240.0.0.0 | 247.255.255.255 | No aplica | No aplica |

Adicional a estas direcciones IP, existe otra clasificación, utilizada en la asignación de IP mediante NAT, para generar intranet en las organizaciones, las direcciones privadas son:

Tabla 2.0. Direcciones IP privadas.

| Tipo | Dirección menor | Dirección más alta | Máscara de Red |
|---|---|---|---|
| A | 10.0.0.0 | 10.255.255.255 | 255.0.0.0 |
| B | 172.16.0.0 | 172.31.255.255 | 255.255.0.0 |
| C | 192.168.0.0 | 192.168.255.255 | 255.255.255.0 |

### 2.3.3. Resolución de direcciones

Cada equipo conectado a la red tiene una dirección física única, además de tener una dirección IP, por lo que cuando un equipo desea transmitir un paquete de información a otro debe existir un mecanismo que relacione ambas direcciones mencionadas.

---

[18]Direccion IP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 20 de enero del 2016, 8:21 am.]

La dirección física con la dirección IP para que la información llegue al destino correcto (Figura 8.0).

Este proceso se realiza en la capa de red (modelo OSI) o en la capa de enlace de red (modelo TCP/IP) mediante el protocolo ARP (Address Resolution Protocol – Protocolo de resolución de dirección), el cual se encarga de asociar la dirección física a una determinada dirección IP, si el equipo destino está ubicado dentro de la red local, el equipo origen envía un mensaje a todos los equipos preguntando a quién le corresponde dicha dirección IP, el equipo destino responde a dicho mensaje agregando su dirección MAC.

Sin embargo, cuando la dirección destino no pertenece al mismo segmento de red, para que un dispositivo envíe datos a la dirección MAC de un dispositivo que está ubicado en otro segmento de la red, el dispositivo origen envía los datos a un gateway por defecto. El gateway por defecto es la dirección IP de la interfaz del router conectada al mismo segmento de red física que el host origen. El host origen compara la dirección IP destino con su propia dirección IP para determinar si las dos direcciones IP se encuentran ubicadas en el mismo segmento. Si el dispositivo receptor no está ubicado en el mismo segmento, el dispositivo origen envía los datos al gateway por defecto.

Si la dirección de subred es distinta, el router responderá con su propia dirección MAC a la interfaz que se encuentra directamente conectada al segmento en el cual está ubicado el host origen. Dado que la dirección MAC no está disponible para el host destino, el router suministra su dirección MAC para obtener el paquete. Luego el router puede enviar la petición ARP (basándose en la dirección IP destino) a la subred adecuada para que se realice la entrega.

Existe otro protocolo RARP (Reverse Address Resolution Protocol –Protocolo de resolución de dirección de retorno) que funciona de manera inversa, para este caso debe existir un servidor que mantiene una base de datos de correspondencia de direcciones MAC a direcciones IP. [19]

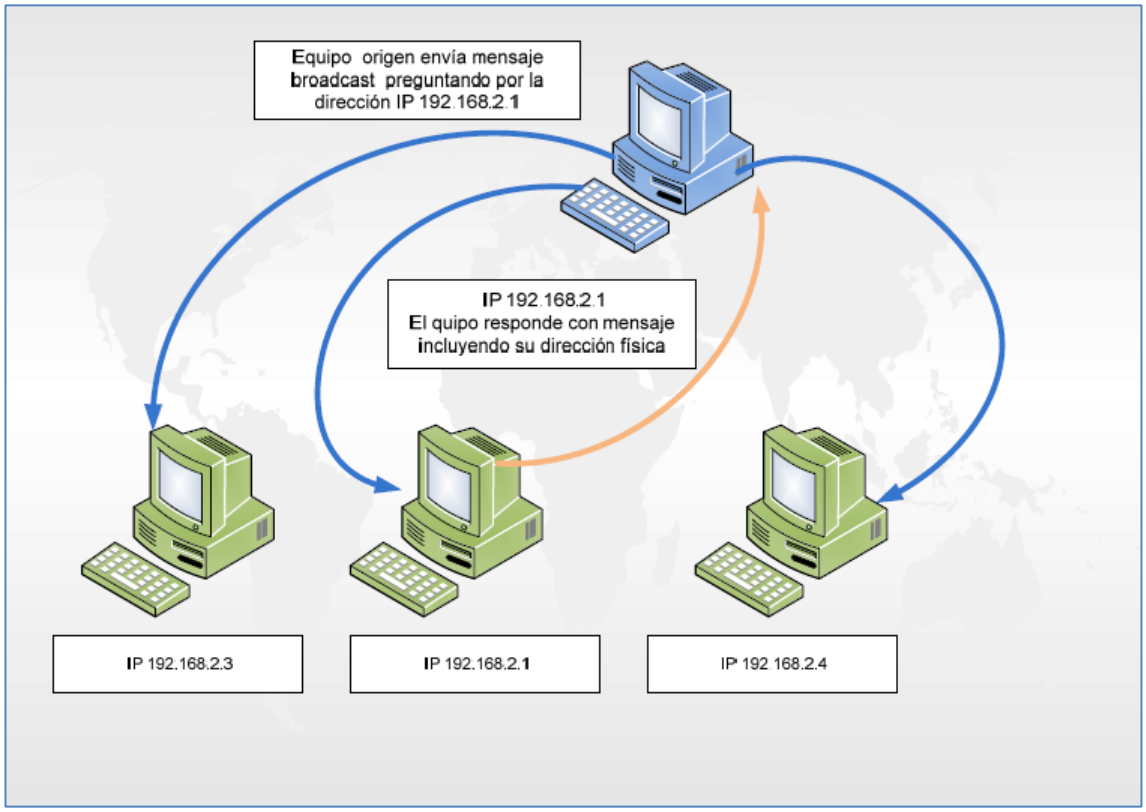



Figura 8.0 Resolución de direcciones.

### 2.3.4. Definición de protocolo

En todo proceso de comunicación es necesario seguir reglas para poder comprender el intercambio de información, por ejemplo para poder entablar una conversación es necesario que los interlocutores hablen el mismo idioma, ya que si no fuese así no podrían comunicarse entre ellos.

---

[19]Resolución de direcciones. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 24 de enero del 2016, 8:15 pm.]

Se presenta la misma situación en las redes de datos, en el momento en que dos equipos intenten comunicarse deberán seguir ciertas reglas para establecer el proceso de intercambio de información, a este proceso se le conoce como protocolo.

Un protocolo, en el contexto de las telecomunicaciones, es un conjunto de reglas que hacen que la comunicación en una red sea más eficiente. Una definición técnica de un protocolo de comunicaciones de datos es: un conjunto de normas, o un acuerdo, que determina el formato y la transmisión de datos.

Cabe mencionar que existen distintos protocolos y cada uno de ellos tiene una tarea en específico.[20]

### 2.3.5. Definición de puertos

Cuando un programa cliente necesita de un servicio particular de un servidor, además del tipo de servicio y localización del servidor, debe indicar el puerto por el que se establecerá la conexión. En este sentido, un puerto es un canal lógico de comunicación que permite a dos equipos intercambiar información.

Los puertos son representados con un valor numérico y se representan mediante una palabra de 2 bytes, por lo que existen $2^{16}$, es decir 65535 puertos, existe una organización que se encarga de regular dichos puertos conocida como IANA (Internet Assigned Numbers Authority - Agencia de asignación de números de internet), dicha organización realiza una clasificación de los puertos en:

---

[20]Definición de protocolo. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 12 de febrero del 2016, 6:37 pm.]

- Puertos bien conocidos, definidos del puerto 1 al 1023, utilizados para servicios bien conocidos como web, correo electrónico, etcétera.

- Puertos registrados, definidos del puerto 1024 al 49151, utilizados por aplicaciones conocidas y registrados en IANA, como es el caso de DB (Database – Base de datos ), escritorio remoto, RADIUS, etcétera.

- Puertos dinámicos y/o privados del puerto 49152 al 65535.[21]

### 2.3.6. Definición de puerta de enlace

El gateway o puerta de enlace es el encargado de interconectar distintas redes utilizando distintos protocolos, es el punto de la red que permite la entrada a otra red, el gateway se asocia al router (dispositivo de capa tres) sin embargo, el gateway es capaz de enlazar redes con diferentes protocolos, además de que este dispositivo puede trabajar en los siete niveles del modelo OSI.

Los gateways pueden ser personalizados para realizar una función específica, por ejemplo para una conversión de protocolos, aplicación de conversión de datos etcétera.[22]

---

[21]Definición de puertos. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 14 de febrero del 2016, 4:18 pm.]

[22]Definición de puerta de enlace. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 14 de febrero del 2016, 8:10 am.]

## 2.4. Base teóricas del Modelo OSI

### 2.4.1. Modelo OSI

Durante las últimas décadas ha habido un enorme crecimiento en la cantidad y tamaño de las redes, muchas de ellas se desarrollaron utilizando implementaciones de hadware y software diferentes, como resultado, muchas de las redes eran incompatibles y se volvió muy difícil la comunicación entre ellas, debido a que utilizaban especificaciones distintas. Para solucionar este problema, la Organización Internacional para la Normalización (ISO) realizó varias investigaciones acerca de los esquemas de red. La ISO reconoció que era necesario crear un modelo de red que pudiera ayudar a los diseñadores de red a implementar redes que pudieran comunicarse y trabajar en conjunto (interoperabilidad) y por tanto, elaboraron el modelo de referencia OSI en 1984.

El modelo de referencia OSI es el modelo utilizado para las redes de datos. Sin embargo, no es el único modelo que existe, ya que existen otros modelos de referencia como lo es el modelo TP/IP.

El modelo de referencia OSI permite que los usuarios vean las funciones de red que se producen en cada capa. Más importante aún, el modelo de referencia OSI es un marco que se puede utilizar para comprender como viaja la información a través de una red. [23]

---

[23]Modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 15 de febrero del 2016, 11:50 am.]

### 2.4.2. Esquema del modelo OSI

En el modelo de referencia OSI, existen siete capas como se muestra en la figura 9.0.[24]

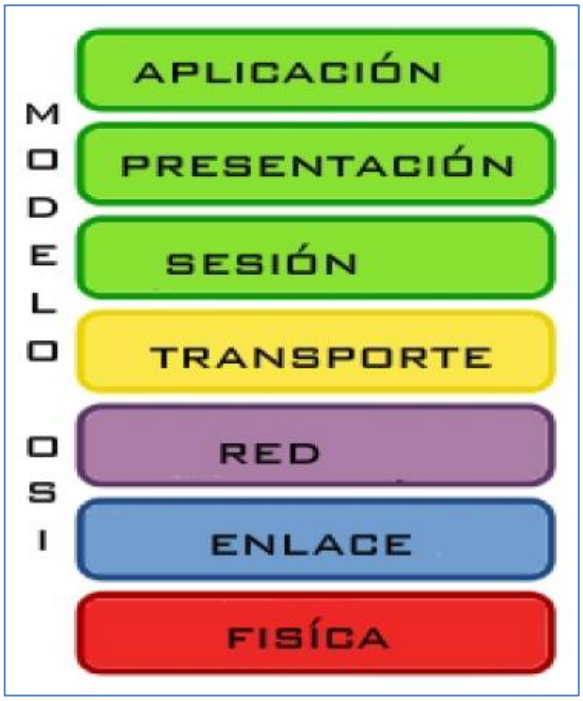



Figura 9.0 Capas del modelo OSI.

El modelo de capas permite comprender de manera sencilla el proceso de comunicación entre equipos, además de que esto implica identificar y solucionar problemas de comunicación de manera más eficiente, permite la interoperabilidad de las tecnologías, estandariza las interfaces y permite un desarrollo mucho más rápido. Las capas que conforman el modelo OSI son: aplicación, presentación, sesión, transporte, red, datos, física.

---

[24]Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 15 de febrero del 2016, 8:16 am.]

### 2.4.3. Capa física

[25] Enumerando el modelo OSI de forma ascendente, la capa física ocupa el lugar número uno, recordando que una computadora o cualquier dispositivo electrónico funciona a base de voltaje, por lo que en esta capa se considera los dispositivos encargados de transportar la señal de un dispositivo a otro y que a su vez son convertidos en ceros y unos para que el dispositivo pueda interpretarlos.

La capa física es la capa que define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre los sistemas finales.

A lo largo de la historia de las redes se han utilizado distintos medios para interconectar los equipos que conforman la red, esto debido a que se encuentran mejoras en los medios de transmisión, lo cual permite tener un mejor desempeño, desde luego el medio no es el único factor que interviene en el tiempo de retardo para trasmitir datos ya que existen otros factores.

Dentro de los elementos que componen a la capa física se encuentran los distintos medios de transmisión como son fibra óptica, cable UTP (Unshielded Twisted Pair – Cable de par trenzado) en sus distintas categorías, atmósfera que es un medio para transmitir ondas de radio además del cable coaxial que prácticamente desaparece. También ha cambiado la topología, por lo que los medios de enlace físico han evolucionado, ofreciendo nuevas ventajas como mayor alcance, mayor velocidad de transmisión, fácil instalación, etcétera, incluyendo estándares que se han desarrollado en función del tipo de red.

---

[25]Capa física del Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 16 de febrero del 2016, 3:21 pm.]

Enseguida se enumeran los componentes pertenecientes a la capa física:

- ✓ Medios de transmisión (coaxial, UTP, fibra óptica, ondas magnéticas en general, DSL, ADSL, etcétera).
- ✓ Repetidores.
- ✓ Concentradores (router, swtich, hub).
- ✓ Tarjeta de red.

### 2.4.4. Capa de enlace

Los bits que son transportados independientemente del medio que se utilice, no serían de utilidad si no fuese posible identificar quién los genera y cuál es su destino, he aquí la importancia de la capa de enlace de datos.

La capa de enlace de datos se divide en dos partes:

- ✓ Estándar LLC (Logical Link Control – Control de enlace lógico), se define en la especificación IEEE 802.2, independiente de la tecnología.
- ✓ Las partes específicas, que dependen de la tecnología e incorporan la conectividad de la capa uno.

El IEEE divide la capa de enlace OSI en dos subcapas separadas, las subcapas IEEE reconocidas son:

- ✓ Control de acceso al medio (MAC) (realizar transiciones hacia los medios).
- ✓ Control de enlace lógico (LLC) (realiza transiciones hasta la capa de red).

Estas capas son de vital importancia ya que garantizan que las tecnologías sean compatibles y que las computadoras puedan establecer la comunicación.

En la tarjeta NIC se encuentra la dirección MAC o dirección física, aunque la tarjeta de red es un dispositivo de capa uno, este dispositivo funciona en las dos capas ya que se conecta directamente con el medio físico.

La capa de enlace lógico permite que la capa de enlace de datos funcione independientemente de las tecnologías existentes, por lo que esta capa proporciona versatilidad en los servicios de los protocolos de la capa de red que está sobre ella, mientras se comunica de forma efectiva con las diversas tecnologías que están por debajo. El LLC, como subcapa, participa en el proceso de encapsulamiento. La PDU (Protocol Data Unit- Unidad de datos de protocolo) del LLC a veces se denomina paquete LLC.

LLC define la forma en que los datos son transferidos sobre el medio físico, proporcionando servicio a las capas superiores.

El LLC transporta los datos del protocolo de la red, un paquete IP, y agrega más información de control para ayudar a entregar ese paquete IP en el destino. Agrega dos componentes de direccionamiento de la especificación IEEE 802.2: el punto de acceso al servicio destino (DSAP) y el punto de acceso al servicio fuente (SSAP). Luego este paquete IP re empaquetado viaja hacia la subcapa MAC para que la tecnología específica requerida le adicione datos y lo encapsule. Un ejemplo de esta tecnología específica puede ser una de las variedades de Ethernet, Token Ring o FDDI (Fibber Distributed Data Interface – Interfaz de Datos Distribuida por Fibra).

La subcapa LLC de la capa de enlace de datos administra la comunicación entre los dispositivos a través de un solo enlace a una red. LLC se define en la especificación IEEE 802.2 y soporta tanto servicios orientados a conexión como servicios no orientados a conexión, los cuales son utilizados por los protocolos superiores. IEEE 802.2 define una serie de campos en las tramas de la capa de enlace de datos que permiten que múltiples protocolos de las capas superiores compartan un solo enlace de datos físico.

De manera general la capa dos realiza las siguientes funciones:

- ✓ Suministra un tránsito confiable de datos a través de un enlace físico.
- ✓ Usa un sistema denominado Control de acceso al medio (MAC).
- ✓ Usa la dirección MAC, que es la dirección física que se ubica en una NIC.
- ✓ Usa el entramado para organizar o agrupar los bits. [26]

### 2.4.5. Capa de red

La tercera capa del modelo OSI es la capa de red, ésta es la encargada de la navegación de los datos a través de la red, se encarga de encontrar la mejor ruta a través de la misma.

A medida que las redes crecen surge la necesidad de interconectarlas entre sí para formar nuevas redes creando el ya conocido Internet, para poder lograr esta comunicación entre las distintas redes, surge la necesidad de nuevos dispositivos encargados de realizar esta operación como los routers.

Los routers son dispositivos de interconexión que operan en la capa tres del modelo OSI. Estos dispositivos unen o interconectan segmentos de red o redes enteras. Hacen pasar paquetes de datos entre redes tomando como base la información de la capa de enlace.

Los routers cuentan con algoritmos capaces de tomar decisiones como calcular la mejor ruta de envío de datos, luego dirigen los paquetes hacia el segmento y el puerto de salida adecuados. Los routers toman paquetes de dispositivos LAN, basándose en información de la capa tres la información es enviada a través de la red, además de que estos dispositivos pueden calcular la menor ruta basándose en la densidad del tráfico y la velocidad del enlace.

---

[26]Capa de enlace del Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 20 de febrero del 2016, 2:31 pm.]

Para que un router pueda encontrar la mejor ruta, lo puede realizar de dos maneras, empleando direccionamiento plano o direccionamiento jerárquico. Un esquema de direccionamiento plano asigna a un dispositivo la siguiente dirección disponible mientras que un direccionamiento jerárquico se asigna a través de la ubicación, el protocolo Internet (IP) es la implementación más popular de un esquema de direccionamiento de red jerárquico.

Los routers requieren direcciones de red para garantizar el envío correcto de los paquetes, por lo que la dirección IP contiene la información necesaria para enrutar un paquete a través de la red. Cada dirección origen y destino que contiene está compuesta por 32 bits. [27]

### 2.4.6. Capa de transporte

Una vez que el router ha elegido la mejor ruta para el envío de datos a través de la red, pasan a la capa de transporte, la cual es la encargada de regular el flujo de información desde el origen hasta el destino de manera confiable y precisa, para llevar a cabo este proceso entran en juego dos protocolos TCP (Transmission Control Protocol – Protocolo de control de transmisión) y UDP (User Datagram Protocol – Protocolo de datagrama de usuario).

Una característica tajante que diferencia TCP de UDP es que el primero es un protocolo orientado a conexión.

#### A. Las características de TCP

- ✓ Orientado a conexión.
- ✓ Confiable.

---

[27] Capa de red - Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 10 de marzo del 2016, 4:22 pm.]

- ✓ Divide los mensajes salientes en segmentos.
- ✓ Re ensambla los mensajes en la estación destino.
- ✓ Vuelve a enviar lo que no se ha recibido.
- ✓ Re ensambla los mensajes a partir de segmentos entrantes.
- ✓ Forma parte de la pila de protocolos TCP/IP.

**B. Las características de UDP**

- ✓ No orientado a conexión.
- ✓ Poco confiable.
- ✓ Transmite mensajes (llamados datagramas del usuario).
- ✓ No utiliza acuses de recibo.

Tanto TCP como UDP utilizan diferentes puertos que les permiten dar seguimiento a la comunicación y pasar información a capas superiores. [28]

### 2.4.7. Capa de sesión

La capa de sesión establece, administra y termina las sesiones entre las aplicaciones. Esto incluye el inicio, la terminación y la re sincronización de dos computadoras que están manteniendo una "sesión". La capa de sesión coordina las aplicaciones mientras interactúan en dos hosts que se comunican entre sí. Las comunicaciones de datos se transportan a través de redes conmutadas por paquetes, al contrario de lo que ocurre con las llamadas telefónicas que se transportan a través de redes conmutadas por circuitos. La comunicación entre dos equipos involucra una gran cantidad de pequeñas conversaciones, permitiendo de esta manera que las dos computadoras participen de forma efectiva.

---

[28] Capa de transporte - Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 13 de marzo del 2016, 6:15 pm.]

Un requisito de estas conversaciones es que cada host tenga que jugar un doble papel: el de solicitar el servicio, como si fuera un cliente y el de contestar como servicio, como lo hace un servidor. La determinación del papel que están desempeñando en un preciso momento se denomina *control de diálogo*.

Las peticiones y respuestas de los equipos que han establecido una sesión son coordinadas por protocolos implementados en la capa cinco. [29]

### 2.4.8. Capa de presentación

Esta capa permite la comunicación entre aplicaciones en diversos sistemas informáticos, de tal forma que sean transparentes para las aplicaciones.

Se ocupa del formato y de la representación de datos, entre las principales funciones de esta capa se encuentran:

- ✓ Formateo de datos.
- ✓ Compresión de datos.
- ✓ Cifrado de datos.

Después de recibir los datos de la capa de aplicación, la capa de presentación ejecuta algunas funciones, o todas ellas, con los datos antes de mandarlos a la capa de sesión.

Por otro lado, en la estación receptora, la capa de presentación toma los datos de la capa de sesión y ejecuta las funciones requeridas antes de pasarlos a la capa de aplicación.

---

[29]Capa de sesión - Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 17 de marzo del 2016, 9:15 am.]

La capa de presentación también se ocupa de la compresión de los archivos.[30]

### 2.4.9. Capa de aplicación

La capa número siete del modelo OSI es llamada capa de aplicación, los usuarios finales interactúan directamente con esta capa, en ella se encuentran todos los programas con los que puede interactuar el usuario que hacen uso de la red.

Por lo que esta capa es la encargada de identificar la disponibilidad de los participantes de la comunicación, sincronizar aplicaciones y controlar la integridad de los datos. La capa de aplicación no brinda servicios a ninguna otra capa OSI. Sin embargo, brinda servicios a los procesos de aplicación que se encuentran fuera del alcance del modelo OSI. [31]

---

[30]Capa de presentación - Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 17 de marzo del 2016, 4:16 pm.]

[31]Capa de aplicación - Esquema del modelo OSI. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 17 de marzo del 2016, 11:00 am.]

## 2.5. Base teórica del Modelo TCP/IP

### 2.5.1. Introducción al modelo TCP/IP

El origen de los protocolos TCP/IP se remonta al año de 1969, por medio de un proyecto de desarrollo e investigación fundado en la Agencia de Proyectos de Investigación Avanzada (ARPA) por sus siglas en inglés, el proyecto consistía en crear un red de intercambio de paquetes la cual fue llama ARPANET, fue construida para estudiar técnicas para brindar comunicaciones de datos robusta, confiable e independientes de los vendedores.

Estas redes experimentales fueron tan exitosas que muchas de las organizaciones comenzaron a utilizarlas en sus comunicaciones de datos diariamente. En 1975 el ARPANET fue convertido de una red experimental a una red operacional y la responsabilidad para administrar esta red fue asignada a DCA (Defense Communications Agency-Agencia de defensa para las comunicaciones) ahora conocida como DISA (Defense Information Systems Agency - Agencia para la defensa de sistemas de información), división que pertenece al Departamento de Defensa de los Estados Unidos.

Los protocolos TCP/IP fueron desarrollados después de que la red se volviera operacional, estos protocolos fueron adoptados como estándares militares en 1983 y todos los nodos conectados a la red se convirtieron al nuevo protocolo. Al mismo tiempo de que TCP/IP fue adoptado como un estándar, el término Internet comenzó a ser utilizado de manera común. En 1983 el viejo ARPANET fue dividido en MILNET, la parte no clasificada de la red de datos de defensa y en un nuevo y más pequeño ARPANET, Internet fue utilizado para referirse a la red entera, actualmente internet es el término genérico utilizado para referirse a toda la red.

La popularidad de TCP/IP creció rápidamente ya que este modelo fue la base para la conexión a Internet, además de ser un protocolo estándar abierto, ser independiente de la conexión física de red (Ethernet, DSL, dial-up, fibra óptica, inalámbrica y cualquier otro medio de transmisión), además de su compatibilidad con diferentes sistemas operativos y hardware, un esquema de direccionamiento común que permite a cualquier dispositivo una dirección única que cualquier otro dispositivo en la red entera, inclusive al ser empleado en redes que no tienen conexión a Internet y una estandarización en los protocolos de capa superior. [32]

### 2.5.2. Arquitectura del protocolo TCP/IP

La arquitectura del protocolo TCP/IP está compuesta de menos capas que las siete utilizadas en el modelo OSI, las 4 capas de modelo TCP/IP se ilustran en la figura 10.0.

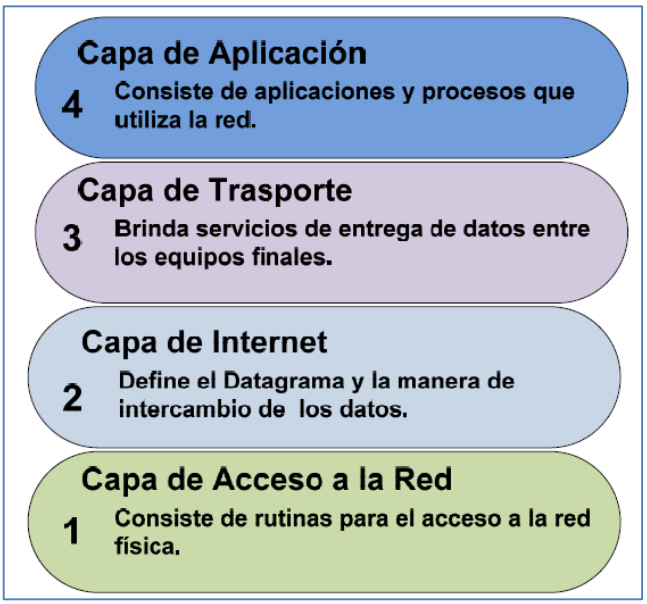



Figura 10.0 Arquitectura TCP/IP.

[32]Introducción al modelo TCP/IP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 15 de abril del 2016, 12:19 pm.]

Como en el modelo OSI, los datos pasan de la capa inferior hacia la superior, es decir, de la capa de *acceso a red* hasta llegar a la *capa de aplicación*, cada capa de la pila agrega controles a la información para garantizar la apropiada entrega, este control de información es llamado encabezado, porque éste es colocado frente a los datos que serán trasmitidos, cada capa trata toda la información que ésta recibe como datos y le agrega su propia cabecera al principio de la información, esto también es conocido como encapsulado, cuando los datos se reciben, cada capa quita su encabezado de los datos hasta llegar a la capa de aplicación, la cual interpreta los datos (figura 11.0.).

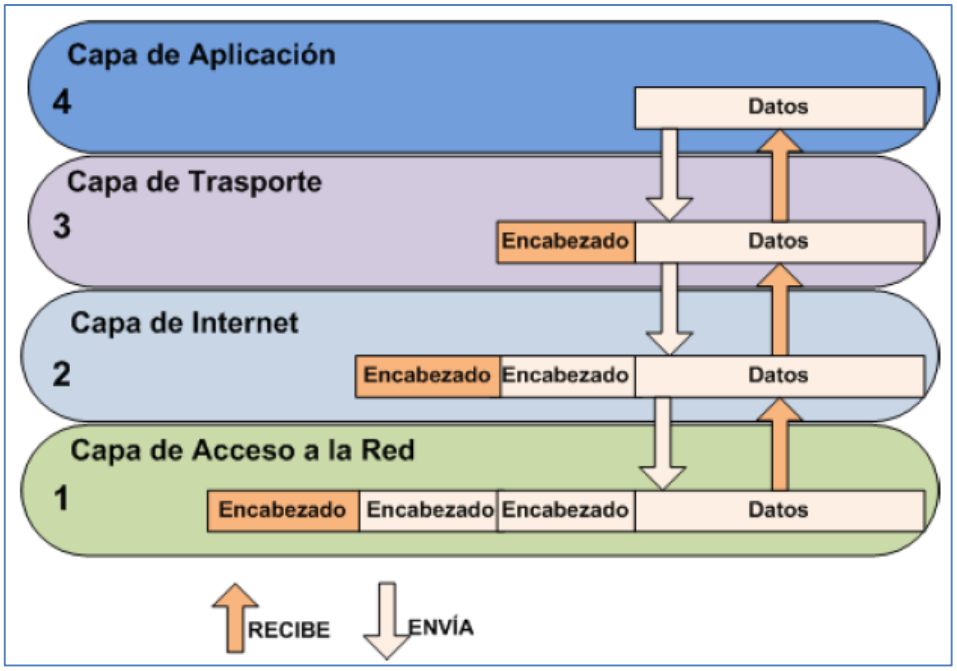



Figura 11.0 Encapsulado de los datos.

Cada capa tiene su propia estructura de datos, la cual está diseñada para ser compatible con las capas que la rodean. Mostrando los términos utilizados por las diferentes capas para definir que los datos sean enviados, las aplicaciones utilizan TCP (Transmission Control Protocol – Protocolo de control de transmisión) para referirse a datos como un flujo (stream), mientras que las aplicaciones que utilizan UDP (User Datagram Protocol Protocolo de datagrama de usuario) llaman a estos datos paquete como mensaje

(message), dentro de la capa de transporte para TCP la estructura se conoce como segmento (segment) y para UDP paquete (packet). En la capa de internet se conocen todos los bloques como datagrama tanto para TCP y UDP, y la capa de acceso a la red la estructura de datos se conoce como frame - marco (figura12.0.). [33]

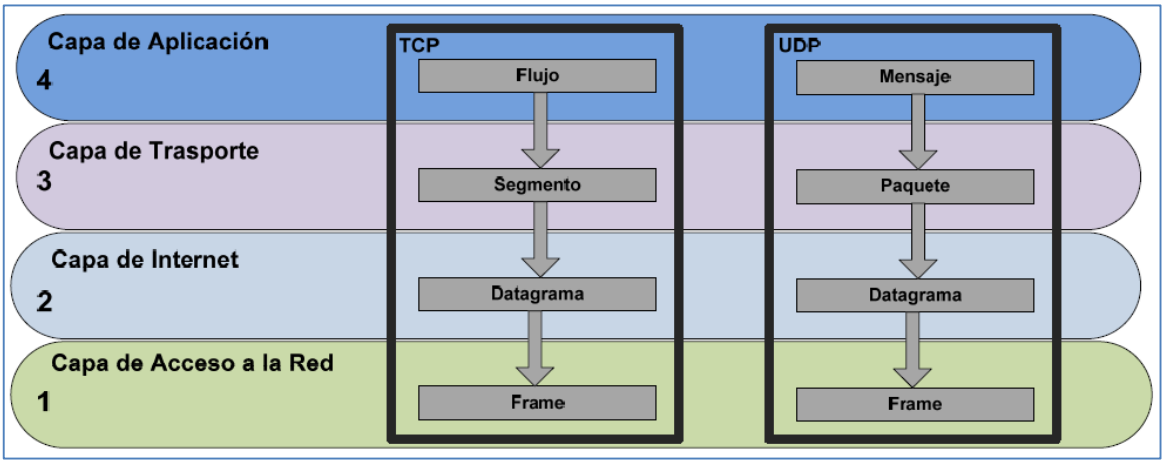



Figura 12.0 Estructura de datos.

### 2.5.3. Capa de acceso a la red

Es la capa más inferior del modelo TCP/IP, los protocolos en esta capa brindan el significado para que el sistema entregue datos a otros dispositivos, esta capa define cómo utilizar la red para trasmitir un datagrama IP, a diferencia de los protocolos de nivel superior, el protocolo de acceso a red deberá conocer los detalles de todo el paquete (estructura del paquete, dirección IP, MAC Address, etcétera) para que el dato pueda ser transmitido correctamente. Esta capa es equivalente a las 3 capas inferiores del modelo OSI (Red, Enlace de datos y Física) (figura 13.0.).

---

[33] Arquitectura del protocolo TCP/IP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 15 de abril del 2016, 9:20 am.]

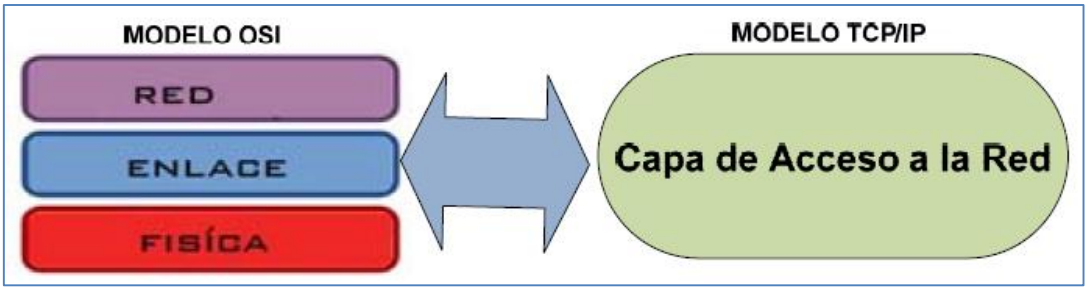



Figura 13.0 Equivalencia de la capa de acceso a la red del modelo TCP/IP con las tres primeras capas del modelo OSI.

Las funciones ejecutadas en este nivel incluyen encapsulamiento de datagramas IP dentro de los frames transmitidos, mapeo de direcciones IP a direcciones físicas *MAC Address*, dos RFC (Request For Comment – Petición de comentario) que definen protocolos en la capa de *acceso a la red* son:

- ✓ RFC 826, Address Resolution Protocol – Protocolo de resolución de direcciones (ARP).
- ✓ RFC 894, Un estándar para la transmisión de datagramas IP sobre redes Ethernet, que especifica cómo IP con encapsulados para transmitirse sobre redes Ethernet. [34]

**A. Protocolo ARP**

Mientras TCP/IP encuentra otros equipos de cómputo en la red con base en la dirección IP única de cada equipo, la transmisión de datos tuvo que ocurrir sobre algún tipo de enlace de datos, el cual debe ser debidamente identificado y relacionado con la dirección IP, la identificación de este medio se realiza por medio del protocolo ARP definido en el RFC 826, comúnmente ubicado en la capa dos del modelo OSI o en la capa uno del modelo TCP/IP.

[34]Capa de acceso a la red. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 18 de abril del 2016, 6:25 pm.]

Las direcciones utilizadas por este protocolo se conocen como MAC Address (Media Access Control Address –Dirección de control de acceso al medio), todas las tarjetas de red para redes Ethernet tiene este identificador, constituido por 48 bits o seis números en formato hexadecimal, los primeros seis números se refieren al fabricante del dispositivo y los últimos seis representan al identificador del dispositivo, también son utilizadas por algunos routers, switches, firewalls, este número es único a nivel mundial para cada dispositivo.

Cuando una máquina envía un paquete, éste es encapsulado por el protocolo IP, el cual contiene la MAC Address de la máquina que se encuentra enviando, para obtener la MAC Address del destino del paquete, se envía un paquete ARP a todo el segmento de red preguntando por algún host con base en su dirección IP, una vez que el host destino es encontrado, éste contesta enviando la relación de su IP con la MAC Address (figura 14.0).

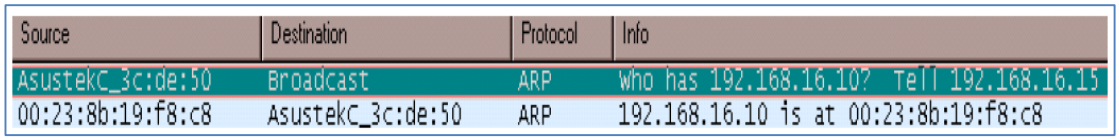


Figura 14.0 Protocolo ARP.

Es importante mencionar que debido a la manera en la que trabaja el protocolo ARP, permite la ejecución de ataques de hombre en el medio, es permitido debido a la vulnerabilidad en el diseño del protocolo. [35]

### 2.5.4. Capa de internet

La capa que sigue en jerarquía después del acceso a red es la capa de internet, en esta capa el protocolo IP (Internet Protocol) es el más importante.

---

[35]Protocolo ARP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 21 de abril del 2016, 12:21 pm.]

La versión de IP utilizada actualmente es la versión 4 (IPv4) definida en el RFC 791, actualmente se busca migrar IPv4 a una versión más reciente llamada IPv6 que se encuentra en crecimiento, ya que brinda mayores beneficios como lo es una gama más grande de direcciones, calidad en el servicio y seguridad desde el diseño, en este tema sólo se considera al protocolo IPv4 ya que es el estándar utilizado actualmente en la mayoría de las redes mundiales. [36]

### A. Protocolo IP

El protocolo IP (Internet Protocol – Protocolo de Internet), definido en el RFC 791 dentro de sus funciones incluye:

- ✓ Define el datagrama, que es la unidad básica de transmisión en Internet definida por el protocolo de internet (Internet Protocol).
- ✓ Define el esquema de direccionamiento de Internet.
- ✓ Mueve datos entre la capa de enlace a red (Network Access Layer) y la capa de transporte (Transport Layer).
- ✓ Determina la ruta a seguir para equipos en otro segmento.
- ✓ Ejecuta fragmentación de paquetes y re ensambla los mismo (cada tipo de red define su unidad de transmisión máxima).

Para realizar el intercambio de paquetes o datagramas, se hace uso de la dirección física (MAC Address) y la dirección IP determinada en esta capa, cada paquete viaja en la red independientemente de cualquier otro paquete.

---

[36]Capa de internet. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 25 de mayo del 2016, 6:27 pm.]

El datagrama es el paquete formado definido por el protocolo de internet el cual contiene una cabecera y los datos, esto implica que mensajes grandes como una enciclopedia sean fragmentados en mensajes más pequeños para su transporte, en la cabecera se tienen los datos necesarios para que el paquete pueda ser entregado, con base en la dirección IP destino, el datagrama está formado por 6 palabras de 32 bits cada una, como se muestra en la figura 15.0.

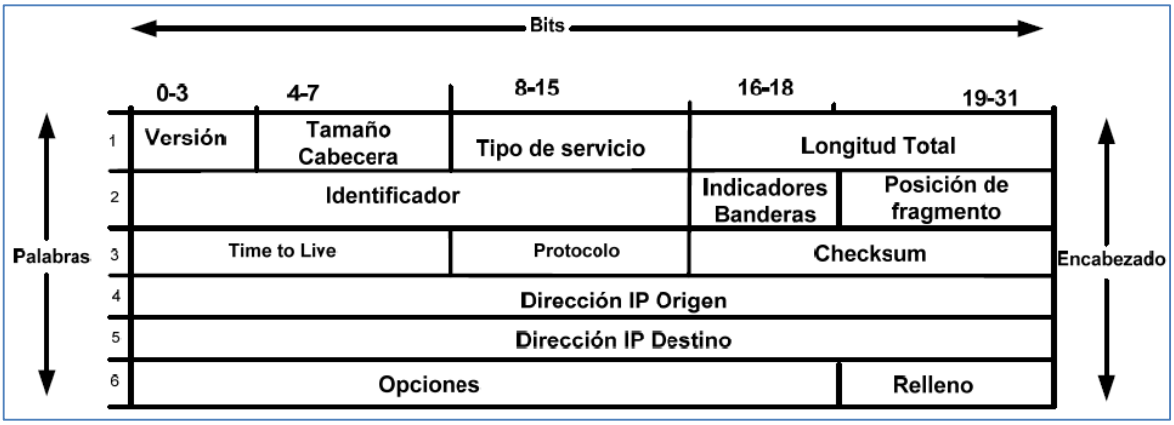



Figura 15.0 Datagrama IP.

La cabecera IP está formada por los campos que se ilustran en la imagen anterior, cada uno tiene usos específicos que ayudan a encaminar la tarea del protocolo. En caso de que el paquete se envíe a un equipo del mismo segmento, el paquete es enviado directamente al host, si el paquete va dirigido a un equipo que no es de la red local, éste es enviado al gateway para que procese su entrega. [37]

[37]Protocolo IP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 28 de mayo del 2016, 8:20 pm.]

## B. Protocolo ICMP

Una parte integral de la capa de Internet es el protocolo ICMP (Internet Control Message Protocol – Protocolo de mensaje de control de Internet) definido en el RFC 792, este tipo de protocolos realizan un seguimiento de control que determinan errores y funciones informativas de TCP/IP, además de ser un protocolo no orientado a conexión, es decir, no realiza un seguimiento de los paquetes que han sido enviados.

ICMP es un protocolo no orientado a conexión, una de sus utilidades primordiales es solucionar problemas en la red por medio de la aplicación ping, ping generalmente utiliza un paquete ICMP especial *petición – echo tipo (8)* (echo-request type (8)) en sus banderas, el cual pregunta si está activo el equipo, en caso de que el host solicitado esté disponible envía una *repetición – echo tipo (0)* (echo-replay type (0)) en sus banderas, el seguimiento de este tipo de prueba se observa en la figura 16.0.

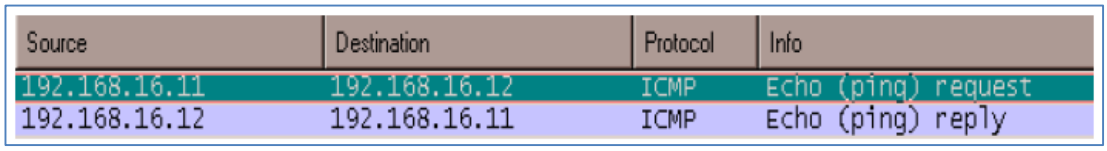



Figura 16.0 Protocolo ICMP.

Existen en total 11 tipos de mensajes ICMP, cada que hay comunicación en la capa de internet, los cuales tienen utilidades específicas, los usos principales que se le dan a este protocolo se muestran en la tabla 3.0. [38]

[38]Protocolo ICMP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 28 de mayo del 2016, 7:29 am.]

Tabla 3.0. Mensajes ICMP.

| Nombre | Descripción |
|---|---|
| Flow Control (Control de flujo). | Cuando el datagrama llega demasiado rápido para ser procesado. |
| Detecting unreachable destinations, (Detectando destinos inalcanzables). | Cuando un destino no es encontrado. |
| Redireting routers (redireccionando rutas). | Es enviado para avisar a un host que utilice otro Gateway, posiblemente por mejor elección. |
| Checking remote host (checando host remoto). | Para verificar si un host remoto se encuentra en operación. |

### 2.5.5. Capa de transporte

Después de la capa de Internet, está definida la capa de transporte equipo a equipo - Host to Host Transport Layer, usualmente conocida como capa de transporte, los protocolos más importantes en esta capa son protocolo de control de transmisión *TCP* y protocolo de datagrama de usuario *UDP*.

*TCP* brinda servicio de entrega de datos confiable en cada punto final con detección y corrección de errores, a diferencia de *UDP* que brinda un servicio de entrega de datos sin conexión, ambos protocolos entregan datos entre la capa de aplicación y la capa de Internet. [39]

[39]Capa de transporte. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 30 de mayo del 2016, 4:56 pm.]

## A. Protocolo TCP

El protocolo *TCP* es utilizado en aplicaciones que requieren la garantía de entrega en sus paquetes, verificando que los datos sean entregados, por lo tanto *TCP* es un protocolo orientado a conexión, la unidad de datos intercambiada entre cada módulo de datos *TCP* es llamada segmento, cada segmento contiene un *checksum* (suma de verificación) para verificar que los datos no tengan daño, si el segmento enviado contiene daños, éste es rechazado hasta recibir un segmento en buen estado, el encabezado de este protocolo es de 32 bits formado por 6 palabras (figura 17.0).

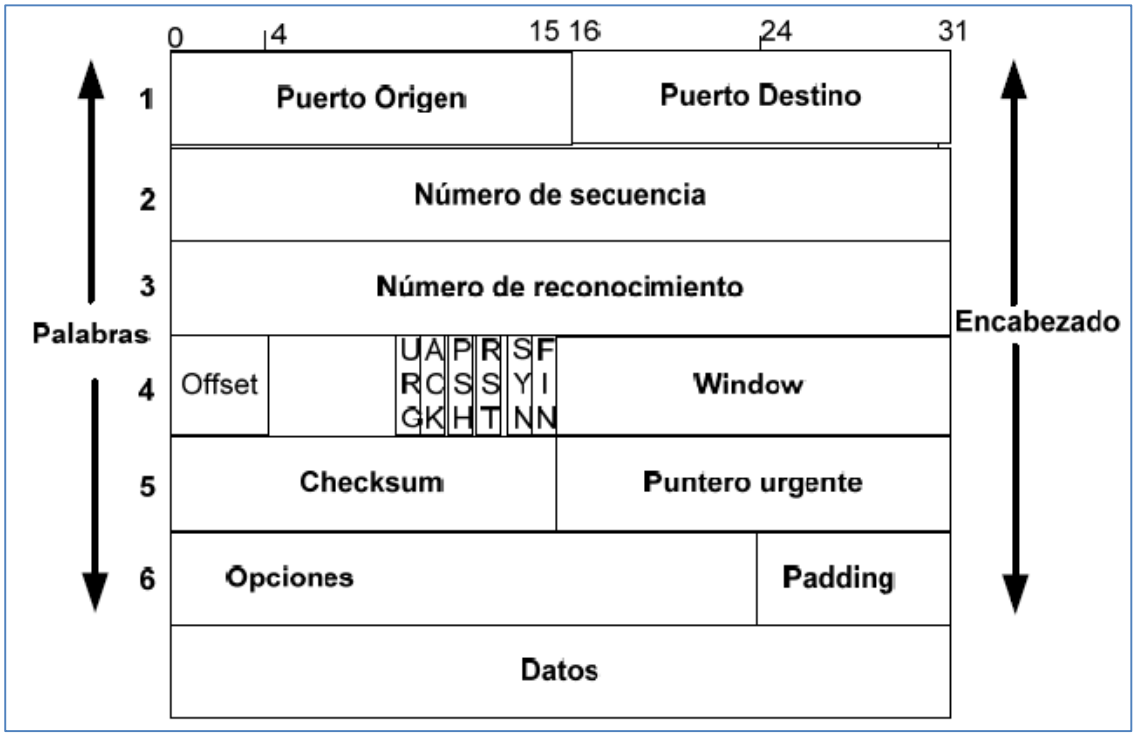



Figura 17.0 Cabecera TCP.

El protocolo TCP establece una conexión punto final a punto final, entre dos equipos, la información de control de esta conexión recibe el nombre de handshake (saludo de mano) es un intercambio entre los 2 puntos finales para establecer un diálogo. El tipo de handshake utilizado por *TCP* recibe el

nombre de Three-way handshake o (Saludo de *3* vías), porque tres segmentos son intercambiados.

*TCP* ve los datos que envía como un flujo continuo de bytes, no como paquetes independientes, por lo tanto *TCP* tiene cuidado en mantener la secuencia en que los datos son enviados y recibidos. El estándar *TCP* no requiere que cada sistema comience numerando los bytes con un número específico, cada sistema elige el número que éste utilizará como punto de comienzo, para mantener el flujo de datos correctamente, cada punto final debe conocer el *ISN* (Initial Sequence Number - número inicial de secuencia) del otro punto final, por razones de seguridad este número es elegido aleatoriamente. [40]

**B. Protocolo UDP**

El protocolo *UDP* definido en el *RFC 768*, permite la entrega de datagramas con un mínimo de carga, es un protocolo sin garantía de entrega y no orientado a conexión, es decir, no tiene mecanismos para verificar que la información ha sido entregada al punto final, utiliza un encabezado de 32 bits donde define el puerto origen y destino en una palabra, cada uno utilizando 16 bits, el formato de mensajes *UDP* es el siguiente (figura 18.0).

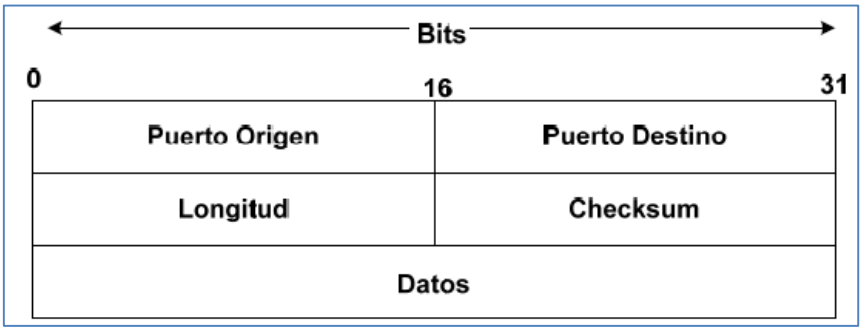



Figura 18.0 Cabecera UDP.

[40]Protocolo TCP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 30 de mayo del 2016, 1:36 pm.]

Uno de los protocolos más conocidos el cual emplea como transporte *UDP* es el protocolo *DNS*, debido a las ventajas de transporte que éste brinda para su finalidad. [41]

### 2.5.6. Capa de aplicación

En la parte superior de la arquitectura *TCP/IP* se encuentra la capa de aplicación, esta capa incluye todos los procesos que utilizan los protocolos de la capa de transporte para la entrega de datos, aquí existen muchos protocolos de aplicación, muchos de éstos brindan servicios a los usuarios, suelen generarse nuevos servicios que se agregan continuamente a esta capa.

Los protocolos más conocidos e implementados son TELNET*, FTP, HTTP, SMTP, SMTP, POP3, DNS, SSH DHCP, NFS*, entre muchos otros que se generan con el paso del tiempo y las necesidades que surgen en el mismo. [42]

#### A. Protocolo TELNET

Este protocolo está definido en el RFC 854 desde el año de 1983, conocido como *The Network Terminal Protocol – Protocolo de terminal para la red,* protocolo orientado a conexión utilizando por lo tanto transporte TCP, brinda autenticación remota sobre la red, es un protocolo inseguro ya que toda la información que viaja por este protocolo está en claro, es decir se puede interpretar todos los datos que fluyen, por lo tanto es punto fácil de ataque, actualmente ya es un protocolo muy poco utilizado pero algunos dispositivos y sistemas lo siguen utilizando como mecanismo de acceso remoto.

---

[41]Protocolo UDP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 30 de mayo del 2016, 3:35 pm.]

[42]Capa de aplicación. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 30 de mayo del 2016, 3:40 pm.]

El protocolo TELNET le asigna al servidor el puerto 23 *TCP*, el cliente puede elegir el que desee, mayor a 1024, es muy útil cuando se desea ver alguna información que brinda un servicio por algún otro puerto conocido. [43]

**B. Protocolo HTTP**

Hypertext Transfer Protocol – Protocolo de transferencia de hipertexto, definido en el RFC 2616, este protocolo es el encargado de traducir el código de los documentos *HTML* en páginas web, fue utilizado por World Wide Web – Red global mundial, desde 1990, especificado como *HTTP* /1.1, está diseñado para el acceso público, considera que la seguridad no es importante (figura 19.0).

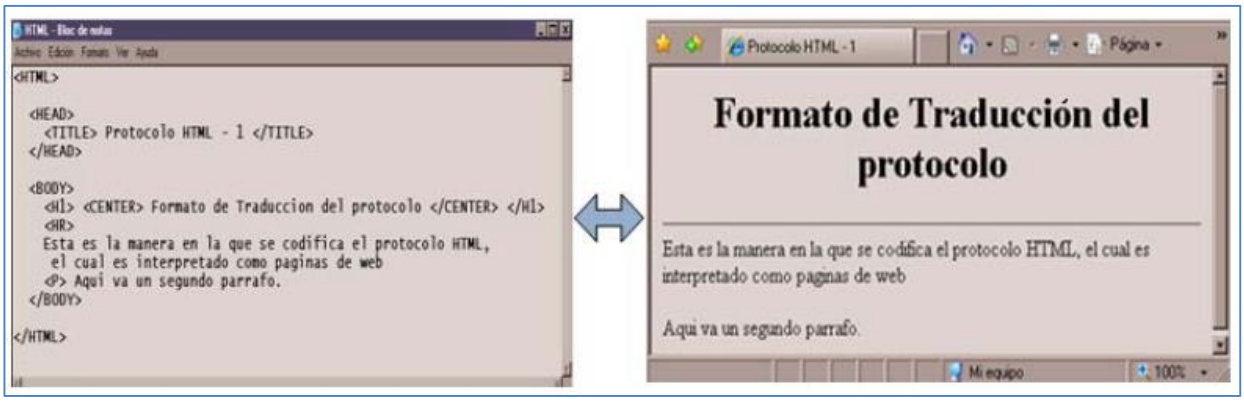



Figura 19.0 Protocolo HTTP.

Es un protocolo orientado a conexión, utiliza el protocolo *TCP* y su servicio se brinda en el puerto 80, actualmente este protocolo combina el protocolo *HTML* con otros lenguajes de programación como aspx, php, jsp, entre otros, razón por la cual, el protocolo se vuelve más vulnerable al añadirle las vulnerabilidades propias de cada uno de los lenguajes de programación. [44]

---

[43]Protocolo TELNET. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 31 de mayo del 2016, 4:20 pm.]

[44]Protocolo HTTP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 02 de junio del 2016, 4 50 pm.]

## C. Protocolo DNS

Todas las máquinas que trabaja sobre *TCP/IP* deben tener una dirección *IP* única para comunicarse con los otros hosts, las computadoras operan fácilmente con direcciones *IP,* a diferencia de las personas que les es más sencillo aprender un nombre, por esta razón los usuarios deberán identificar las direcciones *IP* por un nombre. En los inicios cercanos a *ARPANET* y comienzo de internet, el número de nombres de equipos conectados a la red fue pequeño, por lo tanto la traducción de nombres se realizaba por medio de un archivo llamado *HOST.TXT* que contenía el nombre y direcciones de cada host, este archivo fue alojado en un servidor a cargo de The Network Information Center – Centro de información sobre la red (*NIC*) del Instituto de Investigaciones de Stanford, como la red *ARPANET* siguió creciendo, se creó la necesidad de generar el concepto de servicio de distribución de nombres y dio origen al protocolo *DNS* (Domain Name Server –Servidor de nombres de dominio), el cual fue una manera más eficiente de distribuir los nombres de dominio, esta arquitectura fue liberada en el RFC 882 y 883. (Figuras 20.1 y 20.2).

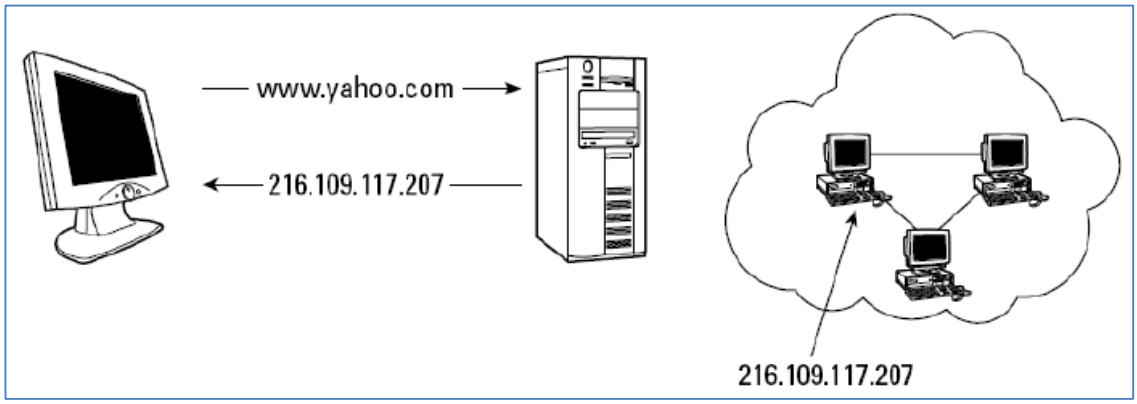



Figura 20.1 Traducción DNS.

Este protocolo es un protocolo de traducción de nombres de dominio a direcciones *IP* y viceversa, aplicado a todos los servicios que hagan uso de los nombres de dominio como es *FTP*, *HTTP*, *SMTP*, *NetBIOS* y muchos más, está definido en los *RFC's* 1034 y 1035 actualmente, es un protocolo no

orientado a conexión UDP el cual utiliza el puerto 53 del lado del servidor para atender las consultas, hoy en día a nivel mundial se cuenta con 13 servidores *DNS* raíz registrados en http://www.root-servers.org/.

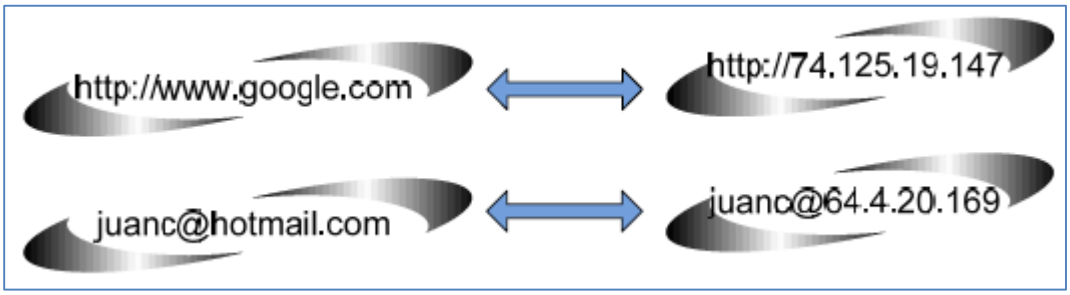



Figura 20.2 Ejemplo de traducción DNS.

Dentro de los ataques a este protocolo se tiene la modificación de la base de datos del servidor, suplantación y denegación de servicio principalmente, ataques que se explicarán en capítulos posteriores. [45]

**D. Protocolo DHCP**

El protocolo *DHCP* (Dynamic Host Configuration Protocol –Protocolo de configuración dinámica de host) es utilizado para asignar un conjunto de configuraciones de red, de manera centralizada y dinámica, evitando al usuario o administrador configurar los datos de cada equipo de manera manual, es un protocolo definido en el RFC 2131, no orientado a conexión UDP hace uso del protocolo *BOOTP* (Bootstrap Protocol – Protocolo Bootstrap) que es un protocolo de configuración de host anterior a *DHCP* resolviendo las limitaciones propias de *BOOTP*, ambos protocolos utilizan el puerto 67 para atender peticiones, los clientes normalmente reservan el puerto 68 dentro de los parámetros que determina este protocolo se encuentran dirección *IP*, servidor *DNS*, gateway, máscara de red.

[45]Protocolo DNS. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 11 de junio del 2016, 6:30 pm.]

Este protocolo fue pensado para equipos de red que cambian continuamente de ubicación, dentro de sus vulnerabilidades se tiene el seguimiento de actividades por parte de un cliente, denegación de servicio así como la posible suplantación de un servidor *DHCP* para varios propósitos (Figura 21.0). [46]

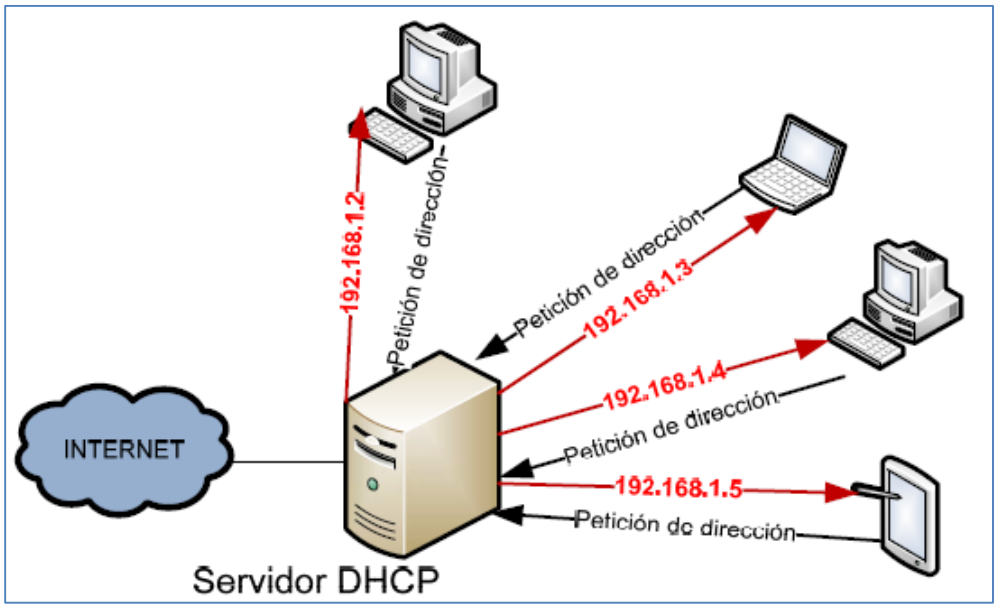



Figura 21.0 DHCP.

[46]Protocolo DHCP. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 11 de junio del 2016, 8:20 am.]

## 2.6. Base teóricas sobre la transmisión de datos

### 2.6.1. Unidades de medida de transmisión de datos

La velocidad de transmisión de datos mide el tiempo que tarda un host o un servidor en poner en la línea de transmisión el paquete de datos a enviar. Aquí se utilizan múltiplos de 10, por unidad de tiempo. Lo que lleva a expresarlos en bits/segundo (b/s o también bps), o en octetos o Bytes (B/s).

En este sentido hay que tener en cuenta que las velocidades que en la mayoría de las ocasiones se muestran en Internet están expresadas en KB/s (Kilobyte por segundo), lo que realmente supone que nos dice la cantidad de bytes (unidad de almacenamiento) que hemos recibido en un segundo, NO la velocidad de trasmisión. Podemos calcular esa velocidad de transmisión (para pasarla a Kbps o Kilobits por segundo) simplemente multiplicando el dato que se nos muestra por 8, por lo que una trasmisión que se nos indica como de 308 KB/s corresponde a una velocidad de transmisión de 2.464 Kbps, a lo que es lo mismo, 2.64 Mbps.

Estas son las unidades de medida utilizadas para la velocidad de transmisión de datos:

- ✓ 1 bps = 1 bit por segundo
- ✓ 1 Kbps = 1000 bps
- ✓ 1 Mbps = 1000 Kbps
- ✓ 1 Gbps = 1000 Mbps

En este ejemplo vemos que la velocidad está expresada en Kilobytes por segundo, eso quiere decir que se están transmitiendo 331 KB de información en 1 segundo. Si se requiere saber la velocidad de transmisión de cada bit que compone un Byte de información, debe multiplicarse por 8 para obtener dicha velocidad expresada en Mbps. [47]


Figura 22.0 Unidades de velocidad de transmisión de información.

[47]Unidades de medida de transmisión de datos. https://cualquiercosadetecnologia.wordpress.com/2013/10/17/unidades-de-medida-de-transmision-de-datos/ [Citado el: 15 de junio del 2016, 4:25 pm.]

### 2.6.2. Medición del ancho de banda de la red

Para ser más exactos, iperf permite medir el ancho de banda entre dos hosts usándolo en modo cliente-servidor y con tcp o udp como protocolos de conexión.

Muchas veces a los administradores e ingenieros de red nos llegan tests de velocidad de clientes “expertos” que pretenden medir el ancho de banda entre dos hosts usando una copia de ficheros por scp.

Lo que no saben éstos es que scp incluye cabeceras ssh que ralentizan la transferencia precisamente por el cifrado que se realiza. SCP es un software para copiar ficheros usando un canal cifrado de ssh. Scp y ftp no son herramientas de medición de red. Además, añadir en la medición el acceso a disco (por la copia) significa incluir un cuello de botella en la “medición”. En los otros protocolos de transferencia de ficheros que no van cifrados, si pasan por un firewall, se puede estar inspeccionando los paquetes a nivel de aplicación, por lo que además, introducimos otros factores en la “medición”.

Iperf no cifra, por defecto no usa disco, simplemente se conecta a un socket y transfiere datos desde el sistema operativo.

Los comandos de iperf son comunes en linux y windows. Iperf permite muchos flags para especificar si el transporte lo hacemos sobre udp, tcp, para fijar la frecuencia de muestreo de estadísticas, el buffer, el tiempo máximo de prueba, tamaño de ventana tcp y un largo etcétera.

**Consideraciones importantes:** iperf trabaja en modo cliente-servidor, un extremo es el cliente y el otro extremo es el servidor.

Entre los extremos no deben estar filtrados los puertos de testeo de iperf. El servidor, por defecto, escucha en el tcp/5001, así que si no funciona, hay que revisar firewall y nats en el tránsito del tráfico.

El cliente iperf sube datos al servidor. Esto es, el cliente se conecta al puerto tcp/5001 del servidor (configuración por defecto) e inyecta datos en el canal de subida del cliente. Es muy importante porque, a veces, la característica cliente se asocia a descarga y en iperf no es así.

En algunas topologías, iperf puede saturar el ancho de banda entre dos redes si existe un cuello de botella. [48]

Ejemplo sobre una vlan con accesos a 100 Mbps:

```
cliente windows ( sube datos ) = 192.168.5.226

servidor linux = 192.168.5.4

comando en servidor:

[root@ratassa-vmware ~]# iperf -s -t 30 -i 5

comando en cliente windows :

C:\>iperf -c 192.168.5.4 -t 30 -i 5
```

Figura 23.1 Comandos en Windows y GNU/Linux.

[48]Medición del ancho de banda de la red. https://capa3.es/medir-el-ancho-de-banda-de-la-red-con-iperf-o-jperf.html [Citado el: 16 de Junio del 2016, 8:10 pm.]

```
-s : modo servidor

-t 30 : duración del test 30 segundos

- i 5 : intervalo de muestra de estadísticas cada 5 segundos

-c 192.168.5.4 : en el cliente ( que tiene ip 192.168.5.226 ) le decimos que se
 conecte al servidor con ip 192.168.5.4

puertos : por defecto el servidor escucha en el puerto tcp 5001 y el cliente us
a un puerto dinámico > 1024 como puerto origen
```

Figura 23.2 Explicación de los flags.

```
[root@ratassa-vmware ~]# iperf -s -t 30 -i 5
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.168.5.4 port 5001 connected with 192.168.5.226 port 46181
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 5.0 sec  56.3 MBytes  94.5 Mbits/sec
[  4]  5.0-10.0 sec  56.3 MBytes  94.5 Mbits/sec
[  4] 10.0-15.0 sec  56.3 MBytes  94.5 Mbits/sec
[  4] 15.0-20.0 sec  56.3 MBytes  94.5 Mbits/sec
[  4] 20.0-25.0 sec  56.3 MBytes  94.5 Mbits/sec
[  4]  0.0-29.8 sec   336 MBytes  94.5 Mbits/sec
```

Figura 23.3 Resultados vistos en el servidor.

```
C:\>iperf -c 192.168.5.4 -t 30 -i 5
------------------------------------------------------------
Client connecting to 192.168.5.4, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 192.168.5.226 port 46181 connected with 192.168.5.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0- 5.0 sec  56.1 MBytes  94.2 Mbits/sec
[1912]  5.0-10.0 sec  55.9 MBytes  93.8 Mbits/sec
[1912] 10.0-15.0 sec  56.2 MBytes  94.3 Mbits/sec
[1912] 15.0-20.0 sec  56.0 MBytes  93.9 Mbits/sec
[1912] 20.0-25.0 sec  55.9 MBytes  93.8 Mbits/sec
[1912] 25.0-30.0 sec  56.0 MBytes  94.0 Mbits/sec
[1912]  0.0-30.0 sec   336 MBytes  93.9 Mbits/sec
```

Figura 23.4 Resultados vistos en el host cliente.

### 2.6.3. Broadcast, Multicast y Unicast

En terminología de redes y comunicaciones hay que tener claro la diferencia entre estos tres términos relacionados con el envío de paquetes.

#### A. Unicast

El término unicast hace referencia al envío de paquetes o información desde un único emisor a un único receptor. Ejemplos básicos de aplicaciones unicast son los protocolos http, smtp, ftp o telnet. Actualmente es la forma predominante de transmisión en Internet.

En términos cotidianos, una comunicación unicast podría ser por ejemplo una llamada telefónica entre dos personas. [49]

[49]Unicast. http://rm-rf.es/broadcast-multicast-y-unicast/ [Citado el: 19 de junio del 2016, 7:20 pm.]

### B. Multicast

Multicast (multidifusión) es el envío de información en una red a múltiples receptores de forma simultánea, un emisor envía un mensaje y son varios los receptores que reciben el mismo.

Si antes hablábamos de que una comunicación unicast era una llamada telefónica entre dos personas, podemos decir que una comunicación multicast podría ser una conferencia, en la que son varias las personas que se comunican entre sí. Un ejemplo claro de comunicación multicast en Internet es un IRC (Internet Relay Chat). [50]

### C. Broadcast

Broadcast es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

En la vida cotidiana, un ejemplo de comunicación Broadcast es el de una emisora de radio, que emite señales sin saber quien la recibe, el receptor decide si recibirla o no, al igual que la señal de televisión, que se envía a todos los receptores. [51]

## 2.6.4. Velocidad de transmisión de datos

Los asuntos de velocidad y de capacidad están determinados por una cantidad de factores. Algunos están dentro de su control, otros no.

---

[50]Multicast. http://rm-rf.es/broadcast-multicast-y-unicast/ [Citado el: 19 de junio del 2016, 2:30 pm.]

[51]Broadcast. http://rm-rf.es/broadcast-multicast-y-unicast/ [Citado el: 19 de junio del 2016, 3:16 pm.]

Al igual que respecto de tantos otros asuntos de rendimiento, el evitar los cuellos de botella es un objetivo importante. La transmisión por red está regulada por el enlace más lento. Los factores que afectan la entrega por red incluyen:

- ✓ Capacidad de transporte (ancho de banda) de la red de área local.
- ✓ Ancho de banda de la conexión a Internet de la institución.
- ✓ Velocidad y capacidad del servidor de red.
- ✓ Tasa de velocidad de lectura y transferencia de datos de los dispositivos de almacenamiento.
- ✓ Tamaño de archivo de imagen.
- ✓ Demanda de usuarios en un momento dado.
- ✓ Cantidad de tráfico que compita en la red (en todos los niveles de red).
- ✓ Velocidad de cualquier paso del procesamiento “a las carreras”.
- ✓ Tiempo requerido para autenticación y otros chequeos de seguridad.
- ✓ Capacidades de la computadora del usuario final, incluyendo:
    - Velocidad de la CPU.
    - Cacheo (caching) de Ram / Disco.
    - Rendimiento del subsistema de video.
    - Velocidad de la conexión a Internet.

Existe una variedad de tecnologías de redes que se pueden encontrar entre un servidor de imágenes y el receptor final. La siguiente tabla presenta algunas de las más importantes, en orden descendiente respecto de la velocidad, medida en MB por segundo.

Tabla 4.0. Velocidades de transferencia de datos en la red.

| Tipo de red | Velocidad en MB/seg. |
|---|---|
| OC-192 | 1250 |
| OC-48 (Red de banda ancha Abilene) | 300 |
| 1000 Base T Ethernet | 125 |

| | |
|---|---|
| VBNS (Red de banda ancha NSF/MCI) | 77.8 |
| FDDI | 12.5 |
| Ethernet 100 Base T | 12.5 |
| DS-3 (T-3) | 5.6 |
| Ethernet 10BaseT | 1.25 |
| Cable módem (hacia el usuario) | 0.2 - 0.5 |
| ADSL (hacia el usuario) | 0.19 – 1 |
| DS-1 (T-1) | 0.19 |
| ISDN (uso residencial) | 0.018 |
| Módem v.90 | 0.007 |

Las más rápidas de estas redes sólo se utilizan para las redes de banda ancha de Internet más importantes. El nivel que le sigue son redes de área local, mientras que las más lentas son servicios para el consumidor. Las velocidades presentadas son máximos teóricos, que rara vez se encuentran en las instalaciones verdaderas, si llegaran a encontrarse. Observe que la red más rápida es casi 175.000 veces más rápida que la más lenta.

Una vez que uno sabe la velocidad de transmisión de una red es posible calcular el tiempo aproximado que le tomará atravesarla a un archivo de cualquier tamaño en particular. La fórmula de velocidad de transmisión es: [52]

**t = r / (v (MB/seg) x 0.8)**

Donde: t = tiempo en segundos, r = cantidad de megabytes en el archivo., v = velocidad de transmisión en MB/seg.

[52]Velocidades de transmicion de datos. https://www.library.cornell.edu/preservation/tutorial-spanish/technical/technicalD-04.html [Citado el: 23 de junio del 2016, 12:45 pm.]

## 2.7. Base teóricas de IPtables

### 2.7.1. Visión general de IPtables

El poder y la flexibilidad de Netfilter se implementa mediante la herramienta de administración de IPtables, una herramienta de línea de comandos similar en sintaxis a su predecesora, ipchains, la cual fue reemplazada por Netfilter o IPtables en el kernel de GNU/Linux 2.4 y versiones superiores.

IPtables usa el subsistema Netfilter para mejorar la conexión de redes, la inspección y el procesamiento, IPtables ofrece ingreso avanzado, acciones de pre y post-enrutamiento, traducción de direcciones de redes y el reenvío de puertos, todo en una interfaz de línea de comandos. [53]

### 2.7.2. Activación del servicio IPtables

Las reglas de cortafuegos se activan únicamente si el servicio IPtables está en ejecución. Para iniciar el servicio, use el siguiente comando:

# systemctl restart iptables

Para garantizar que iptables inicie en el arranque del sistema, use el siguiente comando:

# systemctl enable iptables

Para mostrar las reglas de iptables se ejecuta el siguiente comando:

# iptables –L –n –v

---

[53]Visión general de IPtables. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 23 de Junio del 2016, 10:20 am.]

El siguiente comando iptables ilustra la sintaxis básica del comando:

iptables –A <cadena> -j <destino>

La opción –A específica que la regla debe ser añadida a <chain>. Cada cadena comprende una o más reglas y se conoce también como conjunto de reglas.

Las tres cadenas incorporadas son ENTRADA, SALIDA, REENVÍO. Estas cadenas son permanentes y no se pueden borrar. La cadena especifica el punto en el cual se manipula el paquete.

La opción –j <target> especifica el destino de la regla; es decir, qué hacer si el paquete coincide con la regla. ACEPTAR, ENTREGAR Y RECHAZAR son ejemplos de destinos incorporados. [54]

### 2.7.3. Políticas básicas de cortafuegos

El establecimiento de políticas básicas de cortafuego crea una base para la construcción más detallada, de reglas de usuario.

Cada cadena de IPtables consta de una política predeterminada de cero o más reglas que funcionan en concierto con la política predeterminada para definir todo el conjunto de reglas para el cortafuego.

La política predeterminada para una cadena puede ser DROP o ACCEPT. Los administradores orientados a la seguridad implementan una política predeterminada de DROP y aceptan paquetes específicos en una base, de caso por caso. Por ejemplo, las siguientes políticas bloquean todos los paquetes de salida en la puerta de enlace de red:

---

[54]Activación del servicio IPtables. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 23 de junio del 2016, 11:21 am.]

# iptables –P INPUT DROP

# iptables –P OUTPUT DROP

También se recomienda denegar el tráfico de redes de paquetes reenviados – es decir que el tráfico que sea dirigido desde el cortafuego hasta su nodo de destino – también sea negado para restringir una exposición accidental a la Internet de clientes internos. Para ello, utilice la siguiente regla:

# iptables –P FORWARD DROP

Cuando haya establecido las políticas predeterminadas para cada cadena, puede crear y guardar las nuevas reglas para su red y los requerimientos de seguridad. [55]

### 2.7.4. Filtraje de IPtables comunes

Evitar que los agresores remotos accedan a una LAN es uno de los aspectos más importantes de la seguridad de red. La integridad de una LAN se debe proteger de usuarios malintencionados remotos a través del uso de reglas rigurosas de cortafuegos.

Sin embargo, con una política predeterminada para bloquear todos los paquetes entrantes, salientes y reenviados, es imposible para los usuarios de cortafuegos o puertas de enlace y para que los usuarios internos de LAN comunicarse entre sí o con los recursos externos.

Para permitir a los usuarios realizar funciones relacionadas con la red y usar aplicaciones de redes, los administradores deben abrir algunos puertos de comunicación.

---

[55]Politcas básicas del cortafuego. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 23 de junio del 2016, 9:16 am.]

Por ejemplo, para permitir el acceso al puerto 80 *en el cortafuego*, añada la siguiente regla:

# iptables –A INPUT –p tcp –m tcp --dport 80 –j ACCEPT

De esta manera los usuarios pueden navegar sitios web que se comunican mediante el puerto estándar 80. Para permitir el acceso a sitios de web seguros, debe dar acceso al puerto 443, así:

# iptables –A INPUT –p tcp –m tcp --dport 443 –j ACCEPT

A veces cuando se requiera el acceso remoto a la LAN, los servicios seguros, por ejemplo SSH, sirven para conexión remota cifrada con los servicios de LAN.

Para los administradores con recursos basados en PPP (tales como bancos o cuentas masivas ISP), el acceso telefónico se puede usar para burlar las barreras de cortafuegos de forma segura. Debido a que son conexiones directas, las conexiones de módem están típicamente detrás de un cortafuego o puerta de enlace.

Sin embargo, para usuarios remotos con conexiones de banda ancha, se pueden crear casos especiales. Puede configurar **iptables** para aceptar conexiones desde clientes remotos SSH. Por ejemplo, las siguientes reglas permiten el acceso remoto SSH:

# iptables –A INPUT –p tcp --dport 22 –j ACCEPT

# iptables –A OUTPUT –p tcp --sport 22 –j ACCEPT

Estas reglas permiten el acceso de entrada y salida para un sistema individual, tal como un computador personal conectado directamente a la Internet o a un cortafuego o puerta de enlace. No obstante, no permiten que nodos detrás del cortafuego o puerta de enlace accedan estos servicios. [56]

### 2.7.5. Reglas FORWARD y NAT

La mayoría de los ISP proporcionan un número limitado de direcciones IP dirigibles públicamente a las organizaciones que sirven.

Los administradores deben, por lo tanto, buscar otras formas de acceder a los servicios de Internet sin dar direcciones IP públicas a cada nodo en la LAN. El uso de direcciones IP privadas es la forma más común de permitir que todos los nodos en una LAN accedan correctamente a los servicios internos y externos de red.

Los enrutadores perimetrales (tales como los cortafuegos) pueden recibir transmisiones de entrada desde la Internet y dirigir los paquetes al nodo de LAN. Al mismo tiempo, los cortafuegos o puertas de enlace también pueden dirigir solicitudes de salida desde un nodo de LAN a un servicio de Internet remoto.

El reenvío de tráfico de redes puede ser peligroso a veces, especialmente con la disponibilidad de herramientas de pirateo modernas que pueden enmascarar direcciones IP *internas* y hacer que los agresores utilicen la máquina como un nodo en su LAN.

**Iptables** proporciona políticas de enrutamiento y reenvío que se pueden implementar para prevenir el uso anormal de los recursos de la red.

La cadena de **FORWARD** permite al administrador controlar a dónde se pueden dirigir los paquetes dentro de una LAN. Por ejemplo, para permitir el reenvío de toda la LAN (asumiendo que al cortafuego o puerta de enlace se le asigna una dirección IP interna en eth1), use las siguientes reglas:

---

[56]Filtraje de IPtables comunes. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 23 de junio del 2016, 3:21 pm.]

# iptables –A FORWARD –i eth1 –j ACCEPT
# iptables –A FORWARD –o eth1 –j ACCEPT

Esta regla provee acceso a la red interna detrás del cortafuego o puerta de enlace. La puerta de enlace dirige los paquetes desde un nodo de LAN a su nodo de destino, pasando todos los paquetes a través de su dispositivo **eth1**.[57]

### 2.7.6. Post-enrutamiento y enmascarado de IP

La aceptación de paquetes reenviados a través del dispositivo IP interno de cortafuegos permite que nodos de LAN se comuniquen entre sí; no obstante aún no pueden comunicarse externamente a la Internet.

Para que los nodos LAN con direcciones IP privadas puedan comunicarse con redes públicas externas, configure el cortafuego para *IP masquerading*, el cual enmascara solicitudes desde nodos LAN con la dirección IP del dispositivo externo de cortafuegos (en este caso, eth0):

# iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

Esta regla usa la tabla que concuerda con el paquete de NAT (**-t nat**) y especifica la cadena incorporada POSTROUTING para NAT (**-A POSTROUTING**) en el dispositivo de red externo de cortafuego (**-o eth0**).

POST - Enrutamiento permite la alteración de paquetes cuando están saliendo del dispositivo externo de cortafuegos.

---

[57]Reglas FORWARD y NAT. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 27 de junio del 2016 2:16 pm.]

El destino **-j MASQUERADE** se especifica para enmascarar la dirección IP privada de un nodo con la dirección IP externa del cortafuego o Puerta de enlace.

**Pre-enrutamiento:** si tiene un servidor en la red interna que desee cambiar a externo, use la cadena de destino de PRE-ENRUTAMIENTO en NAT –j DNAT para especificar una dirección de destino IP y puerto en el que los paquetes que solicitan una conexión a su servicio interno pueden ser reenviados.

Por ejemplo, si desea reenviar solicitudes de entrada HTTP a su servidor dedicado Apache HTTP en 172.31.0.23, use el siguiente comando:

# iptables –t nat –A PREROUTING –i eth0 –p tcp --dport 80 –j DNAT --to 172.31.0.23:80

Esta regla especifica que la tabla nat usa la cadena de PRE-ENRUTAMIENTO incorporada para reenviar las solicitudes entrantes de HTTP exclusivamente a la dirección IP de 172.31.0.23. [58]

### 2.7.7. DMZs e IPtables

Puede crear reglas **IPtables** para enrutar el tráfico a algunas máquinas, tales como un servidor HTTP o FTP, en una *zona desmilitarizada* (DMZ). Un DMZ es una subred local especial dedicada a proporcionar servicios en un transportador público, tal como la Internet.

---

[58]Post-enrutamiento y enmascarado de IP. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 27 de junio del 2016, 9:20 pm.]

Por ejemplo, para establecer una regla para solicitudes HTTP de enrutamiento entrantes a un servidor dedicado HTTP en 10.0.4.2 (fuera del rango de LAN 192.168.1.0/24), NAT emplea la tabla **PREROUTING** para reenviar los paquetes al destino apropiado:

# iptables –t nat –A PREROUTING –i eth0 –p tcp --dport 80 –j DNAT --to-destination 10.0.4.2:80

Con este comando, todas las conexiones HTTP al puerto 80 desde fuera de LAN se enrutan al servidor HTTP en una red independiente del resto de la red interna. Esta forma de segmentación puede ser más segura que las conexiones que permiten conexiones HTTP a una máquina en la red.

Si el servidor HTTP está configurado para que acepte conexiones seguras, entonces el puerto 443 también se debe reenviar. [59]

### 2.7.8. Software malintencionado y direcciones IP falsas

Se pueden crear reglas más elaboradas para controlar el acceso a subredes específicas o incluso nodos específico, dentro de una LAN. También puede restringir el contacto a su servidor de algunas aplicaciones dudosas o programas tales como troyanos, worms u otros virus de cliente y servidor.

Por ejemplo, algunos troyanos escanean redes de servicios en puertos de 31337 a 31340 (llamados los puertos *elite* en terminología de ciberpiratas).

---

[59] DMZ e IPtables. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 27 de junio del 2016, 8:00 am.]

Puesto que no hay servicios ilegítimos que se comuniquen a través de estos puertos no estándar, el bloquearlos puede disminuir efectivamente las posibilidades de que nodos infectados en su red se comuniquen de forma independiente con sus servidores maestros remotos.

Las siguientes reglas descargan todo el tráfico TCP para usar puerto 31337:

# iptables –A OUTPUT –o eth0 –p tcp --dport 31337 --sport 31337 –j DROP
# iptables –A FORWARD –o eth0 –p tcp --dport 31337 --sport 31337 –j DROP

También puede bloquear conexiones externas que intenten suplantar rangos de direcciones IP privadas para infiltrar su LAN.

Por ejemplo, si su LAN usa el rango 192.168.1.0/24, usted puede diseñar una regla que instruya al dispositivo de red de Internet (por ejemplo, eth0) para que descargue los paquetes a ese dispositivo con una dirección en su rango IP de LAN.

Puesto que, como política predeterminada, se recomienda rechazar los paquetes reenviados, cualquier otra dirección IP engañosa para el dispositivo externo (eth0) es rechazada automáticamente. [60]

# iptables –A FORWARD –s 192.168.1.0/24 –i eth0 –j DROP

---

[60]Software malintencionado y direcciones IP falsas. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 27 de junio del 2016, 7:50 am.]

### 2.7.9. IPtables y trazado de conexiones

Para inspeccionar y restringir conexiones a servicios basados en su *estado de conexión.* El módulo dentro de **iptables** usa el método llamado *rastreo de conexión* para almacenar información sobre conexiones entrantes. Puede aceptar o negar acceso con base en los siguientes estados de conexión:

- ✓ **NEW:** Un paquete que solicita una nueva conexión, tal como una solicitud HTTP.
- ✓ **ESTABLISHED:** Un paquete que hace parte de una conexión existente.
- ✓ **RELATED:** Un paquete que solicita una nueva conexión pero que hace parte de una conexión existente. Por ejemplo, FTP usa el puerto 21 para establecer una conexión, pero los datos se transfieren en un puerto diferente (por lo general, el puerto 20).
- ✓ **INVALID:** Un paquete que no hace parte de ninguna conexión en la tabla de seguimiento de conexión.

Puede utilizar la funcionalidad de estado de la conexión de **iptables** que rastrea con cualquier protocolo de redes, incluso si el protocolo mismo no tiene estado (tal como UDP). El siguiente ejemplo muestra la regla que usa el seguimiento de conexión para reenviar únicamente los paquetes asociados a una conexión establecida: [61]

# iptables –A FORWARD –m state --state ESTABLISHED, RELATED –j ACCEPT

---

[61]IPtables y trazado de conexiones. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 29 de junio del 2016, 8:45 am.]

## A. IPtables

En Red Hat Enterprise Linux vienen incluidas las herramientas avanzadas para filtraje de paquetes de redes, el proceso de control de paquetes de red en el ingreso, desplazamiento, control y salida de la pila de redes dentro del kernel. Las versiones de kernel anteriores a 2.4 dependían de ipchains para filtraje de paquetes y utilizaban listas de reglas aplicadas a paquetes en cada paso del proceso del filtraje. El kernel 2.4 introdujo iptables (conocido también como netfilter), el cual es similar a ipchains pero extiende ampliamente el alcance y control disponibles para filtrar paquetes de redes. [62]

## B. Filtraje de paquetes

El kernel de Linux emplea la herramienta Netfilter para filtrar paquetes, lo que permite a algunos de ellos ser recibidos o pasados mediante el sistema mientras detiene a otros. Esta herramienta se incorpora en el kernel de Linux y tiene tres tablas o listas de reglas incorporadas, así:

- ✓ **Filter**: la tabla predeterminada para manejar paquetes de redes.
- ✓ **Nat**: sirve para alterar paquetes que crean una nueva conexión y es utilizado por Traducción de dirección de red (NAT).
- ✓ **Mangle**: Sirve para tipos específicos de alteración de paquetes.

Cada tabla tiene un grupo de cadenas incorporadas que corresponde a las acciones realizadas en el paquete por netfilter.

Las cadenas incorporadas para la tabla de filtraje son las siguientes:

---

[62] IPtables.https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 04 de julio del 2016, 12:50 pm.]

- ✓ **IMPUT**: se aplica a los paquetes de redes destinados al host.
- ✓ **OUTPUT**: se aplica a los paquetes de red generados localmente.
- ✓ **FORWARD**: se aplica a los paquetes de redes enrutados a través del host.

Las cadenas incorporadas para la tabla NAT son las siguientes:

- ✓ **PREROUTING**: Altera los paquetes de redes a la llegada.
- ✓ **OUTPUT**: Altera los paquetes de redes generados localmente antes de ser enviados.
- ✓ **POSTROUTING**: Altera los paquetes de redes antes de ser enviados.

Las cadenas incorporadas par la tabla de MANGLE son las siguientes:

- ✓ **INPUT**: Altera los paquetes de redes destinados al host.
- ✓ **OUTPUT**: Altera los paquetes de redes generados localmente antes de ser enviados.
- ✓ **FORWARD**: Altera los paquetes de redes enrutados a través del host.
- ✓ **PREROUTING**: Altera los paquetes entrantes de red antes de ser enrutados.
- ✓ **POSTROUTING**: Altera los paquetes de redes antes de ser enviados.

Cada paquete de redes recibido por o enviado desde el sistema de Linux está sujeto al menos a una tabla. Sin embargo, el paquete puede estar sujeto a varias reglas dentro de cada tabla antes de emerger al final de la cadena. La estructura y propósito de dichas reglas pueden variar, pero suelen tratar de identificar el paquete que ingresa o sale de una dirección IP determinada o de un grupo de direcciones, cuando usan un protocolo determinado y un servicio de redes. La siguiente imagen resume la forma como el subsistema de iptables examina el flujo de paquetes:

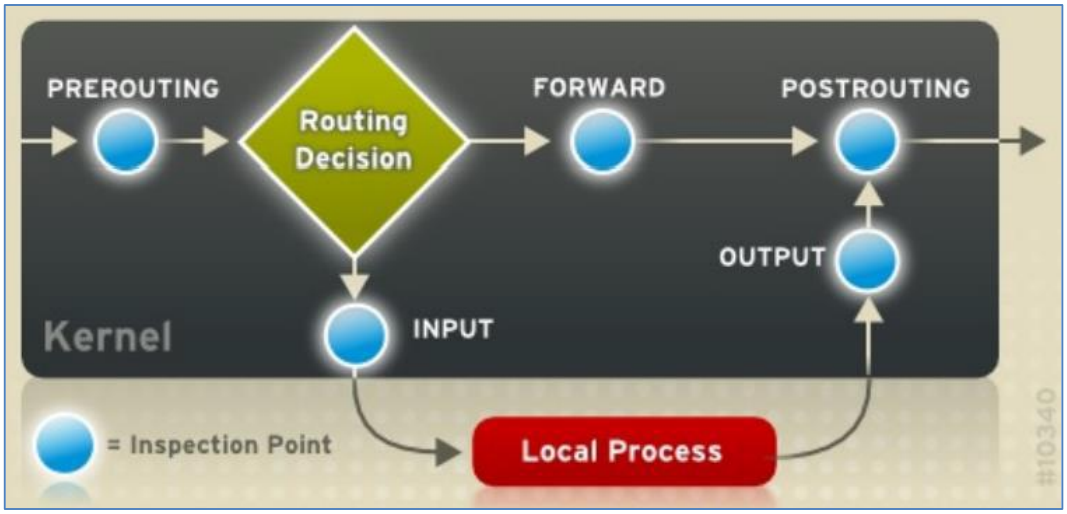



Figura 24.0 Subsistema de iptables examina el flujo de paquetes.

Independientemente de su destino, cuando los paquetes coinciden con una regla determinada en una de las tablas, se les aplica un destino o acción. Si la regla especifica un destino ACCEPT para el paquete coincidente, el paquete omite la parte restante de las revisiones de la regla y puede continuar su destino. Si la regla especifica un destino DROP, ese paquete no podrá acceder al sistema y no se enviará nada al host que envió al paquete. Si la regla especifica un destino QUEUE, el paquete se pasa a espacio de usuario. Si la regla especifica el destino opcional de REJECT, el paquete es descargado, pero se envía al originador del paquete. Cada cadena tiene una política predeterminada para ACCEPT, DROP, REJECT, o QUEUE. Si ninguna de estas reglas en la cadena se aplica al paquete, entonces el paquete es tratado de acuerdo con la política predeterminada.

El comando iptables configura estas tablas y establece las nuevas tablas si es necesario. [63]

---

[63]Filtraje de paquetest. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf [Citado el: 04 de julio del 2016, 5:30 pm.]

## 2.8 Bases Teóricas de Seguridad

### 2.8.1. Principios básicos de seguridad

Cuando se hace referencia a la palabra *Seguridad* tiene definiciones como certeza, firmeza, confianza, sin riesgo, dícese de las cosas ciertas, firmes y libres de peligro o riesgo, estado de las cosas bajo protección, confianza, tranquilidad de una persona, procedente de la idea de que no hay ningún peligro que temer.

Cuando se habla a cerca de seguridad en cómputo se hace referencia a todas las medidas para prevenir pérdidas de cualquier clase, significa que se contemplan tres aspectos básicos de cualquier sistema relacionado con el cómputo, confidencialidad, integridad y disponibilidad (figura 25.0).

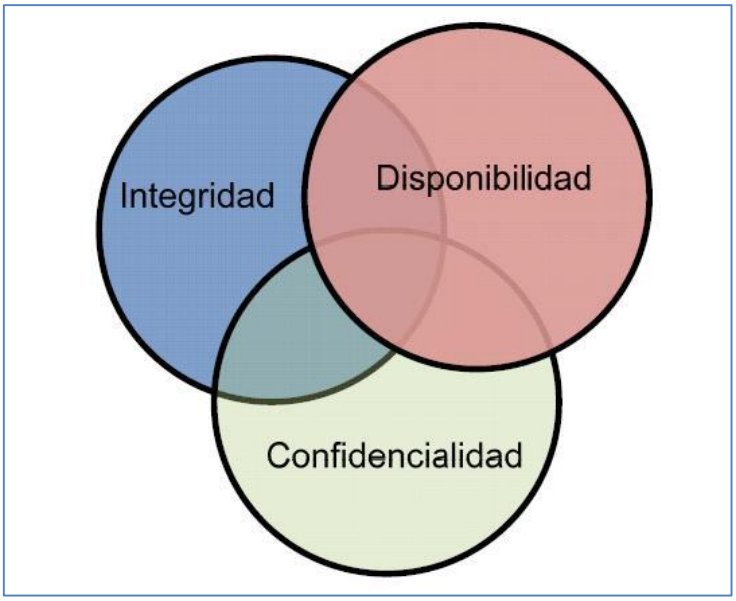



Figura 25.0 Principios de Seguridad.

La seguridad en cómputo intenta asegurar los tres principios, sobre tres activos principales, software, hardware e información, además de la comunicación entre ellos y las debilidades humanas que interactúen con

éstos, esto constituye la base de la seguridad en cómputo. Realizar un análisis de la situación actual brinda el panorama para la construcción de un sistema seguro, ayuda a encontrar el balance correcto de los tres factores (integridad, disponibilidad y confidencialidad) para una implementación óptima de seguridad, al definir cuál es el pilar primordial que se desea asegurar sin dejar de pensar en los otros dos.

Como un profesional en la seguridad es importante brindar un balance entre agregar barreras de seguridad para prevenir ataques y permitir que el sistema siga siendo funcional a los usuarios.

La seguridad, funcionalidad y facilidad de uso, deben ser contempladas para generar este balance. En general, cuando se incrementa la seguridad, la funcionalidad del sistema y facilidad de uso disminuyen.

**Integridad:** Significa que los activos pueden ser modificados sólo por partes autorizadas o caminos autorizados, en este contexto se incluye escribir, cambiar, modificar estado, borrar y crear.

El término integridad, tiene diferentes significados en distintos contextos, por ejemplo, si se refiere a conservar la integridad de algún elemento, puede significar que el mismo es:

- ✓ Idéntico.
- ✓ Certero, exacto.
- ✓ Sin modificaciones, alteraciones o agregaciones.
- ✓ Modificado sólo en formas aceptables.
- ✓ Modificado sólo por personas autorizadas.
- ✓ Modificado sólo por procesos autorizados.
- ✓ Consistente.
- ✓ Con sentido.

Integridad puede también significar dos o más de estas propiedades, la integridad está protegida en la mayoría de las ocasiones con controles de acceso que determinan que personas, procesos o entidades tienen acceso a los recursos para lograr escribir, cambiar, modificar, borrar y crear. En los sistemas de cómputo la integridad de los archivos es garantizada empleando algoritmos hash y firma digital.

**Disponibilidad:** Se refiere a que los activos están accesibles para las partes autorizadas en tiempos apropiados, en otras palabras, si una persona o sistema tuvo acceso legítimo a un conjunto de objetos particulares, este acceso no deberá impedirse.

Disponibilidad aplica tanto a datos como a servicios, se dice que un objeto o servicio está disponible si:

- ✓ Éste está presente en una forma útil.
- ✓ Si tiene capacidad suficiente para prestar el servicio.
- ✓ El servicio se completa dentro de un periodo de tiempo aceptable.
- ✓ El servicio involucra una filosofía de tolerancia a fallas.
- ✓ Tiene concurrencia, que es un acceso simultáneo, administrando tiempos muertos.

En sistemas de cómputo no existe una medida a tomar que garantice la disponibilidad de algún activo al 100%, por las implicaciones en gasto que puede tener esta propiedad de la seguridad, así como las nuevas debilidades que se detectan día a día en los diferentes sistemas en general. En tiempos pasados la seguridad en cómputo fue exitosa enfocándose a la confidencialidad y la integridad, la implementación de disponibilidad es uno de los siguientes grandes cambios.

**Confidencialidad:** Asegura que los activos sean accedidos sólo por partes autorizadas, es importante definir qué se entiende por “acceso”, acceso no sólo se refiere a leer, sino también a ver, imprimir o simplemente conocer la existencia de un activo particular. Confidencialidad en algunas ocasiones es también llamada privacidad o secreto.

La confidencialidad es uno de los principios de seguridad que más importancia se le ha dado desde la antigüedad, en el campo militar, diplomático y político se empleaban distintas formas para transportar la información y en caso de caer en manos de un tercero éste no pudiera interpretarla. [64]

Algunas de las maneras que se utilizan para garantizar la confidencialidad son:

- ✓ ***Valija Diplomática*** (Mecanismo empleado para enviar información de tal manera que sólo la persona involucrada pueda tener acceso a la información).
- ✓ ***Cifrado simétrico*** (Emplean una misma contraseña para ocultar y para recuperar la información).
- ✓ ***Cifrado asimétrico*** (Emplea dos contraseñas diferentes, una para ocultar la información y otra para recuperarla).
- ✓ ***Mecanismos para ocultar la información de los propietarios.***

[64]Principios básicos de seguridad http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 14 de julio del 2016, 11:26 am.]

### 2.8.2. Vulnerabilidad, amenazas, riesgo y control

Cuando se prueba un sistema de cómputo, una de las tareas principales es pensar cómo el sistema podrá tener un mal funcionamiento, de esta manera se busca mejorar el diseño del mismo.

Un sistema de cómputo consta de tres componentes que se valoran por separado; hardware, software y datos, cada uno de estos activos tiene un valor propio el cual afecta al sistema de diferente manera, esta estimación es necesario realizarla ya que lleva a determinar la prioridad de atención de los activos. En este proceso intervienen cinco factores: vulnerabilidades, amenazas, riesgo, ataques y control.

Las **vulnerabilidades** son debilidades en el sistema de seguridad, por ejemplo, un procedimiento, diseño, implementación, que pueden ser explotados causando pérdidas o daños, un ejemplo claro es cuando un sistema particular puede tener acceso a datos sin autorización, debido a que éste no verifica la identidad del usuario antes de permitirle el acceso a los datos.

Una **amenaza** a un sistema de cómputo es un conjunto de circunstancias que pueden ser potencialmente causa de pérdidas o daños, para ver la diferencia entre vulnerabilidad y amenaza, se puede considerar el siguiente ejemplo; un contenedor de agua está limitado por una pared, dicha pared presenta una fisura, el agua al subir de nivel comenzará a ejercer presión en la fisura y causará un daño a la persona que se encuentra fuera al momento que se produzca un desborde de agua, la fisura del muro es una vulnerabilidad que es aprovechada por la fuerza del agua y puede ocasionar lesiones a la persona (figura 26.0).

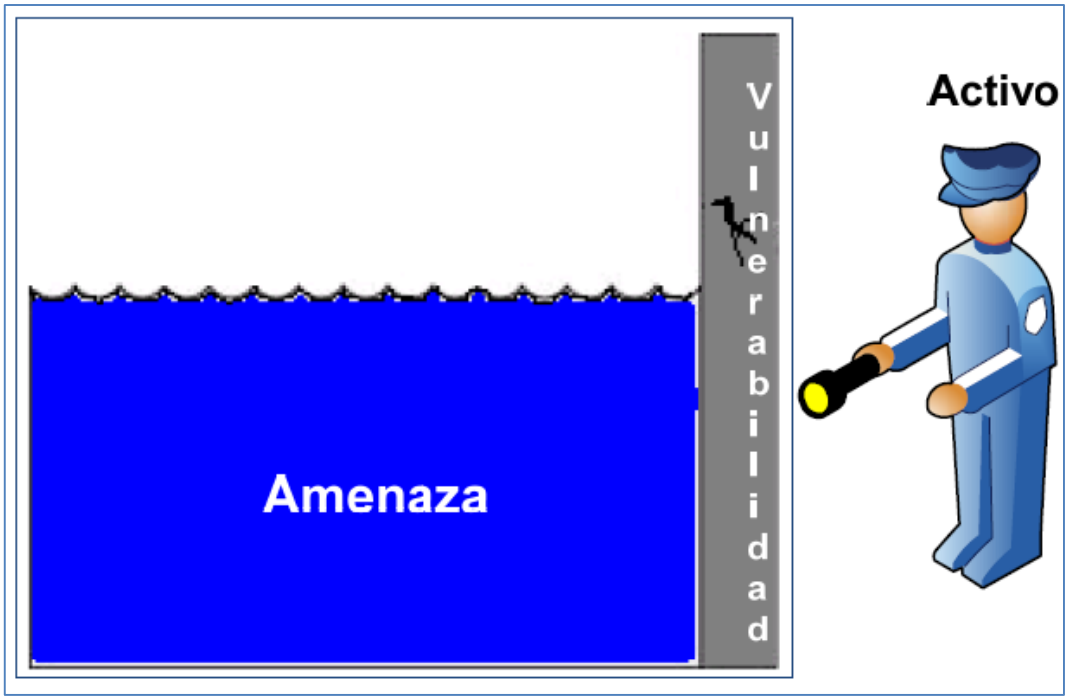



Figura 26.0 Diagrama amenaza, vulnerabilidad y activo.

Un ***riesgo*** está definido como la probabilidad de que una amenaza explote una vulnerabilidad, además si una amenaza explota una vulnerabilidad se lleva a cabo un ataque, en el caso anterior el riesgo está definido con base en la velocidad de aumento en el nivel de agua.

El ***control o mecanismo*** se define como una medida de protección empleada, un control es un dispositivo, acción, procedimiento o técnica que elimina o reduce una vulnerabilidad, en el caso anterior se podría sellar la fisura con la finalidad de disminuir el riesgo.

Las amenazas presentan cuatro tipos básicos de operación (intercepción, interrupción, modificación y fabricación) enfocados a los tres activos de sistemas de cómputo (figura 27).

La ***interrupción*** se refiere a impedir la comunicación entre dos entidades, esto atenta directamente a la disponibilidad.

La ***intercepción*** permite la comunicación entre dos entidades, pero los datos que son trasmitidos pueden ser vistos por un tercero, atenta contra la confidencialidad.

La ***modificación*** involucra a una tercera entidad entre los dos punto principales de una comunicación, permitiéndole modificar la información que se transmiten en ambas direcciones, atenta contra la integridad.

La ***fabricación*****,** es muy similar a la modificación, solo que en ese caso la información transmitida es completamente generada por una tercera entidad, atenta contra la integridad.

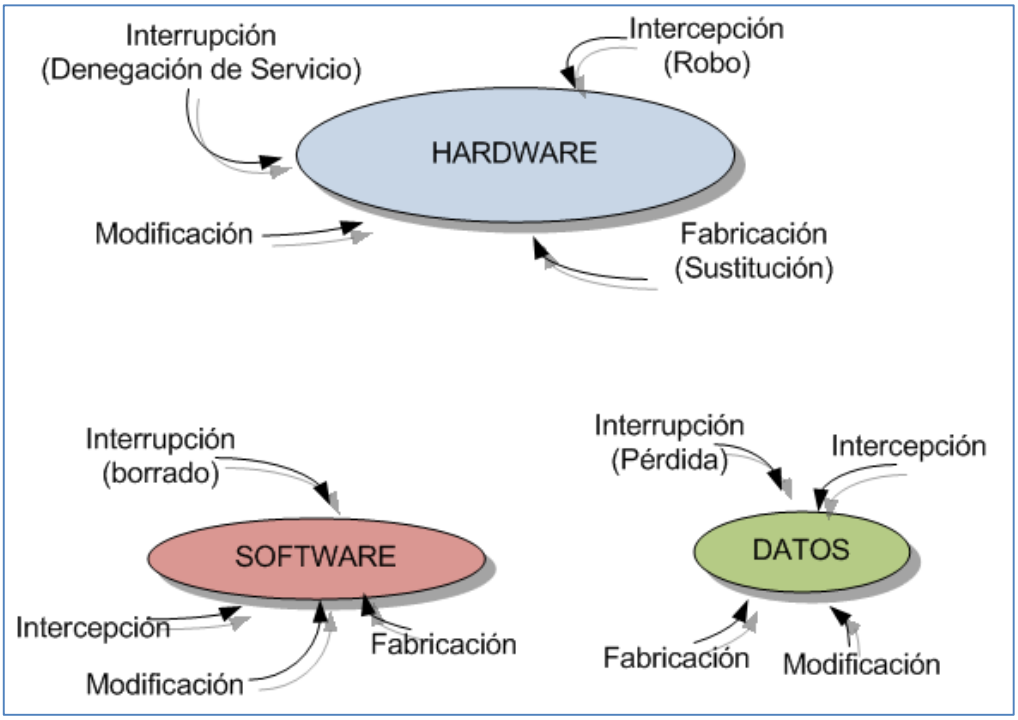



Figura 27.0 Vulnerabilidades de los sistemas de cómputo.

Las vulnerabilidades de hardware son mayores debido a que están compuestas de objetos físicos los cuales son más fáciles de atacar, al agregar un dispositivo, cambiando el mismo, eliminándolo, interceptando el tráfico de él, inundándolo de tráfico hasta que deje de funcionar, ataques físicos al mismo, mojarlo, corto circuito, quemarlo, congelarlo y robo principalmente, sin

embargo, su diseño e implementación puede colocarlo como dispositivo seguro.

Profesionales de la seguridad en cómputo repetidamente encuentran que la más grande amenaza de seguridad es de origen interno, debido a la cantidad de datos a los que tienen acceso para hacer su trabajo los empleados, además de conocer las vulnerabilidades de la institución de una manera más sencilla. [65]

El equipo de cómputo es de uso limitado si no se cuenta con software (Sistemas operativos, controladores, servicios y aplicaciones en general), cualquiera de este software puede ser reemplazado, cambiado, destruido y explotado maliciosamente o accidentalmente al provocar un comportamiento anómalo. Accidentalmente o no, las amenazas explotan las vulnerabilidades de software, En el caso de los datos, éstos pueden ser interpretados en ocasiones por el público en general, ataques a este activo tienen mayor impacto que ataques al software y hardware, ya que en ocasiones los datos se convierten en información que contiene secretos empresariales, cuentas bancarias, bases de datos inmensas con identificaciones personales, registros escolares, historias médicas y muchos más, los cuales si caen en manos equivocadas pueden provocar un daño impresionante o al ser utilizada para provocar otros ataques a partir de la obtención, modificación y divulgación de ésta. [66]

---

[65] Robert Richardson, CSI Computer Crime & Security Survey, 2008, pág. 16-17. [Citado el: 14 de julio del 2016, 9:20 am.]

[66] Vulnerabilidades, amenazas, riesgo y control. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 14 de julio del 2016, 10:30 am.]

### 2.8.3. Ataques

Los ataques son la culminación de una amenaza al explotar una vulnerabilidad, estos en general son acciones que atentan contra la disponibilidad, integridad y confidencialidad de algún activo, éstas son las razones primordiales por las cuales actúa la seguridad buscando limitar la acción de éstos o minimizando su alcance, pueden ser realizados de diferentes maneras como robo, ingeniería social, ataque remoto, acceso físico, ataques internos, penetración, entre otros, así como por diferentes tipos de entidades, humanas, lógicas y naturales.

En general los ataques son producidos por diversas entidades físicas y lógicas, dentro de las físicas se encuentran las personas que buscan realizar algún daño, a éstas se les conoce como atacantes o perpetradores en términos generales
La consecuencia de los ataques depende del tipo de impacto sobre el activo, algunas de las más comunes son:

- ✓ Pérdidas económicas.
- ✓ Pérdida de imagen pública.
- ✓ Responsabilidades legales.
- ✓ Daño o pérdida de la vida.
- ✓ Incumplimiento de acuerdos de servicio para el público o departamentos de gobierno.
- ✓ Violación de acuerdos de confidencialidad.
- ✓ Incapacidad para llevar a cabo tareas críticas.
- ✓ Modificación o pérdida de datos.

El impacto es una representación del daño o percepción de los daños, una vez que se ha culminado el ataque, en relación con la confidencialidad, integridad y disponibilidad.

Para llevar a cabo un ataque se requiere una planeación previa del mismo, es decir, el atacante debe contar con un esquema en el cual se detalle el objetivo y la metodología a emplear, actualmente las razones por las que un atacante desea concretar un plan de ataque son principalmente obtener ganancias económicas y fraudes.

Los ataques comprometen directamente la integridad, confidencialidad y disponibilidad de un sistema o red, en mayor volumen actualmente son virus, accesos no autorizados a recursos digitales, phishing, pharming, DoS, bots, abuso de redes wireless, ataques a DNS, sniffers.

Muchas de las herramientas utilizadas para ejecutar ataques son herramientas empleadas en el campo de la administración, adicional a esto, hay herramientas de uso dedicado para obtener un beneficio directo, algunas de estas herramientas son *ping, traceroute, whois, finger, rusers, nslookup, rcpinfo, telnet, dig.* [67]

**A. Fases de un ataque**

Para llevar a cabo un ataque se sigue una metodología, ésta consta de cinco pasos y está definida por *Certified Ethical Hacker* – Hacker Ético Certificado, reconocimiento, escaneo, obtención del acceso, manteniendo el acceso y encubrimiento de rastros, esta metodología se contempla sólo para ataques lógicos aunque puede ser adaptada para ataques físicos (figura 28.0).

[67]Ataques. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 19 de julio del 2016, 11:20 am.]

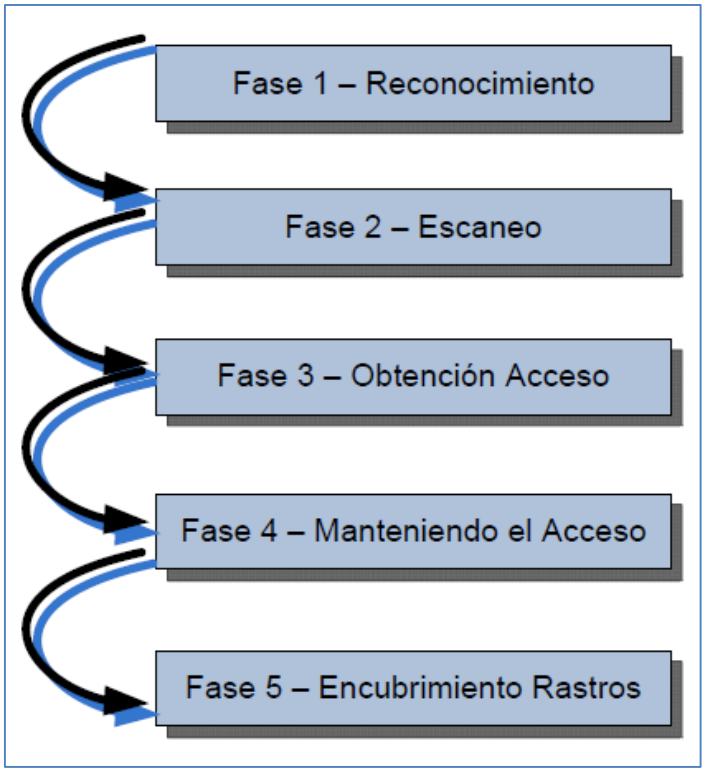



Figura 28.0 Fases de un ataque.

En la ***fase uno reconocimiento****,* se involucra la obtención de información que se encuentra de manera pública del objetivo, se emplean dos tipos de reconocimiento: pasivo y activo. En el caso de los reconocimientos pasivos, éstos consisten en obtener información sin alterar los principios básicos de la seguridad, como métodos de implementación se utilizan búsquedas en Internet, google y en el sitio de la organización donde muestre información sin autenticarse, como es directorio de ejecutivos, características del software empleado en su sitio, principalmente, también es llamado obtención de información, la ingeniería social y dumpster diving (husmear en la basura para localizar documentos importantes) se consideran métodos de obtención de información pasiva. Observar el tráfico de la red es otra forma de reconocimiento pasivo, ya que no se generan paquetes adicionales que puedan observarse, pero existen otros métodos de detección para estos casos, siendo éste un campo de información muy útil, ya que dentro de la

información que se brinda están los rangos de IP, convención de nombre, servidores o redes ocultas.

El reconocimiento activo involucra el descubrimiento de la red sobre hosts individuales, direcciones IP, servicios, versiones de los servicios, sistema operativo, observar el tráfico de red, servidores, redes ocultas o intranets, principalmente, este tipo de escaneo genera un riesgo mayor al crear paquetes en la red para obtener información, este tipo de reconocimiento puede darle al atacante indicadores de medidas de seguridad empleadas en el lugar como es el caso de los firewalls.

Dentro de la fase dos*,* ***escaneo****,* se toma la información descubierta durante la fase de reconocimiento y se utiliza ésta para examinar la red. Algunas de las herramientas que se emplean para esta fase pueden incluir escaneo de puertos, mapeo de red y escaneo de vulnerabilidades.

En la fase tres**,** ***obtención del acceso*****,** se lleva a cabo el ataque real, las vulnerabilidades descubiertas durante el reconocimiento y el escaneo, la información encontrada es explotada para obtener acceso, los métodos principales que se emplean en esta fase son el acceso local a una computadora, Internet, acceso fuera de línea, denegación de servicio, robo de sesión, entre otros.

***Manteniendo el acceso*** es la cuarta fase, una vez obteniendo el acceso se busca generar un mecanismo que permita ingresar al sistema para generar ataques futuros, en ocasiones los mismos atacantes protegen al equipo para evitar que otros puedan atacarlo una vez que se tiene acceso a éste, cuando el equipo es comprometido se le conoce como un sistema zombi.

La última fase, ***encubrimiento de rastros***, es empleada para evitar ser detectado por el personal de seguridad de la institución o del equipo, busca

eliminar toda evidencia que se genere al momento de realizar el ataque para anular acciones legales, algunos de los elementos que buscan eliminar o modificar en esta fase son las bitácoras, alarmas en IDS (Intrusion Detection System – sistema detector de intrusos), registros en firewalls, horario de ingreso a un sistema, cuentas de usuario y log de un sistema. [68]

[68]Fases de Ataque. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf [Citado el: 19 de julio del 2016, 5:50 pm.]

## 2.9. Definiciones de términos

1. **Enrutador**

   El enrutador es el aparato encargado de conectar a dos ordenadores entre si. En su interior contiene las instrucciones y protocolos adecuados como para permitir el envío y la recepción de los paquetes de información entre ambos, de modo que esta llegue a su destino y no se pierda, usando siempre la ruta más adecuada en cada momento. http://www.mastermagazine.info/termino/4892.php [Citado el: 20 de julio del 2016, 9:26 am.]

2. **handshake**

   Es un Técnica de control de entradas/salidas muy utilizada para el intercambio de mensajes entre ordenadores y sus periféricos. Implica un diálogo entre los elementos a intercomunicar. http://glosarios.servidor-alicante.com/electronica-informatica-telecomunicaciones_en/handshake [Citado el: 20 de julio del 2016, 4:51 pm.]

3. **Spoofing**

   Spoofing se conoce a la creación de tramas TCP/IP utilizando una dirección IP falseada; la idea de este ataque - al menos la idea - es muy sencilla: desde su equipo, un pirata simula la identidad de otra máquina de la red para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP del host suplantado. http://www.zonavirus.com/articulos/que-es-el-spoofing.asp [Citado el: 20 de julio del 2016, 2:55 pm.]

4. **Switch**

   La palabra Switch, se puede traducir como conmutador ó interruptor. Se trata de un dispositivo utilizado en redes de área local (LAN - Local Area Network), recordando que una red local es aquella que cuenta con una interconexión de computadoras relativamente cercanas por medio de cables. La función primordial del Switch, es interconectar diversos dispositivos de red, utilizando tecnología para evaluar las direcciones de destino y con ello encaminar los datos exclusivamente al dispositivo que lo debe de recibir.

   http://www.informaticamoderna.com/Switch.htm [Citado el: 21 de julio del 2016, 10:16 am.]

5. **DMZ**

   Una zona DMZ se conoce como una zona desmilitarizada, es decir, una zona segura que no está dentro de nuestra red local, pero que tampoco es externa a nuestra empresa. Por lo tanto, se plantea como un paso intermedio entre nuestra red y el acceso a Internet, que si protegemos por un Firewall debidamente dejaremos como una zona segura dentro de nuestra empresa. http://www.pymesyautonomos.com/tecnologia/zona-dmz-la-red-a-salvo-de-curiosos-en-la-empresa [Citado el: 21 de julio del 2016, 9:27 am.]

6. **Hub**

   Hub significa concentrador, se trata de un dispositivo utilizado en redes de área local (LAN - Local Area Network), una red local es aquella que cuenta con una interconexión de computadoras relativamente cercanas por medio de cables. La función primordial del Hub es concentrar las terminales (otras computadoras cliente) y repetir la señal que recibe de todos los puertos, así todas las computadoras y equipos escuchan los mismo y pueden definir que información les corresponde y enviar a

todas lo que se requiera; son la base de la creación de redes tipo estrella.

http://www.informaticamoderna.com/Hub.htm#defi [Citado el: 21 de julio del 2016, 11:21 am.]

7. **Proxy**

Un proxy web es utilizado para interceptar la navegación de páginas web por motivos de seguridad, anonimato, rendimiento, etc. Un proxy web se puede acceder por una dirección IP, gratuito o de pago, que es agregada a un navegador (también existen programas proxy para evitar el proceso de configuración). Cuando alguien utiliza el navegador, todo lo que se haga en el mismo pasa primero por el proxy (el servidor proxy).

http://www.alegsa.com.ar/Dic/proxy.php [Citado el: 21 de julio del 2016, 5:18 pm.]

8. **Host**

La palabra inglesa host, que en español se traduciría como huesped, se usa en informática sobre todo a nivel de redes, donde en muchas ocasiones (no siempre), se asimila al concepto de servidor.

http://www.mastermagazine.info/termino/5270.php [Citado el: 22 de julio del 2016, 3:36 pm.]

9. **LAN**

LAN son las siglas de Local Area Network, Red de área local. Una LAN es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios).

http://www.masadelante.com/faqs/lan [Citado el: 22 de julio del 2016, 8:06 am.]

10. **ISP**

(Internet Service Provider - Proveedor de servicios de Internet). Empresa que se encarga de conectar y dar servicio de Internet a sus usuarios por algún medio (cable, inalámbrico, satelital, celular, telefónico, etc.).

http://www.alegsa.com.ar/Dic/isp.php [Citado el: 22 de julio del 2016, 11:14 am.]

11. **WAN**

(Wide Area Network - Red de Área Extensa). WAN es una red de computadoras de gran tamaño, generalmente dispersa en un área metropolitana, a lo largo de un país o incluso a nivel planetario.

http://www.alegsa.com.ar/Dic/wan.php [Citado el: 22 de julio del 2016, 4:18 pm.]

12. **Gateway**

En telecomunicaciones, el término gateway (pasarela o puerta de enlace) es un término aplicable en diferentes situaciones y a diferentes dispositivos, programas e incluso computadoras, siempre que actúen como un nodo en una red, en donde su función sea conectar dos redes diferentes.

http://www.alegsa.com.ar/Dic/gateway%20telecomunicaciones.php [Citado el: 22 de julio del 2016, 4:27 am.]

13. **Puerto de red**

Un puerto de red es una interfaz para comunicarse con un programa a través de una red. Un puerto suele estar numerado. La implementación del protocolo en el destino utilizará ese número para decidir a qué programa entregará los datos recibidos.

http://ayudas-it.blogspot.pe/2009/09/puerto-de-red-concepto.html [Citado el: 22 de julio del 2016, 8:00 pm.]

**14. Socket de internet**

Un socket es también una dirección de Internet, combinando una dirección IP (la dirección numérica única de cuatro partes que identifica a un ordenador particular en Internet) y un número de puerto (el número que identifica una aplicación de Internet particular, como FTP, Gopher, o WWW).

http://www.masadelante.com/faqs/socket [Citado el: 22 de julio del 2016, 3:11 pm.]

**15. Router**

Un “Router” es como su propio nombre indica, y fácilmente se puede traducir, un enrutador o encaminador que nos sirve para interconectar redes de ordenadores y que actualmente implementan puertas de acceso a internet como son los router para ADSL, los de Cable o 3G.

https://applendiendo.wordpress.com/2007/10/23/que-es-un-router-y-para-que-sirve/ [Citado el: 23 de julio del 2016, 10:10 am.]

**16. Squid**

Squid puede funcionar como Servidor Intermediario (Proxy) y caché de contenido de Red para los protocolos HTTP, FTP, GOPHER y WAIS, Proxy de SSL, caché transparente, WWCP, aceleración HTTP, caché de consultas DNS y otras muchas más funciones como filtración de contenido y control de acceso por IP y por usuario.

http://informate.suministros.com.ni/informatica/que-es-squid.html [Citado el: 23 de julio del 2016, 03:29 pm.]

**17. Ingenieria social**

La ingenieria social es el término es utilizado para describir un método de ataque, donde alguien hace uso de la persuasión, muchas veces abusando de la ingenuidad o confianza de un usuario, para obtener

informacion que pueda ser utilizada para tener acceso autorizado a la información de las computadoras.
http://www.informatica-hoy.com.ar/software-seguridad-virus-antivirus/Ingenieria-Social-Seguridad-Informatica.php [Citado el: 23 de julio del 2016, 6:14 pm.]

# CAPÍTULO 3

# 3

# MARCO METODOLÓGICO

## 3.1. Nivel y tipo de investigación

### 3.1.1. Nivel de investigación

Proyectiva.

### 3.1.2. Tipo de investigación

Investigación – acción.

### 3.1.3. Diseño de investigación

Transversales descriptivos.

## 3.2. Población y muestra

### 3.2.1. Determinación del Universo / Población

La empresa Conexión Linux SAC consta de 5 empleados, lo cual el universo llega hacer igual a su población y por ende también igual a su muestra.

Universo = Conexión Linux SAC.
Población = Conexión Linux SAC.

### 3.2.2. Selección de muestra

Muestra = Conexión Linux SAC.

# CAPÍTULO 4

# 4

# MARCO REFERENCIAL

## 4.1. La empresa Conexión Linux S.A.C.

Conexión Linux, es una empresa principalmente orientada al sector telecomunicaciones y tecnologías de Información con operaciones simultáneas en Perú, EEUU y México.

Su experiencia especializada se centra en Ingeniería digital TCP orientada en campos de transmisión de datos, voz, proyecto de telecomunicaciones y soluciones de negocios en voz (Call Center). La familia ConexionLinux cuenta con más de 50 clientes distribuidos en todo el Perú con el servicio de Internet Satelital, que garantizan la correcta operación de todas nuestras unidades de negocios.

Nuestra experiencia en el mercado corporativo inicia del año 2002.

### 4.1.1. Misión

“Nuestra misión es de prepararnos para poder ofrecer a nuestros clientes una capacidad de ayudarlos a obtener el máximo beneficio de sus activos”.

### 4.1.2. Visión

“Nuestra visión está orientada a la especialización en Telecomunicaciones y Tecnologías de Información como soporte de las unidades de negocio a fin de ofrecer un valor agregado en un mercado dinámico y exigente. En dicho contexto y en esta era de las Telecomunicaciones digitales”.

### 4.1.3. Contacto

Número telefónico: 5620226.
Correo electrónico: webmaster@conexionlinux.com.

Ubicación: Cal. José Marti Nro. 254 San Miguel – Lima.
RUC: 20536792008
Página web: www.conexionlinux.com

### 4.1.4. Equipo de trabajo

Gerente: Mg. Alex Segura Nuñez.
Administrador: Miguel Chavez Ordoñez.
Redes y telecomunicaciones: Jhim Muñoz Zevallos y Joseph Veliz Castañeda.
Desarrollo de software: Jorge Mingos Tarazona.
Técnico: Claudio Quispe Alejandro.

### 4.1.5. Laboratorios

Cuenta con un laboratorio de informática con 4 terminales para realizar pruebas de software y hardware, se realizan testeos de red, se implementan servidores virtuales y físicos para probar velocidad, vulnerabilidad y ataques.

### 4.1.6. Características de las computadoras del laboratorio

Las características de las computadoras son la siguiente:

a) **PC1:** Procesador Core i5-650 3.2.ghz, 4 Gb de memoria RAM DDR3 (Kingston), 80 Gb de disco duro rígido (Samsung), Placa Madre Ecs H61h2-mv, 2 tarjetas de red DLink, Lectora de DVD Samsung.

b) **PC2:** Procesador Core i5-650 3.2.ghz, 4 Gb de memoria RAM DDR3 (Kingston), 80 Gb de disco duro rígido (Samsung), Placa Madre Ecs H61h2-mv, 2 tarjetas de red DLink, Lectora de DVD Samsung.

c) **PC3:** HP 8200 Elite, Procesador Intel Core i5 2400, 16 Gb de memoria RAM DDR3 (Samsung), 1 Tb de disco duro rígido (Samsung), Placa Madre Intel 8200, Lectora DVD +/- RW.

d) **PC4:** HP 8200 Elite, Procesador Intel Core i5 2400, 16 Gb de memoria RAM DDR3 (Samsung), 1 Tb de disco duro rígido (Samsung), Placa Madre Intel 8200, Lectora DVD +/- RW.

## 4.2. Análisis situacional de la red

La empresa Conexión Linux SAC, se dedica a brindar el servicio de internet satelital a todo el País, tiene la siguiente topología de red (Figura 1.0, Figura 2.0). Donde los servidores tienen el Sistema Operativo GNU/Linux Centos 7.0.

En el servidor master tenemos los siguientes servicios instalados DHCP, DNS y PROXY. Tiene un firewall que solo permite dar acceso de entrada y salida a los socket de las interfaces de red. Como ha de esperarse es fácil vulnerar el servidor master, ya que es él quien da la cara hacia la WAN y LAN, por lo cual ha sido vulnerado en dos ocasiones.

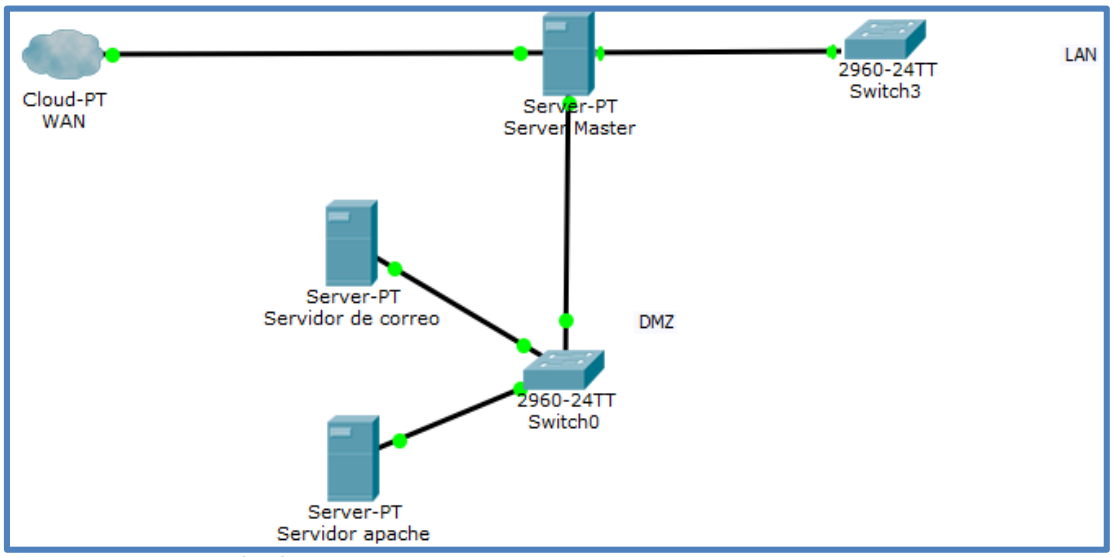


Figura 1.0 Topología física de la red.

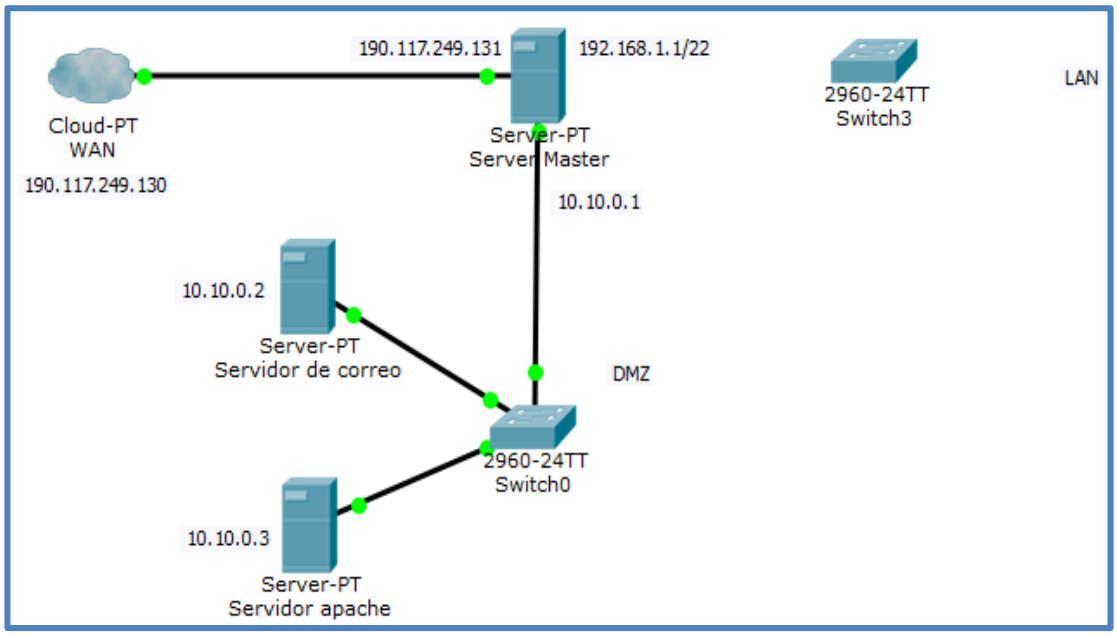


Figura 2.0 Topología lógica de la red.

A continuación se describirá cada uno de los componentes de la red:

### 4.2.1. ISP

El proveedor del servicio de internet es de la empresa Claro, con una ip pública fija (190.117.249.130), y un dominio conexionlinux.com, tiene fibra óptica con 1 Mbyte de velocidad al 10 %, ya que configuran servidores desde Lima a todo el País.

### 4.2.2. Servidor Master

El servidor master de la empresa Conexión Linux SAC. Ya tiene instalado los repositorios DHCP, DNS, VIM y SQUID, el firewall como tal no lo tienen configurado.

El servidor master tiene dos salidas (eth1, eth2), uno al DMZ, y el otro a la LAN.

Sobre la lentitud se observó que el servidor master no bloquea las peticiones del broadcast, peticiones que en un momento determinado queda colgado la red, como también está habilitado el protocolo IPV6, lo cual si se habilita en un host de usuario quedará fuera de la red.

### 4.2.3. DMZ

Cuenta con dos servidores en los DMZ, un servidor de correo (Zimbra), y un servidor apache con MySql. La configuración del servidor master al DMZ es usado con un FORWARD, lo que significa que solo hicieron un puente a las interfaces de red.

### 4.2.4. Host usuario

La empresa cuenta con 5 usuarios, cada uno de ellos tiene una determinada función que son los siguientes:

- ✓ **Gerencia:** tiene acceso a todas las páginas de internet, tiene acceso a los clientes, su tráfico de red queda colgado cada cierto tiempo, tiene acceso al servidor master desde su casa y desde la empresa misma, control total de todos los servicios.

- ✓ **Administración:** se restringe páginas de internet que no tienen que ver con el trabajo, tiene acceso a los clientes, su tráfico de red queda colgado cada cierto tiempo, tiene acceso a los servidores master desde su casa y desde la empresa misma.

- ✓ **Redes y telecomunicaciones:** se restringe páginas de internet que no tienen que ver con el trabajo, acceso total a los servidores de los clientes, su tráfico de red queda colgado cada cierto tiempo, tiene

acceso a todos los servidores en modo soporte desde la WAN y la LAN.

- ✓ **Desarrollo de software:** se restringe páginas de internet que no tiene que ver con el trabajo, tiene acceso a todos los servidores, su tráfico de red queda colgado cada cierto tiempo, ingreso al servidor desde la WAN y LAN.

- ✓ **Técnico:** se restringe páginas de internet que no tienen que ver con el trabajo, tiene acceso a los clientes, su tráfico de red queda colgado cada cierto tiempo, tiene acceso a los servidores desde la WAN y la LAN.

Como podemos observar el servidor master esta vulnerado, ya que se puede ingresar al login desde la LAN y la WAN, sin restricción alguna, esto permite que también los virus en la red puedan entrar y salir sin ningún problema; podemos notar que la red queda colgado cada cierto tiempo, típico caso del broadcast. Si bien es cierto que existe un proxy para restringir páginas web, esto no permite evitar los ataques al servidor, ya que está expuesta al público su eminente ataque.

### 4.2.5. Aplicaciones adicionales del servidor master

**a) Servicio DHCP:** En la empresa Conexión Linux SAC, el servidor master tiene configurado el DHCP en modo automático, en tal sentido si se conecta un Host automáticamente le indicará una IP.

**b) Servicio Proxy:** El servidor master de Conexión Linux SAC, tiene configurado un proxy con squid, quien restringe páginas de internet, cacheo de páginas web, restricción de velocidad de internet, y el acceso al internet.

c) **Servicio DNS:** El servidor master de Conexión Linux SAC, tiene configurado su propio DNS local, de tal manera tiene un nombre propio en la red, quien distribuye los paquetes de manera independiente.

# CAPÍTULO 5

# 5

# DISEÑO Y CONSTRUCCIÓN

## 5.1. Diseño

En este capítulo, se desarrollan las políticas de seguridad, los requerimientos a considerar en el diseño del firewall y las particularidades que se pueden encontrar en el script. Finalmente, se explica el diseño del script de iptables, detallando la estructura y mecanismos que se implementaron al servidor master.

### 5.1.1. Políticas y reglas de seguridad de Conexión Linux SAC

Las políticas de seguridad de la empresa se definen a los usuarios de la siguiente manera:

- ✓ **Gerencia:** tiene acceso a todas las páginas de internet, tiene acceso a los clientes, tiene acceso al servidor master desde su casa y desde la empresa misma, control total de todos los servicios.

- ✓ **Administración:** se restringe páginas de internet que no tienen que ver con el trabajo, tiene acceso al servidor samba desde la empresa misma.

- ✓ **Redes y telecomunicaciones:** se restringe páginas de internet que no tienen que ver con el trabajo, acceso total a los servidores de los clientes, tiene acceso a todos los servidores en modo soporte desde la WAN y la LAN.

- ✓ **Desarrollo de software:** se restringe páginas de internet que no tiene que ver con el trabajo, tiene acceso al servidor apache, ingreso al servidor desde la LAN.

- ✓ **Técnico:** se restringe páginas de internet que no tienen que ver con el trabajo, tiene acceso al servidor samba desde la LAN.

### 5.1.2. Requerimientos generales de la solución

Implementaremos un firewall en el servidor master, un firewall administrable, que administre tanto al DMZ y a la LAN. Usando iptables como herramienta principal en un script. Esto permitirá restringir el acceso al servidor master, es más se bloqueara el protocolo ICMP para que no puedan verlo desde la WAN, nuestra principal importancia radica en los ataques externos, sin dejar de lado los virus en la red, que hacen peticiones desde la LAN hacia la WAN. Por otro lado se bloqueara peticiones del broadcast innecesarios, esto nos ayudara a que el tráfico de red no quede colgado cada cierto tiempo.

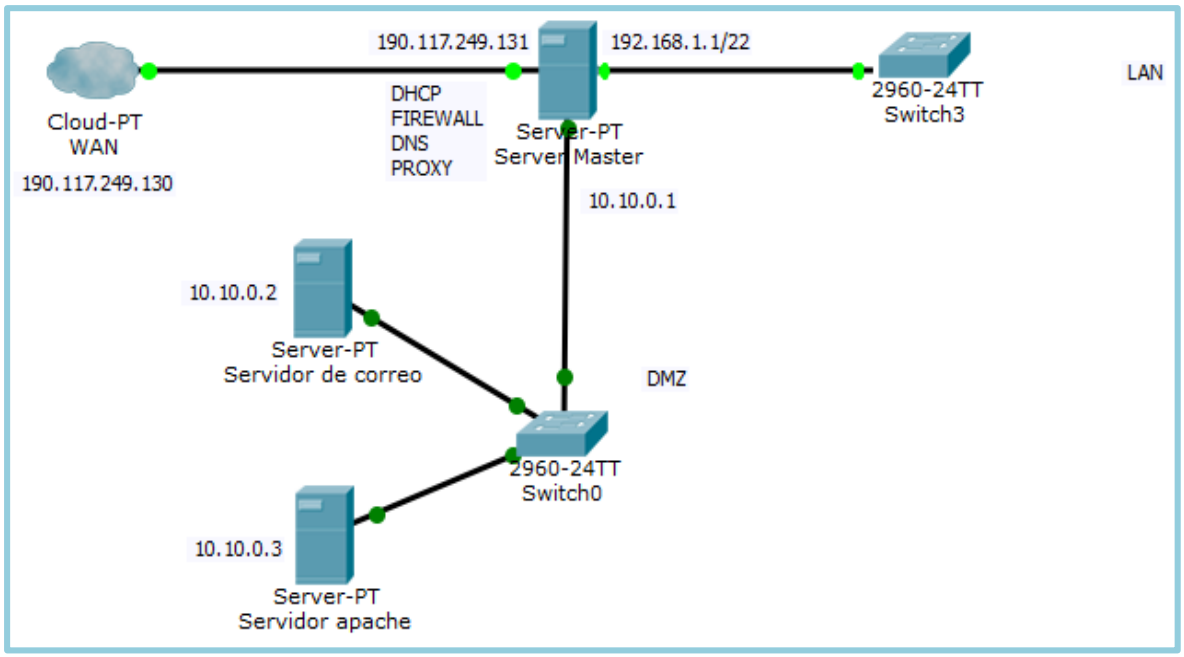



Figura 1.0 Topología lógica de la red.

**a) El firewall con iptables**

Son dos archivos .sh y .cfg, donde lo llamaremos AlliOne.sh y AlliOne.cfg, donde él .sh contiene todo el código y el .cfg será el que contiene las variables. A continuación describiremos brevemente estos dos archivos.

**b) AllInOne.sh**

Los archivos .sh son scripts que ejecutas directamente en la consola. La ventaja de instalar un programa con estos formatos es que por lo general van a funcionar. Como se sabe el SO, leerá línea por línea hasta llegar al último código, entonces si hay un error en una determinada línea, no llegara a terminar de ejecutar el bash.

**c) AllInOne.cfg**

Los archivos .cfg son archivos de configuración, se utilizan para configurar los ajustes iniciales para los programas de ordenador y son utilizados por una amplia gama de programas y aplicaciones. Algunos archivos cfg también se crean cuando se cambian los ajustes de un programa o aplicación informática.

**d) rc.local**

Si queremos que nuestro script arranque automáticamente al iniciar el servidor, tendremos que agregar el comando de ejecución en esta dirección (/etc/rc.local) es un lugar donde se ejecuta los comandos bash al iniciar el systemd.

**e) Permisos del archivo sh**

Tenemos que darle permiso de ejecución al archivo .sh para que pueda ser ejecutado como script, el archivo sh tendrá opciones de (start, restart, stop). El comando a usar para el permiso es: chmod a+x AlliOne.sh.

**f) Comandos a ejecutar**

Los comandos a ejecutar son ./AlliOne.sh start/restart/stop, como es un script se ejecuta con "./".

**g) Broadcast innecesario**

Se llama dominio de broadcast al grupo de hosts lógicamente conectados (en una subred) que pueden recibir mensajes de broadcast. Estos mensajes de broadcast son generados por un host y al llegar al Switch son enviados por todos los demás puertos del mismo.

Algunos protocolos utilizan mensajes de broadcast para iniciar la negociación y/o comunicación, como por ejemplo ARP (ARP Request) y DHCP (DHCP Request), los cuales representan la gran mayoría de los mensajes de broadcast que puedan circular por una red.

El principal problema que se presenta en la transmisión de paquetes de broadcast es el de las llamadas tormentas de broadcast, las cuales disminuyen el rendimiento de la red, llegando al punto de deshabilitarla completamente debido a la saturación de la misma, estas pueden ser producidas de dos formas: por uno o más hosts que envíen mensajes de broadcast a la red de manera continua, debido a algún fallo de configuración, o por uno o más paquetes de broadcast que, al llegar a búcles en la topología, son retransmitidos de manera infinita.

Por lo tanto en el servidor master se bloquea el broadcast innecesario, de esta manera podemos corregir errores de cuellos de botella en la red, errores que se convierten en caídas de red por segundos o hasta minutos, todo esto dependerá de cuantos usuarios tenemos en la subred.

### 5.1.3. Requerimientos del script

#### a) Interfaces del servidor

El servidor master cuenta con tres interfaces de red eth0, eth1 y eth2. Eth0 tiene la ip pública fija del ISP, eth1 es destinado para el DMZ, y eth2 es destinado para la LAN.

#### b) Gateway del ISP

Llamada también puerta de enlace predeterminada, se refiere al ip del router que brinda al servidor master, requisito para enlazar nuestro ip de eth0 al router.

#### c) Rango de ip's (mascara de subred)

Necesitaremos el rango de ip de las dos interfaces eth1 y eth2, esto se realizara indicando con la máscara de red. Por ejemplo 192.168.0.0/22, esto significa que la máscara es 255.255.252.0, es una red para 1024 ip's útiles de las cuales una es la de red 192.168.0.0 y la otra es el broadcast 192.168.3.255.

**d) Ip’s del servidor master**

Cada interfaz de red requiere un ip, en este caso eth0, eth1 y eth2 tienen que tener una ip determinada. Por ejemplo 192.168.0.1, 10.10.0.1.

**e) DNS**

Debemos tener los DNS de nuestro ISP, como también los DNS de google que son 8.8.8.8 y 8.8.4.4.

**f) Nat host**

Hacer un nat a un ip, es darle salida libre sin ningún tipo de restricción, la variable se usa de emergencia, cuando un host usuario necesita ingresar por ejemplo al SIAF y no sabemos los ip’s públicos del SIAF, entonces hasta encontrarlos podemos hacer un NAT a ese host para que tenga salida sin restricción. Como se menciona anteriormente solo se usa en caso de emergencia.

### 5.1.4. Particularidades del script

- ✓ Necesitas un usuario con privilegios de ejecución para que pueda reiniciar o parar el servicio.
- ✓ No cuenta con entorno gráfico, si bien es cierto podemos usar webmin, entre otros, no es del mismo entorno del script que se ejecuta.
- ✓ Nuevas versiones, cada cierto tiempo se muestra nuevas amenazas de ataque, por lo cual las versiones son cada vez mejoradas, que al final se convierte en un bucle, y que el encargado de redes y telecomunicaciones tendrá que hacer sus respectivas modificaciones si sea necesario.

### 5.1.5. Diseño del script

Existen dos archivos para el firewall, AlliOne.sh y AlliOne.cfg. Donde AlliOne.cfg tiene todas las variables con sus valores, y el AlliOne.sh tiene todo el código a ejecutar. A continuación se explicara el contenido y diseño de cada uno de estos dos archivos.

#### A. AllInOne.cfg

```
# Interfaces del servidor AllInOne
IFACE_WAN
IFACE_LAN_DMZ
IFACE_LAN_LOCAL

GATEWAY_WAN

NET_LAN_LOCAL
NET_LAN_DMZ


# IP del servidor AllInOne
IFACE_LAN_LOCAL_IP
IFACE_LAN_DMZ_IP
IFACE_WAN_IP

SERVER_EMAIL_DMZ_IP
SERVER_APACHE_DMZ_IP

SERVER_DNS1_IP
SERVER_DNS2_IP
SERVER_DNS3_IP
SERVER_DNS4_IP


NAT_HOST01_IP
NAT_HOST02_IP
```

Figura 2.0 Diseño del AllInOne.cfg.

#### a) Descripción de variables

- ✓ **IFACE_WAN, IFACE_LAN_DMZ, IFACE_LAN_LOCAL:** Son las interfaces que tiene el servidor y estos son: eth0, eth1 y eth2.

- ✓ **GATEWAY_WAN:** Es la puerta de enlace predeterminada del router que tiene la ip publica fija.

- ✓ **NET_LAN_LOCAL, NET_LAN_DMZ:** Indica el rango de ip's que tendrá eth1 y eth2 para la LAN.

- ✓ **IFACE_LAN_LOCAL_IP, IFACE_LAN_DMZ_IP, IFACE_WAN_IP:** Son las ip's que se asigna a cada interfaz de red.

- ✓ **SERVER_EMAIL_DMZ_IP, SERVER_APACHE_DMZ_IP:** Son las ip's de los servidores que están en los DMZ, con esto filtraremos a donde tiene salida.

- ✓ **SERVER_DNS_IP:** Es el ip de los DNS que nos brinda nuestro ISP, si no tenemos conocimiento podemos poner los DNS de google.

- ✓ **NAT_HOST_IP:** Es el ip del host usuario que no tendrá ninguna restricción por el firewall, esto lo usaremos en caso de emergencia.

## B. AllInOne.sh

Todas las variables descritas en AllInOne.cfg, se mencionan en este archivo. Se iguala a 0 para indicarle que no tiene valor, pero que la variable ha sido creada (Figura 2.0).

```
# ################ Variables #####################################
TC=/sbin/tc
IPTABLES=/sbin/iptables

# Interfaces del servidor AllInOne
IFACE_WAN=0
IFACE_LAN_DMZ=0
IFACE_LAN_LOCAL=0

GATEWAY_SATEL=0

NET_LAN_LOCAL=0
NET_LAN_DMZ=0

# IP del servidor AllInOne
IFACE_LAN_LOCAL_IP=0
IFACE_LAN_DMZ_IP=0
IFACE_WAN_IP=0

SERVER_EMAIL_DMZ_IP=0
SERVER_APACHE_DMZ_IP=0

SERVER_DNS1_IP=0
SERVER_DNS2_IP=0
SERVER_DNS3_IP=0
SERVER_DNS4_IP=0


NAT_HOST01_IP=0
NAT_HOST02_IP=0

# ###############################################################
```

Figura 3.0 Diseño de las variables del AllInOne.sh.

Se recarga las variables mencionadas anteriormente, hacemos un if para saber si el archivo AllInOne.cfg tiene las variables descritas en el sh, si eso se cumple, entonces se ejecuta el cfg y carga toda la información del archivo AllInOne.cfg (Figura 3.0).

```
# Reload Varibles from file
if [ -f /root/AllInOne.cfg ];then
        . /root/AllInOne.cfg
fi
```

Figura 4.0 Recarga de las variables del AllInOne.cfg.

TC también conocido como QoS Linux, sirve para hacer la modulación del tráfico en Linux. Esta no es la única manera de conseguir trabajo de control de tráfico, pero es la forma más sencilla que se puede utilizar (Figura 4.0).

IPTABLES, herramienta que usaremos para el firewall perimetral (Figura 4.0).

De esta manera llamamos a las dos herramientas a usar.

```
TC=/sbin/tc
IPTABLES=/sbin/iptables
```

Figura 5.0 Llamado a las herramientas a usar.

El desarrollo del código se dividirá en casos usando las funciones start, clean, stop y restart, donde todo el código estará en la función start (Figura 5.0).

**a) Función start:** La función start tiene la siguiente estructura:

- ✓ Plus de reglas predeterminadas, donde son indispensables al construir un firewall.
- ✓ Quitamos el broadcast innecesario.
- ✓ Reglas duras: son reglas donde no se pueden modificar ninguna línea del script, ya que están construidas a medida de la empresa. Estas reglas duras se diseña primero denegando todos los accesos a cualquier ip. Para que después se agregue uno por uno cada regla que se necesita y se requiera.

- ✓ Reglas blandas: son reglas que no necesariamente tiene que estar para que el firewall funcione, son reglas que se ponen en caso de abrir puerto o ip's de una determinada aplicación, por ejemplo el SIAF.

**b) Función clean:** La función clean tiene la siguiente estructura:

- ✓ Borramos todas las reglas actuales.
- ✓ Damos acceso libre al servidor, o mejor dicho aceptamos toda las entradas y salidas al servidor master.
- ✓ Y habilitamos el puente del forward en el servidor master.

**c) Función stop:** Esta función hace un llamado a la función clean.

**d) Función restart:** Esta función hace un llamado a la función stop, hace un sleep 1 y para terminar hace una función start.

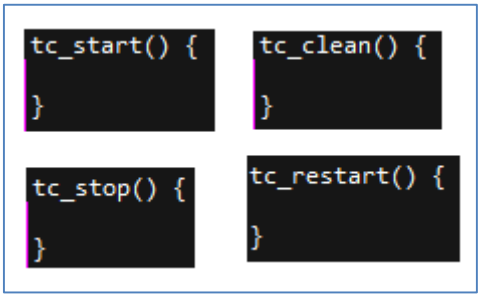



Figura 6.0 Las funciones del script.

### 5.1.6. Construcción del script

La construcción se realizara primero con la declaración de variables fijas para el script principal, estas variables estarán en el archivo AllInOne.cfg, como se menciona en el diseño del firewall.

Una vez construido el archivo AllInOne.cfg, se procederá a construir en script principal, denominado AllInOne.sh, se construirá como establece el diseño realizado en la sección anterior.

#### A. AllInOne.cfg

En este archivo es donde se asignan una serie de variables fijas, donde si por algún motivo se decide cambiar el ip a los DMZ o a las interfaces de red del servidor master, se podrá hacer los cambios en este archivo. Esto nos ayuda a que no tenemos que modificar nada del script principal (Figura 6.0).

```
# Variables
TC=/sbin/tc
IPTABLES=/sbin/iptables

# Interfaces del servidor AllInOne
IFACE_WAN=eth0
IFACE_LAN_DMZ=eth1
IFACE_LAN_LOCAL=eth2

GATEWAY_WAN=186.64.125.5

NET_LAN_LOCAL=192.168.0.0/22
NET_LAN_DMZ=10.10.0.0/22


# IP del servidor AllInOne
IFACE_LAN_LOCAL_IP=192.168.0.1
IFACE_LAN_DMZ_IP=10.10.0.1
IFACE_WAN_IP=186.64.125.6

SERVER_EMAIL_DMZ_IP=10.10.0.2
SERVER_APACHE_DMZ_IP=10.10.0.3

SERVER_DNS1_IP=200.123.31.50
SERVER_DNS2_IP=200.123.31.51
SERVER_DNS3_IP=8.8.8.8
SERVER_DNS4_IP=8.8.4.4


NAT_HOST01_IP=0
NAT_HOST02_IP=0
```

Figura 7.0 Declaración de variables.

El script de variables está definido de la siguiente manera (Figura 6.0).

- ✓ Llamamos a dos herramientas que usaremos en ambos script's, TC y IPTABLES.
- ✓ A continuación se describe las interfaces que usaremos en el servidor master.
- ✓ Definimos el Gateway del router que nos brinda la ip pública fija.
- ✓ Definimos nuestros rangos de red que deberá tener nuestra interfaz en la LAN.
- ✓ Ingresamos las ip's de cada interfaz del servidor master.
- ✓ Ingresamos los ip's de nuestro DMZ.
- ✓ Asignamos los DNS que nuestro ISP nos brinda, y agregamos como una opción los DNS de google.
- ✓ Y por último definimos las variables NAT.

**B. AllInOne.sh**

En este archivo se describe el firewall como tal, por lo tanto tiene una secuencia que en el diseño fue mencionado. Se procederá a tomar pantallazos desde la primera línea y describir cada uno de ellos que hace y porque se asigna dentro del script.

**a) #! /bin/sh**

Como se trata de un script ejecutable, debemos decirle que es un Shell binario, por lo tanto se hace el llamado con #! /bin/sh. Las demás líneas se refiere a quien lo implementa, en que compañía, entre otros datos para poder hacer un backup del software desarrollado (Figura 7.0).

```
#! /bin/sh
# Description : script of Intelligent Traffic
# Implemented by : Joseph Veliz Castañeda
# Company : CONEXIONLINUX.COM
# all right recerved
# Made in the CONEXIONLINUX's laboratories - Lima Peru
# Writing by Joseph Veliz
# Server : Main-Server

# Fecha : 15/11/2015
# Services : Firewall - Traffic Control - DNS - DHCP - Asterisk - SQUID -LDAP - HTTP
```

Figura 8.0 Creando un script ejecutable.

### b) Variables

Todo lo que fue asignado en el AllInOne.cfg se ingresa aquí, el único detalle es que se iguala a cero a todas las variables con el objetivo de declarar las variables en el script principal.

```
# ################## Variables ####################################
TC=/sbin/tc
IPTABLES=/sbin/iptables

# Interfaces del servidor AllInOne
IFACE_WAN=0
IFACE_LAN_LOCAL=0
IFACE_LAN_DMZ=0

GATEWAY_SATEL=0

NET_LAN_LOCAL=0
NET_LAN_DMZ=0

# IP del servidor AllInOne
IFACE_LAN_LOCAL_IP=0
IFACE_LAN_DMZ_IP=0
IFACE_WAN_IP=0

SERVER_EMAIL_LOCAL_IP=0
SERVER_APACHE_DMZ_IP=0

SERVER_DNS1_IP=0
SERVER_DNS2_IP=0
SERVER_DNS3_IP=0
SERVER_DNS4_IP=0

NAT_HOST01_IP=0
NAT_HOST02_IP=0

# ################################################################
```

Figura 9.0 Declaración de variables.

### c) Recarga de variables del archivo

Como ya tenemos las variables declaradas, lo que sigue es llamar a los datos de la variable .cfg, esto se hace con un if, preguntándole si él .cfg tiene todos las variables declaradas como lo tiene el .sh, si es así se procede a ejecutar y todas las variables de nuestro .sh cargará con su nuevo parámetro introducido por el .cfg.

```
# Reload Varibles from file
if [ -f /root/AllInOne.v6.02-r1.cfg ];then
        . /root/AllInOne.v6.02-r1.cfg
fi
```

Figura 10.0 Recarga de variables del AllInOne.cfg.

### d) La función start

La función start es donde contiene todo el código principal, para entrar de lleno al firewall se declara una serie de pasos, a continuación se explicara cada paso y porque se usa (Figura 9.0).

**Step 1:** Detenemos el servicio. Para detener el servicio reiniciamos iptables, esto hace que borre todas las reglas del firewall que está actualmente activo y lo deje en modo default. Entonces hacemos el reinicio con la siguiente línea /sbin/systemctl restart iptables.

**Step2:** Flush de reglas. El flush de reglas borra toda las reglas actuales de iptables, en este caso borramos el nat, el mangle y los iptables generales.

**Step3:** flush de cadenas de usuario. Elimina cadenas o reglas de usuarios, si es que fue creado por otros usuarios.

**Step4:** Elimina todas las cadenas especificadas por el usuario.

**Step5:** Reinicia todos los contadores de iptables.

```
tc_start() {
  tc_clean

                # Step 1: Detenemos servicio
                echo "     DETENEMOS EL SERVICIO DE FIREWALL ..."
                /sbin/systemctl restart iptables


                # Step 2:  FLUSH de reglas
                echo "     COMENZAMOS BORRANDO TODAS LAS REGLAS ACTUALES ..."
                iptables -F
                iptables -X
                iptables -t nat -F
                iptables -t nat -X
                iptables -t mangle -F
                iptables -t mangle -X

                # Step 3: Flush the user chain.. if it exists
                if [ "`iptables -L | grep drop-and-log-it`" ]; then
                        iptables -F drop-and-log-it
                fi

                # Step 4:Delete all User-specified chains
                iptables -X

                # Step 5: Reset all IPTABLES counters
                iptables -Z
```

Figura 11.0 Función start y primeros pasos del script.

**Step6:** Cargamos los módulos principales del kernel. Como se trata de un DMZ debemos ejecutar binarios que iptables necesita para que se ejecute.

**Step7:** Aplicando reglas elementales. Se refiere a los problemas del broadcast innecesario, está relacionado con la lentitud de la LAN, y cuellos de botella que podría tener la red en determinados momentos.

**Step8:** Desactivamos IPV6. Nuevo protocolo de red que en la actualidad tiene la mayoría de Sistemas Operativos, pero ¿Por qué desactivarlo?, la respuesta es que si tenemos un proxy adelante y se habilita en el Sistema Operativo el IPV6, se perderá la conexión de red, ya que el proxy y el firewall trabaja con IPV4.

**Otros:** Como se trabaja con IPV4 implementamos una serie de comandos para habilitar todo el tráfico de IPV4, esto permitirá una fluidez optima del protocolo. Las dos primeras líneas habilitamos el TCP y el RP.

El ip_forward es para habilitar el puente de una interfaz a otra.

Habilitar IP dinámica con IPV4, esto se realiza para usuarios con DHCP automático.

```
# Step 6:  Cargamos modulos al KERNEL
echo "     CARGAMOS LOS MODULOS PRINCIPALES DEL KERNEL ..."
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_nat_irc
/sbin/modprobe ip_conntrack_irc
/sbin/modprobe ip_conntrack_ftp    # Needed for FTP (specifically, to a
/sbin/modprobe ip_nat_ftp
/sbin/modprobe iptable_nat

# Step 7:  Aplicando reglas elementales
echo "     APLICANDO REGLAS DE SEGURIDAD ELEMENTALES ..."

# No respondemos a los broadcast.
echo "          No reponder broadcast"
/bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Step 8: Disable IPV6 problems with squid reported and this correct.
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
echo 1 > /proc/sys/net/ipv6/conf/default/disable_ipv6

# Otros
echo "          Otros"
/bin/echo 1 > /proc/sys/net/ipv4/tcp_syncookies
/bin/echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
# El Ip Forwarding
echo "          IP Forwarding enable"
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
# Para usuarios con IP dinamica
echo "          Para usuarios con IP dinamica"
/bin/echo 1 > /proc/sys/net/ipv4/ip_dynaddr
```

Figura 12.0 Función start y primeros pasos del script.

**Protección en contra de errores de mensajes:** si existe un error en la comunicación entre el host y el servidor en IPV4, este no se reporta y se continúa trabajando, se logra hacer esto con el protocolo ICMP.

```
# Proteccion en contra de errores de mensajes
echo "          Proteccion contra malos errores de mensajes"
/bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

Figura 13.0 Función start y primeros pasos del script.

### e) Políticas y reglas del firewall

Las políticas y reglas del firewall de la empresa Conexión Linux SAC, se dividen en dos: reglas duras y reglas blandas, cada una de ellas fue descrita en el diseño del firewall.

### f) Reglas duras

Reglas que no pueden ser modificadas. Empezaremos denegando el acceso total a todas las interfaces que existe en nuestro servidor master. Para luego dar acceso al LOOPBACK, tanto de entrada como de salida, sin restricción ninguna.

Pero ¿qué es LOOPBACK?, es una interfaz interna que tiene el Sistema Operativo de manera lógica, que hacen que diversas aplicaciones del sistema se comuniquen entre sí. Por lo tanto se tiene que mantener el acceso abierto tanto INPUT como OUTPUT.

```
# ##############################################################################################
            # Step 7: POLITICAS Y REGLAS DE FIREWALL -  REGLAS DURAS
            echo "     DENEGANDO ACCESO A LAS CADENAS IPTABLES TODAS INTERFACES eth0 eth1 etc ..."
iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables --policy OUTPUT DROP


echo "configurando INPUT y OUTPUT"
echo "dando acceso total al loopback"
# todo lo que entra y sale desde el loopback se acepta
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Figura 14.0 Reglas duras.

## g) Acceso al servicio SSH

Para dar acceso al servidor master lo haremos mediante SSH con el puerto 25622, para ellos nos vamos al diseño de la red LAN eth1.

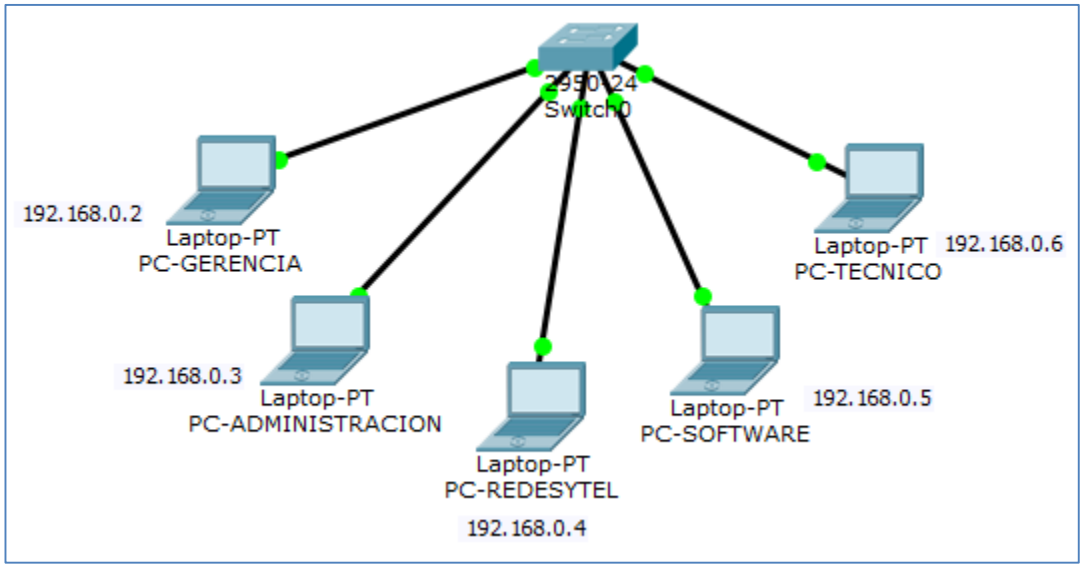



Figura 15.0 Topología lógica de la red LAN.

Como se puede observar según las políticas de seguridad Gerencia y Redesytel deben de tener acceso al servidor master, los otros restantes quedan sin acceso al servidor master. Entonces el código seria de la siguiente manera.

- ✓ **Acceso al servidor SSH (protocolo tcp, puerto 25622) desde LAN**

iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.2 --sport 1024:65535 -d 192.168.0.1 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.2 --dport 1024:65535 -s 192.168.0.1 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 192.168.0.1 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 192.168.0.1 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

Se usa la herramienta iptables con INPUT y OUTPUT, pasaremos a explicar línea por línea.

Hacemos un INPUT a la interfaz eth1 con el protocolo TCP, quien va ingresar es el ip 192.168.0.2 con cualquier puerto al destino 192.168.0.1 con el puerto 25622 en el estado NEW, ESTABLISHED y RELATED, lo cual significa que hará una nueva conexión, lo establece y lo relaciona.

Una vez que el host 192.168.0.2 hace el ingreso al servidor master, el servidor tiene que responder esa petición, por lo tanto hará un OUTPUT.

Hacemos un OUTPUT a la interfaz eth1 con el protocolo TCP, que tiene como destino 192.168.0.2 con un puerto destino cualquiera, desde el ip 192.168.0.1

con puerto 25622 con el estado ESTABLISHED y RELATED, lo cual significa que lo establece y lo relaciona. En este caso no hace una nueva conexión ya que el host usuario inicia la conversación.

Las siguientes dos líneas hace lo mismo con el host 192.168.0.4.

- ✓ **Acceso al servidor SSH (protocolo tcp, puerto 25622) desde WAN**

iptables -t filter -A INPUT -i eth0 -p tcp -s 190.236.178.87 --sport 1024:65535 -d 190.12.87.106 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth0 -p tcp -d 190.236.178.87 --dport 1024:65535 -s 190.12.87.106 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

Según las políticas de seguridad el único que debe acceder al servidor master desde la WAN es el Gerente Mg. Alex Segura, por lo tanto agregamos la regla por la interfaz eth0 desde su ip pública fija de su hogar, lo cual es privado. De igual manera hacemos INPUT y OUTPUT para su host.

**h) Acceso al DMZ (protocolo tcp, puerto 25622) desde la LAN**

iptables -t filter -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.0.5 --sport 1024:65535 -d 10.10.0.3 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.0.5 --dport 1024:65535 -s 10.10.0.3 --sport 25622 -j ACCEPT

iptables -t filter -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 10.10.0.2 --dport 25622 -j ACCEPT

iptables -t filter -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 10.10.0.2 --sport 25622 -j ACCEPT

Hacemos uso del forward para enlazar el puente entre eth1 y eth2, de esta manera solo el encargado de desarrollo de software puede ingresar al servidor apache, y el de redes y telecomunicaciones ingresa al servidor zimbra.

**i) Acceso a otro servidor SSH (protocolo tcp, puerto 25622) desde la LAN**

iptables -t nat -A POSTROUTING -o eth0 -p tcp -s 192.168.0.4 -d 200.114.54.33 --dport 25622 -j SNAT --to 190.12.87.106

iptables -t filter -A FORWARD -i eth1 -o eth0 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 200.114.54.33 --dport 25622 -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 200.114.54.33 --sport 25622 -j ACCEPT

Como el único que puede salir a la WAN hacer soporte es redes y telecomunicaciones, entonces se agrega su ip para que pueda salir a un determinado servidor en la WAN, si hay más servidores que administrar, se agregará en la lista.

**j) Acceso al servicio ICMP**

El servicio ICMP será habilitado para hacer ping al Servidor Master, y también desde la LAN hacia la WAN.

✓ **Permitiendo ICMP al servidor Master (protocolo ICMP) desde LAN**

iptables -t filter -A INPUT -p icmp -i eth1 --icmp-type echo-request -s 192.168.0.0/22 -j ACCEPT

iptables -t filter -A OUTPUT -p icmp -o eth1 --icmp-type echo-reply -d 192.168.0.0/22 -j ACCEPT

Habilitamos el ping para todo la red LAN al Servidor Master, esto nos ayudara a identificar si el servidor está apagado o está caído.

✓ **Permitiendo ICMP a Internet (protocolo ICMP) desde LAN**

iptables -t nat -A POSTROUTING -p icmp -o eth0 -s 192.168.0.0/22 -j SNAT --to 190.12.87.106

iptables -t filter -A FORWARD -i eth1 -o eth0 -p icmp -s 192.168.0.0/22 -d 0/0 -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -p icmp -d 192.168.0.0/22 -s 0/0 -j ACCEPT

Habilitamos el ping a la WAN, para poder verificar si los DNS del ISP están funcionando, como también otros servidores si están disponibles.

**k) Acceso de los DNS externos al Servidor Master**

Tenemos que habilitar el acceso de los DNS del ISP al Servidor Master para que pueda conectarse al internet, incluiremos además los DNS del servidor de Google.

✓ **Permitiendo DNS del servidor Master (protocolo DNS) a DNS Externos**

iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 8.8.8.8 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 8.8.8.8 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 8.8.4.4 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 8.8.4.4 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 190.12.72.226 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 190.12.72.226 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 190.12.72.227 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 190.12.72.227 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

Damos acceso a los DNS de Google y los DNS del ISP, así aseguramos que si el servidor del ISP se cae, podemos coger los DNS de google y no se pierde la conexión a internet.

✓ **Permitiendo DNS desde LAN (protocolo DNS) a DNS Master**

iptables -t filter -A INPUT -i eth1 -p udp -s 192.168.0.0/22 --sport 1024:65535 -d 192.168.0.1 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p udp -d 192.168.0.0/22 --dport 1024:65535 -s 192.168.0.1 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

Como el servidor master tiene un DNS propio, debemos darle acceso a todos los host de la LAN, para que puedan usar el DNS propio del Servidor Master.

**l) Acceso al puerto 80 y 443 para navegación**

Habilitaremos navegación para estos dos puertos de internet.

✓ **Habilitando navegación SQUID protocolo HTTP (puerto 80)**

iptables -t filter -A OUTPUT -o eth0 -p tcp -s 190.12.87.106 --sport 1024:65535 -d 0/0 --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A INPUT -i eth0 -p tcp -d 190.12.87.106 --dport 1024:65535 -s 0/0 --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

Con estos filtros podremos ingresar a páginas http desde la LAN.

✓ **Habilitando navegación SQUID protocolo HTTPS (puerto 443)**

iptables -t filter -A OUTPUT -o eth0 -p tcp -s 190.12.87.106 --sport 1024:65535 -d 0/0 --dport 443 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A INPUT -i eth0 -p tcp -d 190.12.87.106 --dport 1024:65535 -s 0/0 --sport 443 -m state --state ESTABLISHED,RELATED -j ACCEPT

Con estos filtros podremos ingresar a páginas http desde la LAN.

**m) Acceso al proxy SQUID**

El proxy trabaja con un determinado puerto e ip, por lo tanto debemos darle acceso para que puedan navegar los host clientes.

✓ **Habilitando conexión Browsers LAN puerto 3128 al SQUID Master**

iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.0/22 --sport 1024:65535 -d 192.168.0.1 --dport 3128 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.0/22 --dport 1024:65535 -s 192.168.0.1 --sport 3128 -m state --state ESTABLISHED,RELATED -j ACCEPT

Con este filtro todos los usuarios podrán tener acceso a la INTERNET.

**n) Acceso a los servidores DMZ desde el Servidor Master**

Debemos implementar acceso para administrar desde el servidor master todo el DMZ.

✓ **Permitiendo acceso al servidor DMZ (protocolo TCP, puerto 25622) desde Servidor Master**

iptables -t filter -A OUTPUT -o eth2 -p tcp -d 10.10.0.0/22 --dport 1024:65535 -s 10.10.0.1 --sport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth2 -p tcp -s 10.10.0.0/22 --sport 1024:65535 -d 10.10.0.1 --dport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

Con estas reglas el quien ingrese al servidor master podrá acceder a los DMZ.

**o) Acceso a los host NAT**

Cuando un host usuario necesita administrar páginas del estado por ejemplo SIAF, estos servidores del SIAF tiene una ip publica fija, que tenemos que averiguar cuáles son, por lo tanto se debe habilitar el acceso total de ese usuario, hasta que se averigüe cuáles son esas ip publicas fijas.

✓ **Habilitando NAT para HOST de emergencia**

for NAT in ${NAT_HOST_IP[@]}; do

if [ "$NAT" != "0" ]; then

```
iptables -t nat -A POSTROUTING -s $NAT -d 0/0 -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -s $NAT -d 0/0 -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -d $NAT -s 0/0 -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

fi
done
```

Realizaremos un for, ya que se trata más de una variable, y de ahí realizaremos un if, que si la variable NAT tiene una ip realice el filtro, en caso contrario termine y pase a la siguiente línea.

**p) Reglas blandas**

Son reglas que pueden ser modificadas cada cierto tiempo, estas reglas fueron creadas en caso de que existiera nuevos filtros que se debieran hacer, como es de suponer en el caso del SIAF, que son sistemas del estado e implementan su propio servidores.

Entonces si un usuario hace referencia que no puede acceder a los sistemas del estado, se le hace un NAT, seguidamente se busca cuáles son esas IP publicas fijas, y se agrega a estas reglas blandas, una vez que se obtiene las IP'S públicas fijas se quita el NAT a ese usuario.

✓ **Habilitando servicios SIAF**

```
if [ "$HOST_SIAF_IP" != "0" ]; then
```

```
iptables -t nat -A POSTROUTING -s $HOST_SIAF_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -s $HOST_SIAF_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -d $HOST_SIAF_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

else

for SIAF in ${SERV_SIAF_IP[@]}; do

if [ "$SIAF" != "0" ]; then

iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SIAF -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -d $SIAF -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SIAF -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

fi
done
fi
```

Como se observa primero se realiza un if al HOST_SIAF_IP, pero ¿Qué es la variable HOST_SIAF_IP?, es el IP del host usuario, el cual se le hace un NAT para que pueda usar el SIAF hasta que se encuentren todas las IP'S de los servidores.

Si la variable HOST_SIAF_IP, no tiene el IP, entonces hacemos un FOR a todas los IP'S del servidor, habilitando las reglas y el acceso a esa lista de IP'S.

De esta manera podemos hacer para cualquier entidad del estado que tiene sus propios servidores. Solo será cosa de agregar los IP'S en el archivo .cfg y realizar el if en el archivo .sh.

- ✓ **Habilitando protocolo HTTP o HTTPS para puertos diferentes del 80 o 445**

Existen páginas web para navegar que no solo usan el puerto 80 y 445, para ello agregamos una regla que no es necesaria, pero útil lo siguiente.

```
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p tcp -m multiport -s $IFACE_WAN_SATEL_IP -d 0/0 --dports 7777:7779,8080,2082 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p tcp -m multiport -d $IFACE_WAN_SATEL_IP -s 0/0 --dports 7777:7779,8080,2082 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Esto hace que los usuarios que usan el proxy puedan tener acceso a todas las páginas que se trabaja

**q) Funcion clean**

La función clean, como su propio nombre lo dice, limpia todas las reglas de iptables y acepta el INPU, OUTPUT y FORWARD, en otras palabras baja el firewall para que todos puedan entrar y salir de la LAN.

```
tc_clean()
{
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
echo "0" > /proc/sys/net/ipv4/ip_forward
}
```

Hacemos un echo para ingresar el valor cero al ipv4 en modo forward.

**r) Funcion stop**

Esta función llama a la función tc_clean, ya que realiza lo mismo.

```
tc_stop()
{
        tc_clean
}
```

**s) Funcion restart**

Esta función llama a la función stop, espera un segundo, y inicia nuevamente con la funcion start.

```
tc_restart() {
```

```
        tc_stop
        sleep 1
        tc_start
}
```

**t) Agregamos casos de uso**

```
case "$1" in
        start)
                echo -n "Starting Intelligent Traffic: "
                echo ""
                tc_start
                echo "CONEXION LINUX started Intelligent Traffic v 1.2 : done"
                echo "Write by Joseph Veliz - Conexion Linux SAC laboratories"
                echo "done"
                ;;

        stop)
                echo -n "Stopping Intelligent Traffic: "
                tc_stop
                echo "done"
                ;;

        restart)
                echo -n "Restarting Intelligent Traffic: "
                tc_restart
                echo "done"
                ;;

        show)
                tc_show
```

```
        ;;

    nat)
        tc_nat
        ;;

    clean)
        tc_clean
        ;;

  *)
        echo "Usage: /etc/init.d/AllInOne {start|stop|restart|show|nat|clean}"
        ;;

esac
exit 0
```

# CAPÍTULO 6

# 6

# APLICACIÓN Y PRUEBAS

## 6.1. Aplicación del firewall

Una vez construido el firewall con iptables, se pasa a realizar una serie de instrucciones para dejarlo funcionando correctamente, los pasos son lo siguiente:

### a) Permisos del archivo AllInOne.cfg y AllInOne.sh

En el servidor master, se tiene usuarios creados y cada uno de ellos tienen privilegios 0 (no tienen privilegios de administración del SO.), entonces si hablamos de seguridad, el firewall debe ser ejecutado por ciertos usuarios, en este caso por Gerencia y por el encargado de Redes y Telecomunicaciones.

Entonces procederemos a crear un grupo de usuarios en el Sistema Operativo y le daremos privilegios de ejecución al script del firewall.

Los usuarios asegura y soporte, lo ingresamos al grupo administración:

```
#groupadd –g 5000 administracion
#usermod –aG administracion asegura
#usermod –aG administracion soporte
```

Con estos dos comandos ingresamos a los dos usuarios al grupo administración, ahora damos permiso de ejecución a ese grupo:

```
chmod g+x AlliOne.sh.
chmod g+x AlliOne.cfg.
```

**b) Ejecución del script AllInOne.sh al iniciar el Sistema Operativo**

Cuando se reinicia el Sistema Operativo GNU/Linux el script creado no se ejecutará automáticamente, ya que no es parte del systemd como arrancador, entonces si queremos que el script creado se inicie automáticamente al iniciar el Sistema Operativo, se debe ingresar el siguiente comando:

./root/AllInOne.sh start

A la carpeta /etc/rc.local

Con esto le decimos que inicie el script al iniciar el Sistema Operativo.

**c) Comandos para administrar el script AllInOne.sh**

El script tiene 4 funciones: start, restart, stop y clean, entonces se ejecutará los siguientes comandos:

./AllInOne.sh start
./AllInOne.sh stop
./AllInOne.sh restart
./AllInOne.sh clean

## 6.2. Pruebas

Las pruebas del firewall se realizaran de dos maneras, vulnerabilidad y lentitud. En el caso de vulnerabilidad se probará que el firewall restringe los accesos no permitidos, minimice los ataques y solo tengan acceso las personas autorizadas. Mientras en la lentitud se verificará que el broadcast este desactivado, que los puertos y ip's estén habilitados solo lo necesario, y que el proxy trabaje sin problemas con el firewall.

### 6.2.1. Vulnerabilidad

#### A. WAN

Primero trataremos de ingresar al IP pública de Conexión Linux mediante SSH, con el PUTTY, desde una ip desconocida.

Como primer paso haremos un ping a la IP publica:

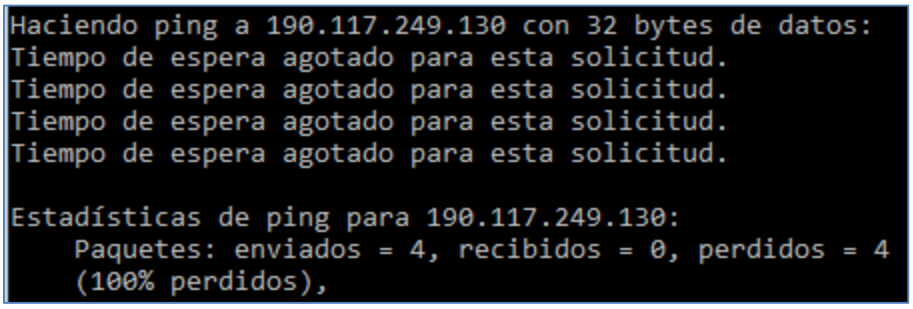



Figura 1.0 Ping desde la WAN al servidor master.

Podemos observar que los paquetes no son recibidos, esto quiere decir que el servidor está apagado o lo más lógico que desactivaron el protocolo ICMP para evitar saturación del servidor master.

Ingresamos con el puTTY a la IP pública con el puerto 25622.

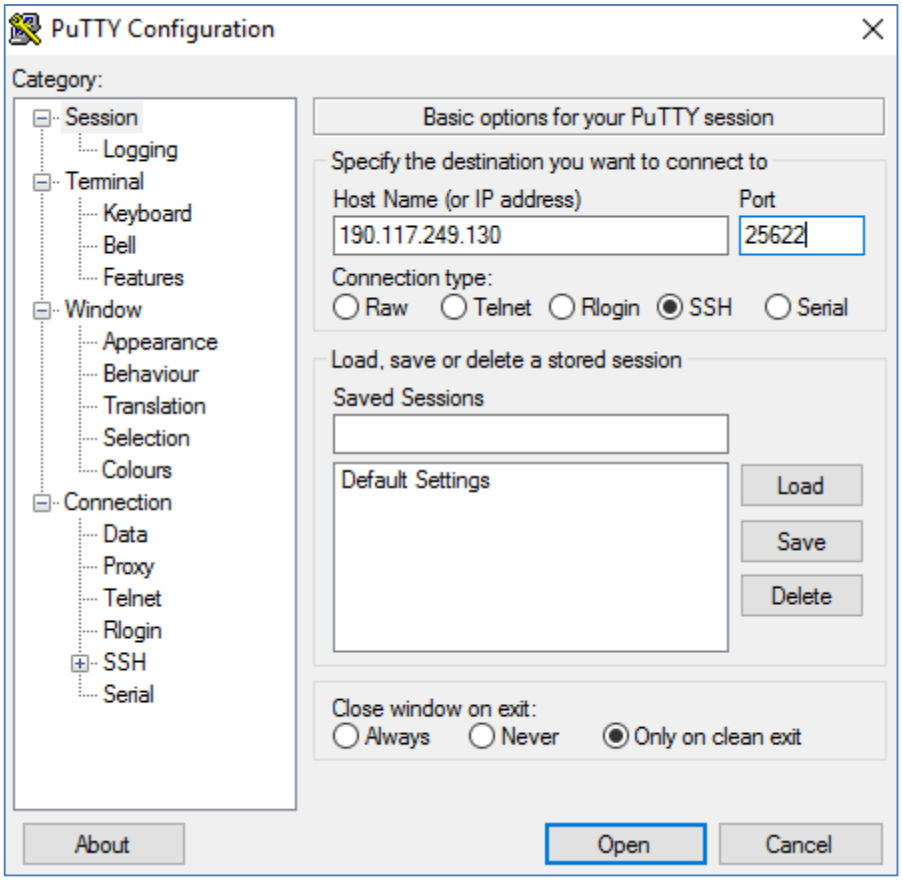



Figura 2.0 Accediendo por Putty al servidor.

Error en la red, no se puede acceder al servidor master.

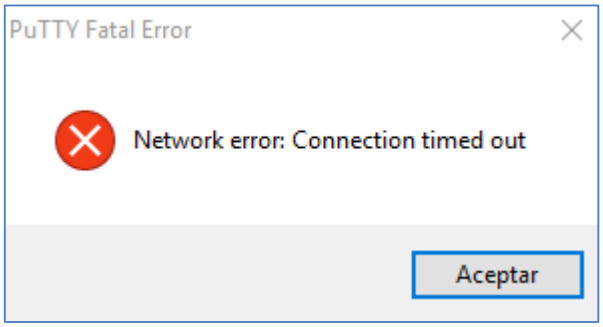



Figura 3.0 Mensaje de error.

Esto nos garantiza que la conexión al servidor master solo podrá acceder el gerente desde la WAN, otro usuario estará fuera de conexión. Por lo cual el

firewall nos garantiza que los ataques externos se minimizan, a no ser por la red del Gerente en su hogar.

**B. LAN**

En la red LAN de la empresa está habilitado el protocolo ICMP, por lo tanto todos los host usuario pueden hacer ping al servidor master.

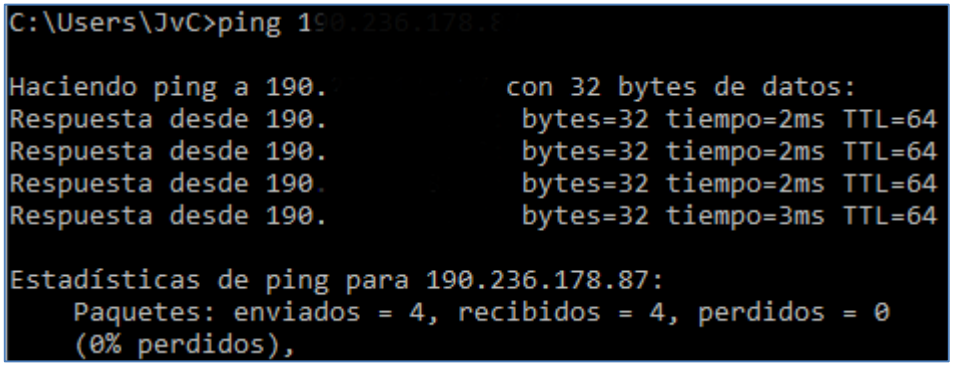



Figura 4.0 Haciendo ping al servidor master desde la LAN.

Los que tendrán acceso al servidor master son Gerencia y redes. Mientras el encargado de desarrollo de software tendrá acceso al servidor apache.


Figura 5.0 Ingreso por putty.

Con estas reglas se pretende minimizar los ataques desde la LAN, ya que son dos usuarios con acceso al servidor master.

### 6.2.2. Lentitud

El firewall ayuda a minimizar la lentitud, según Red Hat un buen firewall contiene reglas predeterminadas que ayuda al buen funcionamiento de la red LAN, como por ejemplo bloquear el broadcast innecesario, deshabilitar las versionas de protocolo que no se trabajen, limpiar reglas de firewall en todos los contextos y bloquear los mensajes de error del protocolo. Solo son algunos ejemplos que recomienda Red Hat.

#### a) Iperf3

Iperf3 es una herramienta que se usa para medir el tráfico de red, se tiene que instalar en el servidor como en el usuario para poder medir el tráfico.

**Desde el servidor activado el firewall:**

```
[root@localhost ~]# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.171, port 40244
[  5] local 192.168.1.170 port 5201 connected to 192.168.1.171 port 40245
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  53.1 MBytes   445 Mbits/sec
[  5]   1.00-2.00   sec  45.4 MBytes   380 Mbits/sec
[  5]   2.00-3.00   sec  62.5 MBytes   526 Mbits/sec
[  5]   3.00-4.00   sec  47.5 MBytes   398 Mbits/sec
[  5]   4.00-5.00   sec  59.8 MBytes   502 Mbits/sec
[  5]   5.00-6.00   sec  62.0 MBytes   519 Mbits/sec
[  5]   6.00-7.00   sec  47.5 MBytes   398 Mbits/sec
[  5]   7.00-8.00   sec  61.7 MBytes   517 Mbits/sec
[  5]   8.00-9.00   sec  61.8 MBytes   520 Mbits/sec
[  5]   9.00-10.00  sec  60.6 MBytes   510 Mbits/sec
[  5]  10.00-10.04  sec  1.73 MBytes   400 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec   564 MBytes   471 Mbits/sec                  receiver
```

Figura 6.0 Haciendo iperf3 desde el servidor.

Se puede observar que recibe 564 MBytes desde el host cliente.

**Desde el host usuario activado el firewall:**

```
[root@localhost ~]# iperf3 -c 192.168.1.170
Connecting to host 192.168.1.170, port 5201
[  4] local 192.168.1.171 port 40245 connected to 192.168.1.170 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  56.0 MBytes   470 Mbits/sec   45    198 KBytes
[  4]   1.00-2.00   sec  45.0 MBytes   378 Mbits/sec    1    276 KBytes
[  4]   2.00-3.00   sec  63.1 MBytes   529 Mbits/sec   41    247 KBytes
[  4]   3.00-4.00   sec  47.0 MBytes   394 Mbits/sec   91    194 KBytes
[  4]   4.00-5.00   sec  59.7 MBytes   501 Mbits/sec    0    228 KBytes
[  4]   5.00-6.00   sec  62.2 MBytes   522 Mbits/sec    0    247 KBytes
[  4]   6.00-7.00   sec  47.5 MBytes   398 Mbits/sec   91    204 KBytes
[  4]   7.00-8.00   sec  61.4 MBytes   515 Mbits/sec    0    236 KBytes
[  4]   8.00-9.00   sec  62.1 MBytes   521 Mbits/sec   45    181 KBytes
[  4]   9.00-10.00  sec  60.4 MBytes   507 Mbits/sec    0    202 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   564 MBytes   474 Mbits/sec  314             sender
[  4]   0.00-10.00  sec   564 MBytes   473 Mbits/sec                  receiver
```

Figura 7.0 Haciendo iperf3 desde el host usuario.

El host cliente envía 564 MBytes y recibe 564 MBytes de transferencia. Como se puede observar al tener el firewall corriendo el tráfico de red se limpia y evita que los virus en la red puedan salir hacia la WAN, no solo eso, el broadcast es rechazado por el servidor master y el protocolo IPV6 es deshabilitado por completo, de esta manera tendremos un tráfico de red más limpia y ordenada.

### 6.2.3. Medicion de variables

#### A. Variable independiente

Firewall con iptables.

#### a) Enrutador con capacidades de filtrado

Se refiere si al llegar un paquete al servidor master será enrutado o no, como por ejemplo la autorización del ip origen e ip destino.

De acuerdo a este punto de la WAN hacia el servidor master existen solo dos conexiones habilitadas, los demás son negados, y desde la LAN también dos conexiones:

- ✓ **De la WAN hacia el servidor master**

Numero de conexiones habilitadas: 2 ip's.

Numero de conexiones deshabilitadas: todo lo demás.

- ✓ **De la LAN hacia el servidor master**

Numero de conexiones habilitadas: 2 ip's.

Numero de conexiones deshabilitadas: todo lo demás.

**b) Servidor de control a nivel circuito**

Consiste en validar una sesión, un servidor de control a nivel circuito examina cada establecimiento de conexión a nivel transporte, para asegurarse que lo siguió legítimamente a través del llamado triple saludo o handshake.

De acuerdo al concepto se puede medir la conexión establecida entre un host usuario y el servidor master, para esto se usa el protocolo SSH, recordemos que SSH trabaja en forma segura en la transferencia de comunicación entre dos host's, a menos que sea atacado por REPLAY.

- ✓ **De la WAN hacia el servidor master**

Numero de conexiones establecidas: 2 ip’s

Numero de conexiones no establecidas: todo lo demás

- ✓ **De la LAN hacia el servidor master**

Numero de conexiones establecidas: 2 ip’s

Numero de conexiones no establecidas: todos lo demás

**B. Variable dependiente**

**Vulnerabilidad** y lentitud

**a) Escaneo**

El escaneo se realiza con la herramienta wireshark, es una herramienta que me permite filtrar todo la comunicación del servidor, por este medio es fácil detectar intrusos en tu red, o intentos de penetración, el wireshark se ejecutará de la siguiente manera:

```
[root@localhost ~]# tshark -i enp0s3 -R "tcp.port == 22"
tshark: -R without -2 is deprecated. For single-pass filtering use -Y.
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp0s3'
  2 0.015072397 192.168.1.34 -> 192.168.1.170 TCP 60 58279 > ssh [ACK] Seq=1 Ack=1 Win=62 Len=0
  3 0.527121118 192.168.1.170 -> 192.168.1.34 SSH 214 Encrypted response packet len=160
  4 0.577604576 192.168.1.34 -> 192.168.1.170 TCP 60 58279 > ssh [ACK] Seq=1 Ack=161 Win=61 Len=0
  5 0.984795569 192.168.1.170 -> 192.168.1.34 SSH 294 Encrypted response packet len=240
  6 1.035605793 192.168.1.34 -> 192.168.1.170 TCP 60 58279 > ssh [ACK] Seq=1 Ack=401 Win=61 Len=0
 11 1.485593824 192.168.1.170 -> 192.168.1.34 SSH 294 Encrypted response packet len=240
```

Figura 8.0 Haciendo tshark al servidor master.

Como se puede observar el único puerto para entrar por el protocolo ssh, es por el puerto 25622, solo que en la figura se muestra como ejemplo el puerto 22. Entonces medimos las variables:

Número de trazas al día: 4
Número de ingresos malicioso por día: 0

**b) Obtencion de acceso**

Debemos hacer un scanner cada cierto tiempo al servidor master, para verificar que usuarios están conectados al servidor como administrador.

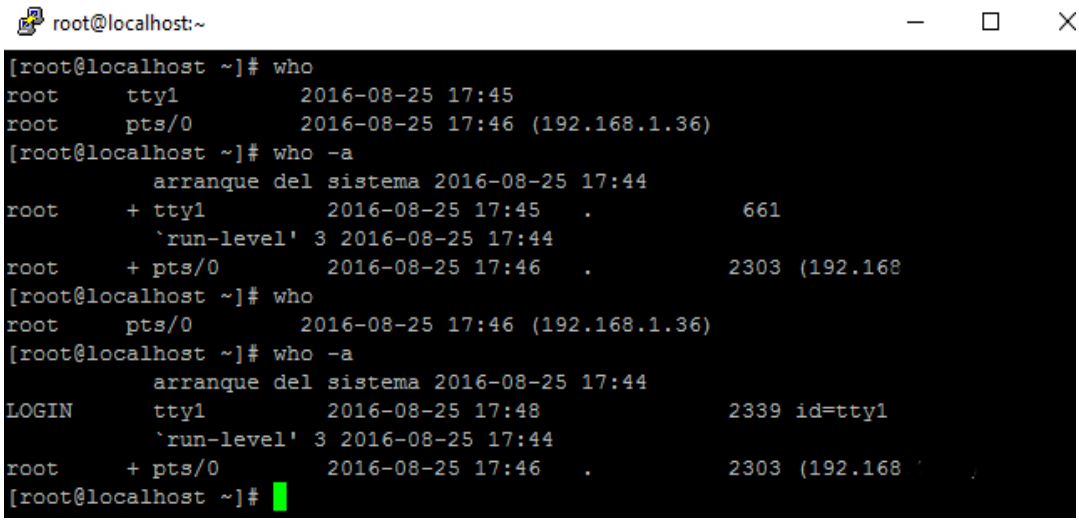



Figura 9.0 Haciendo who al servidor master.

Y esto se realiza con el comando who, me indica quienes están conectados al servidor en modo login.

✓ **De la WAN hacia el servidor master**

Numero de usuarios conectados (login): 2 ip's

Numero de usuarios no conectados (login): todos lo demás

- ✓ **De la LAN hacia el servidor master**

Numero de usuarios conectados (login): 2 ip’s

Numero de usuarios no conectados (login): todos los demás

**c) Manteniendo el acceso**

Para evitar que el sistema operativo sea un zombi, debemos cambiar la contraseña cada 2 meses, esto se realizará por parte del administrador de red y generará contraseñas con un diccionario.

Numero de contraseñas cambiadas por año: 6

**d) Encubrimiento de rastros**

Cada usuario que ingrese al sistema operativo GNU/Linux queda registrado en el SSH, los datos que se guardan son: fecha, hora e ip. Cada comando digitado se guarda en la SHELL de Linux llamado history, si el administrador de red restringio permisos a ese archivo, es imposible borrar todo lo que digita el atacante, entonces se puede medir por número de rastros protegidos:

Numero de rastros protegidos en el sistema: 2
Numero de rastros no protegidos en el sistema: 0

Vulnerabilidad y **lentitud**

### a) Capacidades de transporte de la red LAN

En la red LAN se usa el cable UTP Cat-6 que garantiza al usuario velocidades de gigabit que permiten transmisiones de datos de hasta 10 gigabits por segundo. Todos estos datos son dentro de los 100 metros recomendados por CISCO.

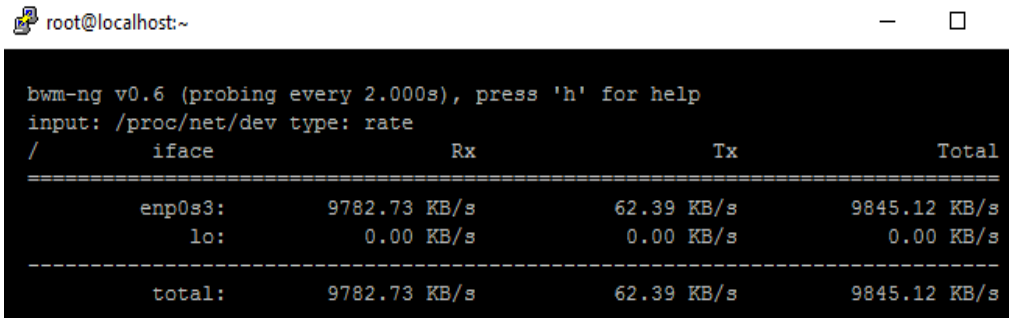



Figura 10.0 Bwm-ng al servidor master.

### b) Banda ancha del ISP

Se mide también con la herramienta bwm-ng, los KB/s recibidos.

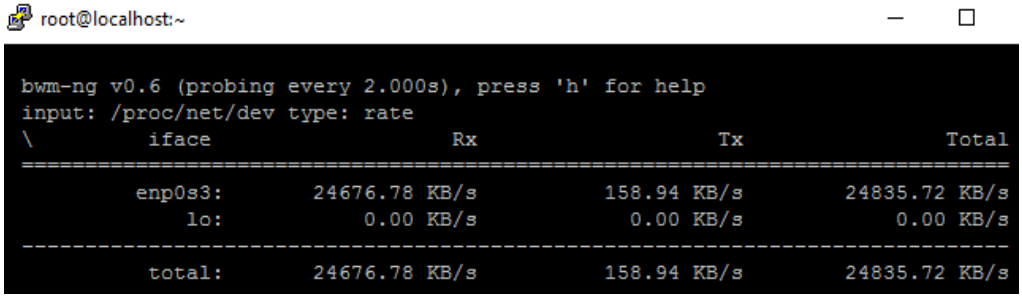



Figura 10.0 Bwm-ng al servidor master.

### c) Velocidad y capacidad del servidor de red

Se obtiene las propiedades de la tarjeta de red, hasta cuanto puede transmitir datos y hasta cuanto puede recibir datos.

```
[root@localhost ~]# ethtool enp0s3
Settings for enp0s3:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: off (auto)
        Supports Wake-on: umbg
        Wake-on: d
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

Figura 11.0 Propiedades de la interfaz enp0s3.

### d) Demanda de usuarios en un momento dado

El problema común del broadcast innecesario, podemos observar el trafico que causa el broadcast (Figura 12 y 13).

root@localhost:~

```
bwm-ng v0.6 (probing every 2.000s), press 'h' for help
input: /proc/net/dev type: rate
|         iface                   Rx                   Tx                Total
==============================================================================
         enp0s3:           1.40 Kb/s            0.83 Kb/s            2.24 Kb/s
             lo:           0.00 Kb/s            0.00 Kb/s            0.00 Kb/s
------------------------------------------------------------------------------
          total:           1.40 Kb/s            0.83 Kb/s            2.24 Kb/s
```

Figura 12.0 Trafico de broadcast en la interfaz enp0s3.

```
bwm-ng v0.6 (probing every 2.000s), press 'h' for help
input: /proc/net/dev type: rate
/         iface                   Rx                   Tx                Total
==============================================================================
        enp0s3:           1.17 Kb/s            0.00 Kb/s            1.17 Kb/s
            lo:           0.00 Kb/s            0.00 Kb/s            0.00 Kb/s
------------------------------------------------------------------------------
         total:           1.17 Kb/s            0.00 Kb/s            1.17 Kb/s
```

Figura 13.0 Trafico sin broadcast en la interfaz enp0s3.

### e) Cantidad de tráfico que compita en la red

Se pudo demostrar esto con la sección anterior de iperf3, donde se mide la transferencia de un archivo del servidor al host usuario (Figura 14 y 15).

```
[root@localhost ~]# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.171, port 40244
[  5] local 192.168.1.170 port 5201 connected to 192.168.1.171 port 40245
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  53.1 MBytes   445 Mbits/sec
[  5]   1.00-2.00   sec  45.4 MBytes   380 Mbits/sec
[  5]   2.00-3.00   sec  62.5 MBytes   526 Mbits/sec
[  5]   3.00-4.00   sec  47.5 MBytes   398 Mbits/sec
[  5]   4.00-5.00   sec  59.8 MBytes   502 Mbits/sec
[  5]   5.00-6.00   sec  62.0 MBytes   519 Mbits/sec
[  5]   6.00-7.00   sec  47.5 MBytes   398 Mbits/sec
[  5]   7.00-8.00   sec  61.7 MBytes   517 Mbits/sec
[  5]   8.00-9.00   sec  61.8 MBytes   520 Mbits/sec
[  5]   9.00-10.00  sec  60.6 MBytes   510 Mbits/sec
[  5]  10.00-10.04  sec  1.73 MBytes   400 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec   564 MBytes   471 Mbits/sec                  receiver
```

Figura 14.0 Haciendo iperf3 desde el servidor.

```
[root@localhost ~]# iperf3 -c 192.168.1.170
Connecting to host 192.168.1.170, port 5201
[  4] local 192.168.1.171 port 40245 connected to 192.168.1.170 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  56.0 MBytes   470 Mbits/sec   45    198 KBytes
[  4]   1.00-2.00   sec  45.0 MBytes   378 Mbits/sec    1    276 KBytes
[  4]   2.00-3.00   sec  63.1 MBytes   529 Mbits/sec   41    247 KBytes
[  4]   3.00-4.00   sec  47.0 MBytes   394 Mbits/sec   91    194 KBytes
[  4]   4.00-5.00   sec  59.7 MBytes   501 Mbits/sec    0    228 KBytes
[  4]   5.00-6.00   sec  62.2 MBytes   522 Mbits/sec    0    247 KBytes
[  4]   6.00-7.00   sec  47.5 MBytes   398 Mbits/sec   91    204 KBytes
[  4]   7.00-8.00   sec  61.4 MBytes   515 Mbits/sec    0    236 KBytes
[  4]   8.00-9.00   sec  62.1 MBytes   521 Mbits/sec   45    181 KBytes
[  4]   9.00-10.00  sec  60.4 MBytes   507 Mbits/sec    0    202 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   564 MBytes   474 Mbits/sec  314             sender
[  4]   0.00-10.00  sec   564 MBytes   473 Mbits/sec                  receiver
```

Figura 15.0 Haciendo iperf3 desde el host usuario.

## f) Capacidades de la computadora del usuario final

El hardware de los usuario son core i5, por lo cual mostraremos las propiedades de la tarjeta de red para saber su transferencia de datos.

```
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>wmic nic where "speed is not null" get name,speed
Name                                  Speed
Intel(R) Centrino(R) Wireless-N 1000  54000000
Realtek PCIe FE Family Controller     9223372036854775807
Microsoft ISATAP Adapter              100000
Microsoft Teredo Tunneling Adapter    100000
```

Figura 16.0 Velocidad de la tarjeta de red del host usuario.

### 6.2.4. Versiones del AllInOne

Se llegaron a tener 3 versiones del firewall, esto fue a medida de ajustar algunos parámetros y de facilitar la automatización del script, no se descarta tener que mejorar el script y llegar a tener más versiones nuevas, todo se modificará de acuerdo a las necesidades de la empresa y requerimientos de los usuarios. A continuación mostraremos las tres versiones que pasó el firewall.

**A. Versión 1 del AllInOne:**

La primera versión del firewall no tiene variables definidas, se desarrolló directamente con las reglas duras, no se consideró las reglas blandas, lo cual si habría un pequeño cambio más adelante, se tendría que modificar línea por línea. Por lo tanto solo existe un solo archivo que es el AllInOne.sh (Anexo).

**B. Versión 2 del AllInOne:**

En esta versión integramos las variables en otro archivo .cfg, como también integramos las reglas blandas, ya que son necesarios en caso de abrir puertos y ip´s que no están determinados por las políticas de seguridad. Por lo tanto existen dos archivos el AllInOne.sh y el AllInOne.cfg (Anexo).

**C. Versión 3 del AllInOne:**

En la versión 3 del AllInOne (versión final), se incluye el código programación en bash, esto hace que las variables no se repita uno en uno, por lo contrario incluir if y for anillados hará que el código sea corto y entendible. Por lo tanto existen dos archivos AllInOne.sh y AllInOne.cfg (Anexo).

A continuación se muestra la versión 3 del firewall, versión final:

**a) AllInOnev3.sh**

```
############################ Variables ############################
TC=/sbin/tc
IPTABLES=/sbin/iptables

# Interfaces del servidor AllInOne
IFACE_WAN=0
IFACE_LAN_DMZ=0
IFACE_LAN_LOCAL=0

# IP del modem satelital
GATEWAY_MODEM=0

# RED LAN AND DMZ
NET_LAN_DMZ=0
NET_LAN_LOCAL=0

# IP del servidor AllInOne
IFACE_LAN_LOCAL_IP=0
IFACE_LAN_DMZ_IP=0
IFACE_WAN_IP=0
IP_ADMINISTRADOR=0
IP_GERENCIA=0
IP_SOFTWARE=0
IP_WAN_ADMINISTRADOR=0
IP_WAN_GERENCIA=0
IP_DMZ_APACHE=0

#DNS
SERVER_DNS_IP[0]=0
```

```
SERVER_DNS_IP[1]=0
SERVER_DNS_IP[2]=0
SERVER_DNS_IP[3]=0

# IP de los servidores del SIAF
HOST_SIAF_IP=0
SERV_SIAF_IP[0]=0
SERV_SIAF_IP[1]=0
SERV_SIAF_IP[2]=0
SERV_SIAF_IP[3]=0
SERV_SIAF_IP[4]=0
SERV_SIAF_IP[5]=0
SERV_SIAF_IP[6]=0
SERV_SIAF_IP[7]=0
SERV_SIAF_IP[8]=0
SERV_SIAF_IP[9]=0

# IP de los servidores del SIS
HOST_SIS_IP=0
SERV_SIS_IP[0]=0

# IP de los servidores del RENIEC
HOST_RENIEC_IP=0
SERV_RENIEC_IP[0]=0

# IP con nat simple
NAT_HOST_IP[0]=0
NAT_HOST_IP[1]=0

#################################################################
# Reload Varibles from file
```

```
if [ -f /root/AllInOne.internetsalma.v6.02-r1.cfg ];then
. /root/AllInOne.internetsalma.v6.02-r1.cfg
fi

tc_start() {
tc_clean
# Step 1: Detenemos servicio
echo " DETENEMOS EL SERVICIO DE FIREWALL ..."
/sbin/service iptables restart
# Step 2: FLUSH de reglas
echo " COMENZAMOS BORRANDO TODAS LAS REGLAS ACTUALES ..."
iptables -F # Flush all chains
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
# Step 2: Flush the user chain.. if it exists
if [ "`iptables -L | grep drop-and-log-it`" ]; then
iptables -F drop-and-log-it
fi

# Step 2:Delete all User-specified chains
iptables -X

# Step 3: Reset all IPTABLES counters
iptables -Z

# Step 4: Cargamos modulos al KERNEL
echo " CARGAMOS LOS MODULOS PRINCIPALES DEL KERNEL ..."
/sbin/modprobe ip_conntrack
```

```
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_nat_irc
/sbin/modprobe ip_conntrack_irc
/sbin/modprobe ip_conntrack_ftp # Needed for FTP (specifically, to allow incoming ftp-data connections)
/sbin/modprobe ip_nat_ftp
/sbin/modprobe iptable_nat

# Step 5: Aplicando reglas elementales
echo " APLICANDO REGLAS DE SEGURIDAD ELEMENTALES ..."
# Quitamos los pings.
# echo " Quitando ping"
# /bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# No respondemos a los broadcast.
echo " No reponder broadcast"
/bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Step 6: Disable IPV6 problems with squid reported and this correct.
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
echo 1 > /proc/sys/net/ipv6/conf/default/disable_ipv6
# Otros
echo " Otros"
/bin/echo 1 > /proc/sys/net/ipv4/tcp_syncookies
/bin/echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
# El Ip Forwarding
echo " IP Forwarding enable"
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
# Para usuarios con IP dinamica
echo " Para usuarios con IP dinamica"
/bin/echo 1 > /proc/sys/net/ipv4/ip_dynaddr
```

```
# Proteccion en contra de errores de mensajes
echo " Proteccion contra malos errores de mensajes"
/bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
#Code Listing 2.7: Enable reverse path filtering

#################################################################
# Step 7: POLITICAS Y REGLAS DE FIREWALL - REGLAS DURAS
echo " DENEGANDO ACCESO A LAS CADENAS IPTABLES TODAS INTERFACES eth0 eth1 etc ..."
iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables --policy OUTPUT DROP

echo "configurando INPUT y OUTPUT"
echo "dando acceso total al loopback"
# todo lo que entra y sale desde el loopback se acepta
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

###############--SSH--###############

echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde lan "

iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p tcp -s $IP_GERENCIA --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p tcp -d $IP_GERENCIA --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p tcp -s $IP_ADMINISTRADOR --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p tcp -d $IP_ADMINISTRADOR --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde wan "

iptables -t filter -A INPUT -i $IFACE_WAN -p tcp -s $IP_WAN_ADMINISTRADOR --sport 1024:65535 -d $IFACE_WAN_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_WAN -p tcp -d $IP_WAN_ADMINISTRADOR --dport 1024:65535 -s $IFACE_WAN_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -t filter -A INPUT -i $IFACE_WAN -p tcp -s $IP_WAN_GERENCIA --sport 1024:65535 -d $IFACE_WAN_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_WAN -p tcp -d $IP_WAN_GERENCIA --dport 1024:65535 -s $IFACE_WAN_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo acceso a otro servidor SSH (protocolo tcp, puerto 25622) desde lan "

iptables -t nat -A POSTROUTING -o $IFACE_WAN -p tcp -s $IP_ADMINISTRADOR -d 0/0 --dport 25622 -j SNAT --to $IFACE_WAN_IP
iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_WAN -p tcp -s $IP_ADMINISTRADOR --sport 1024:65535 -d 0/0 --dport 25622 -j ACCEPT

iptables -t filter -A FORWARD -i $IFACE_WAN -o $IFACE_LAN_LOCAL -p tcp -d $IP_ADMINISTRADOR --dport 1024:65535 -s 0/0 --sport 25622 -j ACCEPT

iptables -t nat -A POSTROUTING -o $IFACE_WAN -p tcp -s $IP_GERENCIA -d 0/0 --dport 25622 -j SNAT --to $IFACE_WAN_IP
iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_WAN -p tcp -s $IP_GERENCIA --sport 1024:65535 -d 0/0 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i $IFACE_WAN -o $IFACE_LAN_LOCAL -p tcp -d $IP_GERENCIA --dport 1024:65535 -s 0/0 --sport 25622 -j ACCEPT

echo "permitiendo acceso al servidor apache (protocolo tcp, puerto cualquiera) desde lan "

iptables -t nat -A POSTROUTING -o $IFACE_LAN_DMZ -p tcp -s $IP_SOFTWARE -d $IP_DMZ_APACHE --dport 25622 -j SNAT --to $IFACE_LAN_DMZ_IP
iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_LAN_DMZ -p tcp -s $IP_SOFTWARE --sport 1024:65535 -d $IP_DMZ_APACHE --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i $IFACE_LAN_DMZ -o $IFACE_LAN_LOCAL -p tcp -d $IP_SOFTWARE --dport 1024:65535 -s $IP_DMZ_APACHE --sport 25622 -j ACCEPT

###############--HACIENDO PING--###############

echo "permitiendo icmp al servidor Master (protocolo icmp) desde lan"
iptables -t filter -A INPUT -p icmp -i $IFACE_LAN_LOCAL --icmp-type echo-request -s $NET_LAN_LOCAL -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -o $IFACE_LAN_LOCAL --icmp-type echo-reply -d $NET_LAN_LOCAL -j ACCEPT

```
echo "permitiendo icmp a Internet (protocolo icmp) desde lan"

iptables -t nat -A POSTROUTING -p icmp -o $IFACE_WAN -s $NET_LAN_LOCAL -j SNAT --to $IFACE_WAN_IP
iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_WAN -p icmp -s $NET_LAN_LOCAL -d 0/0 -j ACCEPT
iptables -t filter -A FORWARD -i $IFACE_WAN -o $IFACE_LAN_LOCAL -p icmp -d $NET_LAN_LOCAL -s 0/0 -j ACCEPT

################--SERVICIO DNS--################

echo "permitiendo dns del servidor Master (protocolo dns) a DNS Externos"
for DNS in ${SERVER_DNS_IP[@]}; do
if [ "$DNS" != "0" ]; then
iptables -t filter -A OUTPUT -o $IFACE_WAN -p udp -s $IFACE_WAN_IP --sport 1024:65535 -d $DNS --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN -p udp -d $IFACE_WAN_IP --dport 1024:65535 -s $DNS --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
fi
done

echo "permitiendo dns desde LAN (protocolo dns) a DNS Master"

iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p udp -s $NET_LAN_LOCAL --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p udp -d $NET_LAN_LOCAL --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
###############--PROXY--###############

echo "habilitando navegacion squid protocolo http (puerto 80) "
iptables -t filter -A OUTPUT -o $IFACE_WAN -p tcp -s $IFACE_WAN_IP --sport 1024:65535 -d 0/0 --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN -p tcp -d $IFACE_WAN_IP --dport 1024:65535 -s 0/0 --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando navegacion squid protocolo https (puerto 443) "
iptables -t filter -A OUTPUT -o $IFACE_WAN -p tcp -s $IFACE_WAN_IP --sport 1024:65535 -d 0/0 --dport 443 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN -p tcp -d $IFACE_WAN_IP --dport 1024:65535 -s 0/0 --sport 443 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando conexion Browsers LAN puerto 3128 al squid Master "
iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p tcp -s $NET_LAN_LOCAL --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 3128 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p tcp -d $NET_LAN_LOCAL --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 3128 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando nat para HOST de emergencia"
for NAT in ${NAT_HOST_IP[@]}; do
if [ "$NAT" != "0" ]; then
iptables -t nat -A POSTROUTING -s $NAT -d 0/0 -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
```

```
iptables -A FORWARD -s $NAT -d 0/0 -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -d $NAT -s 0/0 -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT
fi
done

#####################################################################
# Step 8: POLITICAS Y REGLAS DE FIREWALL - REGLAS BLANDAS

echo "habilitando servicios SIAF"

if [ "$HOST_SIAF_IP" != "0" ]; then
iptables -t nat -A POSTROUTING -s $HOST_SIAF_IP -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
iptables -A FORWARD -s $HOST_SIAF_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -d $HOST_SIAF_IP -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT
else

for SIAF in ${SERV_SIAF_IP[@]}; do
if [ "$SIAF" != "0" ]; then
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SIAF -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
iptables -A FORWARD -d $SIAF -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -s $SIAF -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT

fi
```

```
done
fi

echo "habilitando servicios SIS"
if [ "$HOST_SIS_IP" != "0" ]; then
iptables -t nat -A POSTROUTING -s $HOST_SIS_IP -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
iptables -A FORWARD -s $HOST_SIS_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -d $HOST_SIS_IP -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT

else
for SIS in ${SERV_SIS_IP[@]}; do
if [ "$SIS" != "0" ]; then
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SIS -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
iptables -A FORWARD -d $SIS -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -s $SIS -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT

fi
done
fi

echo "habilitando servicios RENIEC"
if [ "$HOST_RENIEC_IP" != "0" ]; then
iptables -t nat -A POSTROUTING -s $HOST_RENIEC_IP -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
```

```
iptables -A FORWARD -s $HOST_RENIEC_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -d $HOST_RENIEC_IP -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT
else
for RENIEC in ${SERV_RENIEC_IP[@]}; do
if [ "$RENIEC" != "0" ]; then

iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $RENIEC -o $IFACE_WAN -j SNAT --to $IFACE_WAN_IP
iptables -A FORWARD -d $RENIEC -i $IFACE_LAN_LOCAL -o $IFACE_WAN -j ACCEPT
iptables -A FORWARD -s $RENIEC -i $IFACE_WAN -o $IFACE_LAN_LOCAL -j ACCEPT

fi
done
fi

echo "habilitando protocolo http o https para puertos diferentes del 80 o 445."
iptables -t filter -A OUTPUT -o $IFACE_WAN -p tcp -m multiport -s $IFACE_WAN_IP -d 0/0 --dports 7777:7779,8080,2082 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN -p tcp -m multiport -d $IFACE_WAN_IP -s 0/0 --dports 7777:7779,8080,2082 -m state --state ESTABLISHED,RELATED -j ACCEPT

}

tc_clean() {
iptables -F
```

```
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
echo "0" > /proc/sys/net/ipv4/ip_forward
#tc qdisc del dev eth0 root
#tc qdisc del dev eth1 root
}
tc_stop() {
                tc_clean
        }
tc_nat()        {
tc_clean
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o $IFACE_WAN -j SNAT --to-source=$IFACE_WAN_IP
                }

tc_restart() {
        tc_stop
        sleep 1
        tc_start
                }
tc_show() {
                echo ""
                echo "eth0:"
                tc qdisc show dev eth0
```

```
                tc class show dev eth0
                tc filter show dev eth0
                echo ""
                echo "eth1:"
                tc qdisc show dev eth1
                tc class show dev eth1
                tc filter show dev eth1
                echo ""
        }
case "$1" in
start)
echo -n "Starting Intelligent Traffic: "
echo ""
tc_start
echo "started Intelligent Traffic v 1.2 : done"
echo "Write by Joseph Veliz - Conexionlinux laboratories"
echo "done"
;;
stop)
echo -n "Stopping Intelligent Traffic: "
tc_stop
echo "done"
;;
restart)
echo -n "Restarting Intelligent Traffic: "
tc_restart
echo "done"
;;
show)
tc_show
;;
```

```
nat)
tc_nat
;;
clean)
tc_clean
;;
*)
echo "Usage: /etc/init.d/AllInOne {start|stop|restart|show|nat|clean}"
;;
esac
exit 0
```

**b) AllInOne.cfg**

```
# Variables
TC=/sbin/tc
IPTABLES=/sbin/iptables

# Interfaces del servidor AllInOne
IFACE_WAN=eth0
IFACE_LAN_DMZ=eth1
IFACE_LAN_LOCAL=eth2

#IP del modem
GATEWAY_MODEM=190.117.249.130

#LAN DMZ
NET_LAN_DMZ=10.10.0.0/22
NET_LAN_LOCAL=192.168.0.0/22
```

```
#IP del servidor AllInOne
IFACE_LAN_LOCAL_IP=192.168.0.1
IFACE_LAN_DMZ_IP=10.10.0.1
IFACE_WAN_IP=190.117.249.131
IP_ADMINISTRADOR=192.168.0.4
IP_GERENCIA=192.168.0.2
IP_SOFTWARE=192.168.0.5
IP_WAN_ADMINISTRADOR=
IP_WAN_GERENCIA=
IP_DMZ_APACHE=10.10.0.3

# DNS
SERVER_DNS_IP[0]=200.123.31.50
SERVER_DNS_IP[1]=200.123.31.51
SERVER_DNS_IP[2]=8.8.8.8
SERVER_DNS_IP[3]=8.8.4.4

# IP de los servidores del SIAF
HOST_SIAF_IP=0
SERV_SIAF_IP[0]=200.4.212.77
SERV_SIAF_IP[1]=190.116.32.20
SERV_SIAF_IP[2]=190.116.32.80
SERV_SIAF_IP[3]=200.4.212.4
SERV_SIAF_IP[4]=190.116.32.18
SERV_SIAF_IP[5]=190.116.32.78
SERV_SIAF_IP[6]=200.4.212.60
SERV_SIAF_IP[7]=190.116.32.19
SERV_SIAF_IP[8]=190.116.32.79
SERV_SIAF_IP[9]=200.4.212.9
```

```
# IP de los servidores del SIS
HOST_SIS_IP=0
SERV_SIS_IP[0]=190.102.140.150

# IP de los servidores del RENIEC
HOST_RENIEC_IP=0
SERV_RENIEC_IP[0]=200.106.55.121

# IP con nat simple
NAT_HOST_IP[0]=192.168.0.20
NAT_HOST_IP[1]=0
```

# CAPÍTULO 7

# 7

# CONCLUSIONES Y RECOMENDACIONES

## 7.1. Conclusiones y recomendaciones

### 7.1.1. Conclusiones:

- ✓ Se implementó la herramienta iptables (GNU/Linux), por dos razones importantes, la primera por ser software libre y licenciado por GPL (Licencia Publica General) según Richard Stallman, cada software incluido al kernel de Linux debe de ser probado y recomendado para la organización de software libre; y la segunda porque empresas como IBM, NASA, entre otras muchas, usan GNU/Linux como sistema de seguridad, lo cual garantiza su funcionamiento si de seguridad se trata.

- ✓ Se verifico que existía vulnerabilidad de acceso en el puerto 22, configurado por defecto en el protocolo SSH, por lo tanto se plantea usar el puerto 25622 que es un puerto de UNIX no registrado formalmente. De esta manera para un hacker no será fácil rastrear el puerto de conexión.

- ✓ Se desarrolló un diseño de acuerdo a las políticas de seguridad de la empresa Conexión Linux SAC, donde solo Gerencia y Administrador de red tienen acceso al servidor master. Mientras que en los medios de accesibilidad se usó el protocolo SSH con un puerto específico para minimizar la vulnerabilidad, como también el bloqueo del protocolo ICMP hacia la WAN.

- ✓ Se logró integrar al algoritmo del firewall el bloqueo del broadcast innecesario, bloqueo de protocolo IPV6, como también reglas predeterminadas por el firewall, para garantizar el tráfico limpio. Mientras para asegurar la integridad de los paquetes de datos se

realizó dos reglas: duras y blandas, esto con la finalidad de garantizar el funcionamiento de la red.

### 7.1.2. Recomendaciones:

- ✓ Se recomienda usar llaves de seguridad en el protocolo SSH, de manera que solo usuarios con llave tengan acceso al servidor master.

- ✓ Se recomienda no modificar las reglas duras del firewall, a menos que tenga conocimiento de iptables y logre identificar a que hace referencia cada parámetro.

- ✓ Si va modifica, eliminar o agregar código en las reglas blandas, se tiene que entender que son reglas para filtrar lo necesario, por tal motivo tendríamos que tener cuidado los filtros que debemos agregar para no sufrir ciertas vulnerabilidades.

- ✓ Los host NAT, son ips de la LAN que tienen salida e ingreso libre, por lo cual solo se debe usar en caso de emergencia, y máximo debe durar 24 horas para encontrar las ip's públicas que lo requiere, en caso contrario podría vulnerar la red LAN.

- ✓ Si se desea implementar este código a otras pequeñas o medianas empresas, se recomienda poner el autor del código e implementarlo con la Licencia Publica General. Tener cuidado con los parámetros definidos en los dos archivos, si se ejecuta el firewall sin haber modificado el interfaz de su nuevo servidor, se podría cerrar ingresos y hasta dejar sin acceso a la internet.

# CAPÍTULO 8

# 8

# REFERENCIAS BIBLIOGRÁFICAS

- ✓ **DISEÑO E IMPLEMENTACIÓN DE UN ESQUEMA DE SEGURIDAD PERIMETRAL PARA REDES DE DATOS Tesis en Ingeniería De Computación México.** Conceptos Generales de Seguridad. [En línea] 08 de Diciembre del 2014. [Citado el: 22 de Abril del 2016.]. http://132.248.9.195/ptb2011/mayo/0669103/0669103_A1.pdf.

- ✓ **SEGURIDAD EN REDES.** Conceptos Generales de un Firewall. [En línea] 2009. [Citado el: 04 de Enero del 2016.]. http://cursoslibres.academica.mx/206/seguridad-en-redes/3-protecciones-y-herramientas-de-seguridad/dispositivos-firewall.

- ✓ **RED HAT ENTERPRISE LINUX 6 GUÍA DE SEGURIDAD.** IPtables. [En línea] 2011. [Citado el: 25 de Abril del 2016.]. https://access.redhat.com/documentation/es-ES/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-es-ES.pdf.

- ✓ **UNIDADES DE MEDIDA DE TRANSMISIÓN DE DATOS.** Unidades de Velocidad de Transmisión de Información. [En línea] 17 de Octubre del 2013. [Citado el: 26 de Abril del 2016.]. https://cualquiercosadetecnologia.wordpress.com/2013/10/17/unidades-de-medida-de-transmision-de-datos/.

- ✓ **CAPA 3.** Medición del ancho de banda de la red con iperf o jperf. [En línea] 08 de Enero del 2013. [Citado el: 26 de Abril del 2016]. http://capa3.es/medir-el-ancho-de-banda-de-la-red-con-iperf-o-jperf.html.

- ✓ **BLOG DE UN SYSADMIN UNIX, GNU/LINUX.** Broadcast, Multicast y Unicast. [En línea] 03 de Noviembre del 2008. [Citado el: 27 de Abril del 2016]. http://rm-rf.es/broadcast-multicast-y-unicast/.

- ✓ **PROTOCOLO TCP/IP.** Internet y TCP/IP. [En línea] 2003. [Citado el: 29 de Abril del 2016]. http://protocolotcpip.galeon.com/.

- ✓ **BLOG CESAR CABRERA.** ¿Qué es el modelo OSI? [En línea] 22 de Junio del 2015. [Citado el: 28 de Abril del 2016]. http://cesarcabrera.info/blog/%C2%BFque-es-el-modelo-osi-definicion/.

- ✓ **DIGITALIZACIÓN DE IMÁGENES.** Redes: Velocidad. [En línea] 2003. [Citado el: 28 de Abril del 2016]. https://www.library.cornell.edu/preservation/tutorial-spanish/technical/technicalD-04.html.

- ✓ **RED HAT TRAINING & CERTIFICATION RH124-RHEL7-en-1-20141208.** Red Hat System Administration I. 08 de Diciembre del 2014. [Citado el: 25 de Abril del 2016].

- ✓ **RED HAT TRAINING & CERTIFICATION RH134-RHEL7-en-1-20150420.** Red Hat System Administration II. 20 de Abril del 2015. [Citado el: 25 de Abril del 2016].

# CAPÍTULO 9

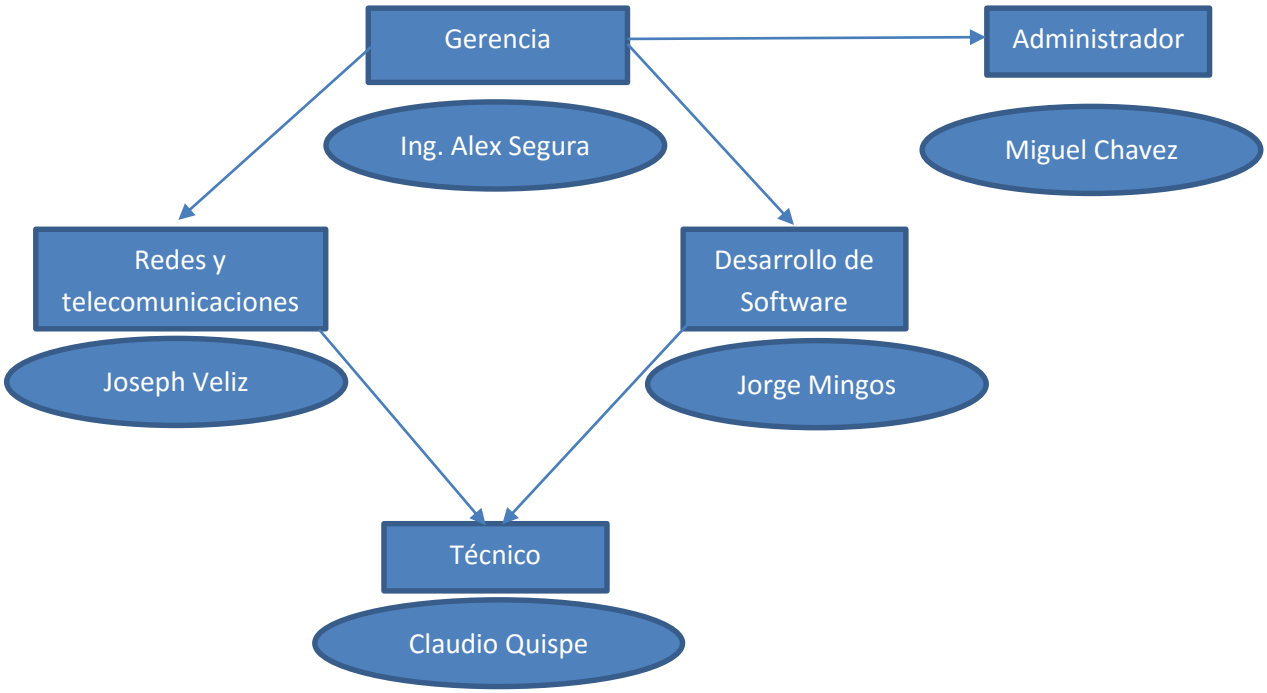
# 9

# ANEXOS

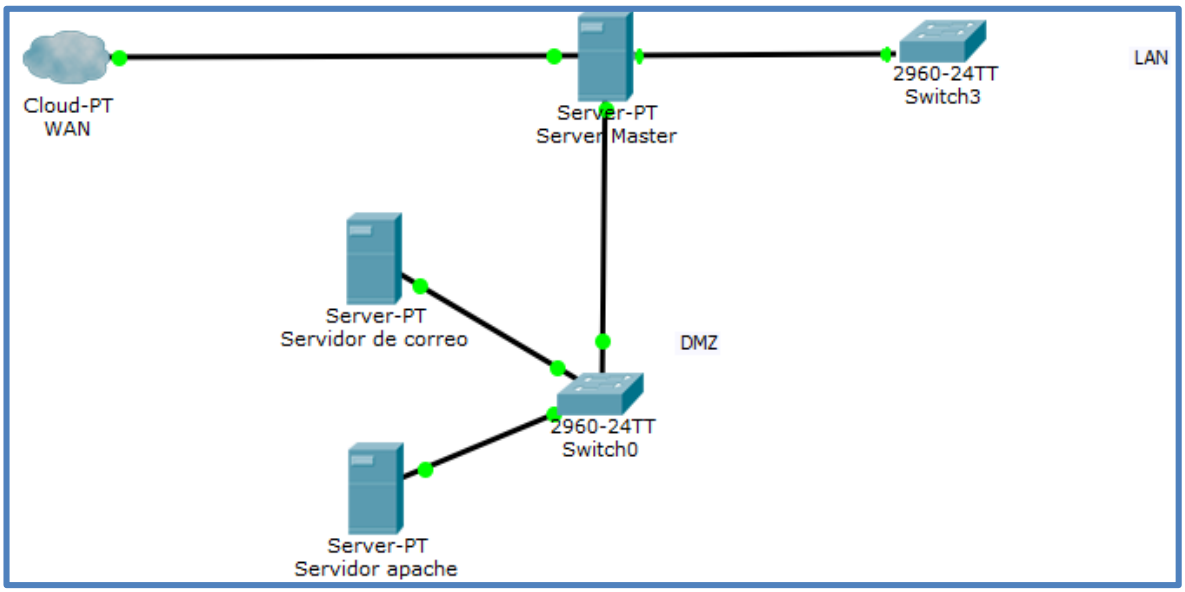
## 9.1. Matriz de consistencia.

| PLANTEAMIENTO DEL PROBLEMA | OBJETIVOS | HIPÓTESIS | VARIABLES | DIMENSIONES/INDICADORES | METODOLOGÍA |
|---|---|---|---|---|---|
| **SINTOMAS:**<br>1.- LAN vulnerable<br>2.- Detección de intrusos al servidor master<br>3.- Lentitud en la LAN<br><br>**CAUSAS:**<br>1.- No tiene un Sistema de Seguridad Perimetral.<br>2.- Seguridad no administrada<br>3.- Broadcast y puertos innecesarios habilitados.<br><br>**PRONOSTICO:**<br>Si se continúan trabajando de esta manera, lo más probable será que se pierda el control total de los servicios y el acceso total a la INTERNET<br><br>**CONTROL DEL PRONOSTICO:**<br>Aplicando un firewall con iptables, se reducirá la vulnerabilidad y lentitud de la red LAN en la empresa Conexión Linux SAC.<br><br>**FORMULACIÓN DEL PROBLEMA PRINCIPAL:**<br>¿Cómo minimizar la vulnerabilidad y lentitud de la red LAN, utilizando un firewall con iptables en la empresa Conexión Linux SAC?<br><br>**ESPECIFICOS:**<br>1.- ¿Que puertos y que ip´s son vulnerables a los DMZ y al servidor master?<br><br>2.- ¿Cuál sería la política de seguridad en el nivel de usuario y medios de acceso al servidor master?<br><br>3.- ¿De qué manera se debe optimizar el tráfico de red y asegurar su transmisión de datos? | **GENERAL:**<br><br>Implementar un firewall con iptables para reducir la vulnerabilidad y lentitud de la red LAN en la empresa Conexión Linux SAC.<br><br>**ESPECIFICOS:**<br><br>1.- Realizar un testeo para verificar las vulnerabilidades en la red LAN.<br><br>2.- Identificar e implementar los niveles de usuario, accesos al servidor master, así como también los medios de accesibilidad desde la LAN y WAN.<br><br>3.- Implementar un algoritmo que asegure la velocidad de transmisión y la integridad de los paquetes de datos. | **GENERAL:**<br><br>"La implementación de un firewall con iptables reducirá la vulnerabilidad y lentitud de la red LAN en la empresa Conexión Linux SAC."<br><br>**ESPECIFICA:**<br><br>1.- ¿Se obtendrá un filtro de acuerdo a las necesidades de la empresa?<br><br>2.- ¿Se podrá identificar que usuarios tendrán acceso a los DMZ, y que medios de accesibilidad usar?<br><br>3.- ¿Mejorará el tráfico de red y la integridad de los paquetes de datos? | **VARIABLE INDEPENDIENTE:**<br><br>FIREWALL CON IPTABLES<br><br>**VARIABLE DEPENDIENTE:**<br><br>**V**ULNERABILIDAD Y LENTITUD | **VARIABLE INDEPENDIENTE:**<br><br>FIREWALL CON IPTABLES<br><br>1) Enrutador con Capacidades de Filtrado.<br>2) Servidor de Control a Nivel Circuito.<br><br>**VARIABLE DEPENDIENTE:**<br><br>**VULNERABILIDAD** Y LENTITUD<br><br>1) Escaneo.<br>2) Obtención Acceso.<br>3) Manteniendo el Acceso.<br>4) Encubrimiento de rastros.<br><br>1) Capacidades de transporte de la red LAN.<br>2) Ancho de banda del ISP.<br>3) Velocidad y capacidad del servidor de red.<br>4) Demanda de usuarios en un momento dado.<br>5) Cantidad de tráfico que compita en la red.<br>6) Capacidades de la computadora del usuario final. | **Tipo de investigación:**<br>Proyectiva.<br><br>**Nivel de investigación:**<br>Investigación – acción.<br><br>**Diseño:**<br>Transversales descriptivos. |

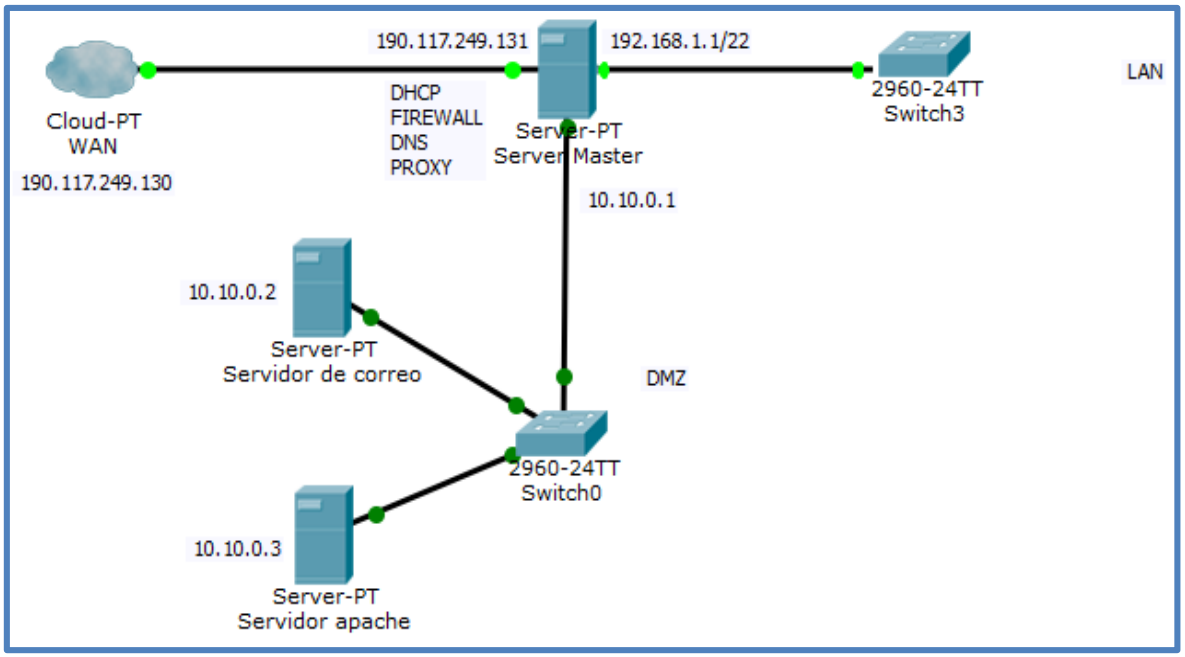
## 9.2. Estructura orgánica de la empresa.



## 9.3. Topologia física de la red




## 9.4. Topologia lógica de la red



### 9.5. Tiempo y horario

| Procedimientos a seguir: | 2016 | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Marzo | | | | | | | | | Abril | | | | | | | | | |
| | 21 | 22 | 23 | 24 | 25 | 28 | 29 | 30 | 31 | 1 | 4 | 5 | 6 | 7 | 8 | 11 | 12 | 13 | 14 |
| Recolección de datos. | X | X | X | X | | | | | | | | | | | | | | | |
| Instalación del S.O. Centos 7.0 | | | | | X | | | | | | | | | | | | | | |
| Diseño del firewall perimetral. | | | | | | X | X | X | X | X | | | | | | | | | |
| Implementación del servidor master. | | | | | | | | | | | X | X | X | X | X | | | | |
| Comprobar accesos y vulnerabilidades al servidor Master. | | | | | | | | | | | | | | | | X | X | X | X |
| Conectar el servidor Master al DMZ. | | | | | | | | | | | | | | X | X | X | X | X | X |

| Lunes | Martes | Miércoles | Jueves | Viernes |
|---|---|---|---|---|
| 9:00 – 12:00 pm | 9:00 – 12:00 pm | 9:00 – 12:00 pm | 9:00 – 12:00 pm | 9:00 – 12:00 pm |
| 2:00 – 5:00 pm | 2:00 – 5:00 pm | 2:00 – 5:00 pm | 2:00 – 5:00 pm | 2:00 – 5:00 pm |

## 9.6. Instrumentos de medición

Los instrumentos de medición (Wireshark, bwm-ng, iperf, arp-a y nmap) que se usaron para realizar la tesis cuentan con la Licencia Pública General (GPL), con excepción a iperf que cuenta con la licencia BSD.

Todas estas herramientas a diferencia del shareware y freeware, tienen una licencia que les respalda, y garantizan el funcionamiento de sistema. Empresas como IBM, NASA, entre muchas otras empresas en el mundo usan GNU/Linux en seguridad y sus herramientas que traen con ellas para el funcionamiento. Para que un software libre sea licenciado por la comunidad, tienen que haber pasado por pruebas realizadas por expertos de la comunidad de software libre.

Toda la documentación de la Licencia Pública General lo puedes seguir en el siguiente link: https://www.gnu.org/licenses/licenses.es.html.

Toda la documentación de la licencia BSD lo puedes seguir en el siguiente link: http://blog.desdelinux.net/hablemos-de-la-licencia-bsd/.

A continuación describiremos cada uno de estas herramientas que fueron utilizadas:

a) **Wireshark:** Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red. La utilización de esta herramienta puede parecer de gran complejidad en un principio, pero es de gran utilidad una vez conocida su interfaz y su forma de operar. Existen diferentes usos para los cuales puede aplicarse Wireshark. Dentro del análisis dinámico de códigos maliciosos se la utiliza para detectar conexiones ocultas del propio malware con direcciones remotas para obtener otros archivos, para reportarse a un panel de control en caso de una botnet, entre otras variantes.

En primera instancia, para realizar un análisis dinámico de un código malicioso se procede a infectar un sistema en un entorno controlado. Por lo general, se recurre a una máquina virtual. De esta forma, es posible ejecutar Wireshark y seleccionar la interfaz de red de la máquina virtual para comenzar a capturar los paquetes de red.

**b) Bwm-ng:** Una herramienta que nos reportará, mediante el terminal, información del ancho de banda consumido a través de las distintas interfaces de red o, incluso, de los discos. Lo tenemos tanto para Linux/UNIX como para Windows, aunque para este último S.O. no están activas todas las opciones.

**c) Iperf:** Iperf es una herramienta que nos permite realizar un test de velocidad de red.

Para ser más exactos, iperf permite medir el ancho de banda entre dos hosts usándolo en modo cliente-servidor y con tcp o udp como protocolos de conexión.

**d) Nmap:** Explora redes, determina el nombre del nodo y escanea puertos. Se debe usar en modo root.

## 9.7. Versión 1 del AllInOne

### a) AllInOne.sh

```
tc_start() {
tc_clean

# Step 1: Detenemos servicio
echo "     DETENEMOS EL SERVICIO DE FIREWALL ..."
/sbin/systemctl restart iptables

# Step 2:  FLUSH de reglas
echo "     COMENZAMOS BORRANDO TODAS LAS REGLAS ACTUALES"
             iptables -F
# Flush all chains
             iptables -X
             iptables -t nat -F
             iptables -t nat -X
             iptables -t mangle -F
             iptables -t mangle -X

# Step 3: Flush the user chain.. if it exists
             if [ "`iptables -L | grep drop-and-log-it`" ]; then
                   iptables -F drop-and-log-it
             fi

 # Step 4:Delete all User-specified chains
             iptables -X

 # Step 5: Reset all IPTABLES counters
             iptables -Z
```

```
# Step 6:  Cargamos modulos al KERNEL
echo "     CARGAMOS LOS MODULOS PRINCIPALES DEL KERNEL"
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_nat_irc
/sbin/modprobe ip_conntrack_irc
/sbin/modprobe ip_conntrack_ftp # Needed for FTP (specifically, to allow incoming ftp-data connections)
/sbin/modprobe ip_nat_ftp
/sbin/modprobe iptable_nat

# Step 7:  Aplicando reglas elementales
echo "APLICANDO REGLAS DE SEGURIDAD ELEMENTALES "
# No respondemos a los broadcast.
echo "          No reponder broadcast"
/bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Step 8: Disable IPV6 problems with squid reported and this correct.
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
echo 1 > /proc/sys/net/ipv6/conf/default/disable_ipv6

# Otros
echo "          Otros"
/bin/echo 1 > /proc/sys/net/ipv4/tcp_syncookies
/bin/echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
# El Ip Forwarding
echo "          IP Forwarding enable"
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
# Para usuarios con IP dinamica
```

```
echo "          Para usuarios con IP dinamica"
/bin/echo 1 > /proc/sys/net/ipv4/ip_dynaddr

# Proteccion en contra de errores de mensajes
echo "          Proteccion contra malos errores de mensajes"
/bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

echo "          DENEGANDO ACCESO A LAS CADENAS IPTABLES TODAS INTERFACES eth0 eth1 etc ..."
iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables --policy OUTPUT DROP

echo "configurando INPUT y OUTPUT"
echo "dando acceso total al loopback"
# todo lo que entra y sale desde el loopback se acepta
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde lan "
iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.2 --sport 1024:65535 -d 192.168.0.1 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.2 --dport 1024:65535 -s 192.168.0.1 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 192.168.0.1 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 192.168.0.1 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo acceso al servidor DMZ(protocolo tcp, puerto 25622) desde Servidor Master "
iptables -t filter -A OUTPUT -o eth2 -p tcp -d 10.10.0.0/22 --dport 1024:65535 -s 10.10.0.1 --sport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth2 -p tcp -s 10.10.0.0/22 --sport 1024:65535 -d 10.10.0.1 --dport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde wan "
iptables -t filter -A INPUT -i eth0 -p tcp -s 190.236.178.87 --sport 1024:65535 -d 190.12.87.106 --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -p tcp -d 190.236.178.87 --dport 1024:65535 -s 190.12.87.106 --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo acceso al DMZ (protocolo tcp, puerto 25622) desde lan "
iptables -t filter -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.0.5 --sport 1024:65535 -d 10.10.0.3 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.0.5 --dport 1024:65535 -s 10.10.0.3 --sport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 10.10.0.2 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 10.10.0.2 --sport 25622 -j ACCEPT

```
echo "permitiendo acceso a otro servidor SSH (protocolo tcp, puerto 25622) desde lan "
iptables -t nat -A POSTROUTING -o eth0 -p tcp -s 192.168.0.4 -d 200.114.54.33 --dport 25622 -j SNAT --to 190.12.87.106
iptables -t filter -A FORWARD -i eth1 -o eth0 -p tcp -s 192.168.0.4 --sport 1024:65535 -d 200.114.54.33 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i eth0 -o eth1 -p tcp -d 192.168.0.4 --dport 1024:65535 -s 200.114.54.33 --sport 25622 -j ACCEPT

echo "permitiendo icmp al servidor Master (protocolo icmp) desde lan"
iptables -t filter -A INPUT -p icmp -i eth1 --icmp-type echo-request -s 192.168.0.0/22 -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -o eth1 --icmp-type echo-reply -d 192.168.0.0/22 -j ACCEPT

echo "permitiendo icmp a Internet (protocolo icmp) desde lan"
iptables -t nat -A POSTROUTING -p icmp -o eth0 -s 192.168.0.0/22 -j SNAT --to 190.12.87.106
iptables -t filter -A FORWARD -i eth1 -o eth0 -p icmp -s 192.168.0.0/22 -d 0/0 -j ACCEPT
iptables -t filter -A FORWARD -i eth0 -o eth1 -p icmp -d 192.168.0.0/22 -s 0/0 -j ACCEPT

echo "permitiendo dns del servidor Master (protocolo dns) a DNS Externos"
iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 8.8.8.8 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 8.8.8.8 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 8.8.4.4 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 8.8.4.4 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 190.12.72.226 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 190.12.72.226 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth0 -p udp -s 190.12.87.106 --sport 1024:65535 -d 190.12.72.227 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p udp -d 190.12.87.106 --dport 1024:65535 -s 190.12.72.227 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "permitiendo dns desde LAN (protocolo dns) a DNS Master"
iptables -t filter -A INPUT -i eth1 -p udp -s 192.168.0.0/22 --sport 1024:65535 -d 192.168.0.1 --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p udp -d 192.168.0.0/22 --dport 1024:65535 -s 192.168.0.1 --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando navegacion squid protocolo http (puerto 80) "
iptables -t filter -A OUTPUT -o eth0 -p tcp -s 190.12.87.106 --sport 1024:65535 -d 0/0 --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -t filter -A INPUT -i eth0 -p tcp -d 190.12.87.106 --dport 1024:65535 -s 0/0 --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando navegacion squid protocolo https (puerto 443) "
iptables -t filter -A OUTPUT -o eth0 -p tcp -s 190.12.87.106 --sport 1024:65535 -d 0/0 --dport 443 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p tcp -d 190.12.87.106 --dport 1024:65535 -s 0/0 --sport 443 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando conexion Browsers LAN puerto 3128 al squid Master "
iptables -t filter -A INPUT -i eth1 -p tcp -s 192.168.0.0/22 --sport 1024:65535 -d 192.168.0.1 --dport 3128 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp -d 192.168.0.0/22 --dport 1024:65535 -s 192.168.0.1 --sport 3128 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -s $NAT -d 0/0 -o eth0 -j SNAT --to 190.12.87.106
iptables -A FORWARD -s $NAT -d 0/0 -i eth1 -o eth0 -j ACCEPT
iptables -A FORWARD -d $NAT -s 0/0 -i eth0 -o eth1 -j ACCEPT
```

## 9.8. Versión 2 del AllInOne

### a) AllInOne.sh

```
##################### Variables ############################
TC=/sbin/tc
IPTABLES=/sbin/iptables
# Interfaces del servidor AllInOne
IFACE_WAN_SATEL=0
```

IFACE_LAN_LOCAL=0
IFACE_WIFI_LOCAL=0
IFACE_WAN_GUEST=0
# IP del modem satelital
GATEWAY_SATEL=0
# RED LAN AND WLAN
NET_LAN_LOCAL=0
NET_WIFI_LOCAL=0
NET_LAN_GUEST=0
# IP del servidor AllInOne
IFACE_WAN_GUEST_IP=0
IFACE_LAN_LOCAL_IP=0
IFACE_WIFI_LOCAL_IP=0
IFACE_WAN_SATEL_IP=0
#DNS
SERVER_DNS_IP[0]=0
SERVER_DNS_IP[1]=0
SERVER_DNS_IP[2]=0
SERVER_DNS_IP[3]=0
# IP de los servidores del SIAF
HOST_SIAF_IP=0
SERV_SIAF_IP[0]=0
SERV_SIAF_IP[1]=0
SERV_SIAF_IP[2]=0
SERV_SIAF_IP[3]=0
SERV_SIAF_IP[4]=0
SERV_SIAF_IP[5]=0
SERV_SIAF_IP[6]=0
SERV_SIAF_IP[7]=0
SERV_SIAF_IP[8]=0
SERV_SIAF_IP[9]=0

```
# IP con nat simple
NAT_HOST_IP[0]=0
NAT_HOST_IP[1]=0
#####################################################################
# Reload Varibles from file
if [ -f /root/AllInOne.internetsalma.v6.02-r1.cfg ];then
. /root/AllInOne.internetsalma.v6.02-r1.cfg
fi
tc_start() {
tc_clean

# Step 1: Detenemos servicio
echo " DETENEMOS EL SERVICIO DE FIREWALL ..."
/sbin/service iptables restart

# Step 2: FLUSH de reglas
echo " COMENZAMOS BORRANDO TODAS LAS REGLAS ACTUALES ..."
iptables -F # Flush all chains
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
# Step 2: Flush the user chain.. if it exists
if [ "`iptables -L | grep drop-and-log-it`" ]; then
iptables -F drop-and-log-it
fi
# Step 2:Delete all User-specified chains
iptables –X

# Step 3: Reset all IPTABLES counters
```

```
iptables –Z

# Step 4: Cargamos modulos al KERNEL
echo " CARGAMOS LOS MODULOS PRINCIPALES DEL KERNEL ..."
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_nat_irc
/sbin/modprobe ip_conntrack_irc
/sbin/modprobe ip_conntrack_ftp # Needed for FTP (specifically, to allow incoming ftp-data connections)
/sbin/modprobe ip_nat_ftp
/sbin/modprobe iptable_nat

# Step 5: Aplicando reglas elementales
echo " APLICANDO REGLAS DE SEGURIDAD ELEMENTALES ..."
# Quitamos los pings.
# echo " Quitando ping"
# /bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# No respondemos a los broadcast.
echo " No reponder broadcast"
/bin/echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Step 6: Disable IPV6 problems with squid reported and this correct.
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
echo 1 > /proc/sys/net/ipv6/conf/default/disable_ipv6

# Otros
echo " Otros"
/bin/echo 1 > /proc/sys/net/ipv4/tcp_syncookies
/bin/echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
```

```
# El Ip Forwarding
echo " IP Forwarding enable"
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
# Para usuarios con IP dinamica
echo " Para usuarios con IP dinamica"
/bin/echo 1 > /proc/sys/net/ipv4/ip_dynaddr
# Proteccion en contra de errores de mensajes
echo " Proteccion contra malos errores de mensajes"
/bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
#Code Listing 2.7: Enable reverse path filtering

#####################################################################
# Step 7: POLITICAS Y REGLAS DE FIREWALL - REGLAS DURAS
echo " DENEGANDO ACCESO A LAS CADENAS IPTABLES TODAS INTERFACES eth0 eth1 etc ..."
iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables --policy OUTPUT DROP
echo "configurando INPUT y OUTPUT"
echo "dando acceso total al loopback"
# todo lo que entra y sale desde el loopback se acepta
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

###############--SSH--###############
echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde lan "
#iptables -t filter -A INPUT -p tcp --dport 25622 -i eth1 -j LOG --log-level 4 --log-prefix "Acceso SSH "
```

```
iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p tcp -s $NET_LAN_LOCAL --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p tcp -d $NET_LAN_LOCAL --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT
echo "permitiendo acceso al servidor SSH (protocolo tcp, puerto 25622) desde wan "
#iptables -t filter -A INPUT -p tcp --dport 25622 -i eth0 -j LOG --log-level 4 --log-prefix "Acceso SSH "
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p tcp -s 0/0 --sport 1024:65535 -d $IFACE_WAN_SATEL_IP --dport 25622 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p tcp -d 0/0 --dport 1024:65535 -s $IFACE_WAN_SATEL_IP --sport 25622 -m state --state ESTABLISHED,RELATED -j ACCEPT
echo "permitiendo acceso a otro servidor SSH (protocolo tcp, puerto 25622) desde lan "
iptables -t nat -A POSTROUTING -o $IFACE_WAN_SATEL -p tcp -s $NET_LAN_LOCAL -d 0/0 --dport 25622 -j SNAT --to $IFACE_WAN_SATEL_IP

iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -p tcp -s $NET_LAN_LOCAL --sport 1024:65535 -d 0/0 --dport 25622 -j ACCEPT
iptables -t filter -A FORWARD -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -p tcp -d $NET_LAN_LOCAL --dport 1024:65535 -s 0/0 --sport 25622 -j ACCEPT

###############--HACIENDO PING--###############
echo "permitiendo icmp al servidor Master (protocolo icmp) desde lan"
```

```
iptables -t filter -A INPUT -p icmp -i $IFACE_LAN_LOCAL --icmp-type echo-request -s $NET_LAN_LOCAL -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -o $IFACE_LAN_LOCAL --icmp-type echo-reply -d $NET_LAN_LOCAL -j ACCEPT
iptables -t filter -A INPUT -p icmp -i $IFACE_WIFI_LOCAL --icmp-type echo-request -s $NET_WIFI_LOCAL -j ACCEPT
iptables -t filter -A OUTPUT -p icmp -o $IFACE_WIFI_LOCAL --icmp-type echo-reply -d $NET_WIFI_LOCAL -j ACCEPT
echo "permitiendo icmp a Internet (protocolo icmp) desde lan"
iptables -t nat -A POSTROUTING -p icmp -o $IFACE_WAN_SATEL -s $NET_LAN_LOCAL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t filter -A FORWARD -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -p icmp -s $NET_LAN_LOCAL -d 0/0 -j ACCEPT
iptables -t filter -A FORWARD -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -p icmp -d $NET_LAN_LOCAL -s 0/0 -j ACCEPT

################--SERVICIO DNS--################
echo "permitiendo dns del servidor Master (protocolo dns) a DNS Externos"
for DNS in ${SERVER_DNS_IP[@]}; do
if [ "$DNS" != "0" ]; then
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p udp -s $IFACE_WAN_SATEL_IP --sport 1024:65535 -d $DNS --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p udp -d $IFACE_WAN_SATEL_IP --dport 1024:65535 -s $DNS --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
fi
done
echo "permitiendo dns desde LAN (protocolo dns) a DNS Master"
```

```
iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p udp -s $NET_LAN_LOCAL --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p udp -d $NET_LAN_LOCAL --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WIFI_LOCAL -p udp -s $NET_WIFI_LOCAL --sport 1024:65535 -d $IFACE_WIFI_LOCAL_IP --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_WIFI_LOCAL -p udp -d $NET_WIFI_LOCAL --dport 1024:65535 -s $IFACE_WIFI_LOCAL_IP --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

###############--PROXY--###############
echo "habilitando navegacion squid protocolo http (puerto 80) "
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p tcp -s $IFACE_WAN_SATEL_IP --sport 1024:65535 -d 0/0 --dport 80 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p tcp -d $IFACE_WAN_SATEL_IP --dport 1024:65535 -s 0/0 --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT
echo "habilitando navegacion squid protocolo https (puerto 443) "
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p tcp -s $IFACE_WAN_SATEL_IP --sport 1024:65535 -d 0/0 --dport 443 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p tcp -d $IFACE_WAN_SATEL_IP --dport 1024:65535 -s 0/0 --sport 443 -m state --state ESTABLISHED,RELATED -j ACCEPT
echo "habilitando conexion Browsers LAN puerto 3128 al squid Master "
```

```
iptables -t filter -A INPUT -i $IFACE_LAN_LOCAL -p tcp -s $NET_LAN_LOCAL --sport 1024:65535 -d $IFACE_LAN_LOCAL_IP --dport 3128 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_LAN_LOCAL -p tcp -d $NET_LAN_LOCAL --dport 1024:65535 -s $IFACE_LAN_LOCAL_IP --sport 3128 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WIFI_LOCAL -p tcp -s $NET_WIFI_LOCAL --sport 1024:65535 -d $IFACE_WIFI_LOCAL_IP --dport 3128 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -o $IFACE_WIFI_LOCAL -p tcp -d $NET_WIFI_LOCAL --dport 1024:65535 -s $IFACE_WIFI_LOCAL_IP --sport 3128 -m state --state ESTABLISHED,RELATED -j ACCEPT

echo "habilitando conexion Browsers LAN puerto 3128 al squid Master "
iptables -t nat -A POSTROUTING -s NAT_HOST_IP[0] -d 0/0 -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -s NAT_HOST_IP[0] -d 0/0 -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -d NAT_HOST_IP[0] -s 0/0 -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

#################################################################
# Step 8: POLITICAS Y REGLAS DE FIREWALL -  REGLAS BLANDAS

echo "habilitando servicios SIAF"
if [ "$SERV_SIAF_IP" != "0" ]; then
iptables -t nat -A POSTROUTING -s $SERV_SIS_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -s $SERV_SIS_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
```

```
iptables -A FORWARD -d $SERV_SIS_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

else
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF01_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF02_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF03_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF04_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF05_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF06_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF07_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF08_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF09_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
```

```
iptables -t nat -A POSTROUTING -s $NET_LAN_LOCAL -d $SERV_SIAF10_IP -o $IFACE_WAN_SATEL -j SNAT --to $IFACE_WAN_SATEL_IP
iptables -A FORWARD -d $SERV_SIAF01_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF01_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF02_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF02_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF03_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF03_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF04_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF04_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF05_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF05_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF06_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF06_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF07_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF07_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
```

```
iptables -A FORWARD -d $SERV_SIAF08_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF08_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF09_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF09_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT
iptables -A FORWARD -d $SERV_SIAF10_IP -i $IFACE_LAN_LOCAL -o $IFACE_WAN_SATEL -j ACCEPT
iptables -A FORWARD -s $SERV_SIAF10_IP -i $IFACE_WAN_SATEL -o $IFACE_LAN_LOCAL -j ACCEPT

echo "habilitando protocolo http o https para puertos diferentes del 80 o 445 "
iptables -t filter -A OUTPUT -o $IFACE_WAN_SATEL -p tcp -m multiport -s $IFACE_WAN_SATEL_IP -d 0/0 --dports 7777:7779,8080,2082 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -i $IFACE_WAN_SATEL -p tcp -m multiport -d $IFACE_WAN_SATEL_IP -s 0/0 --dports 7777:7779,8080,2082 -m state --state ESTABLISHED,RELATED -j ACCEPT
}
```

**b) AllInOne.cfg**

```
# Variables
TC=/sbin/tc
IPTABLES=/sbin/iptables
# Interfaces del servidor AllInOne
IFACE_WAN_SATEL=eth0
IFACE_LAN_LOCAL=eth1
```

IFACE_WAN_GUEST=0
# IP del modem satelital
GATEWAY_SATEL=190.117.249.130
# LAN WLAN
NET_LAN_LOCAL=10.10.0.0/22
NET_LAN_GUEST=0
# IP del servidor AllInOne
IFACE_WAN_GUEST_IP=0
IFACE_LAN_LOCAL_IP=10.10.0.1
IFACE_WAN_SATEL_IP=192.168.10.2
# DNS
SERVER_DNS_IP[0]=200.123.31.50
SERVER_DNS_IP[1]=200.123.31.51
SERVER_DNS_IP[2]=8.8.8.8
SERVER_DNS_IP[3]=8.8.4.4
# IP de los servidores del SIAF
HOST_SIAF_IP=0
SERV_SIAF_IP[0]=200.4.212.77
SERV_SIAF_IP[1]=190.116.32.20
SERV_SIAF_IP[2]=190.116.32.80
SERV_SIAF_IP[3]=200.4.212.4
SERV_SIAF_IP[4]=190.116.32.18
SERV_SIAF_IP[5]=190.116.32.78
SERV_SIAF_IP[6]=200.4.212.60
SERV_SIAF_IP[7]=190.116.32.19
SERV_SIAF_IP[8]=190.116.32.79
SERV_SIAF_IP[9]=200.4.212.9
# IP con nat simple
NAT_HOST_IP[0]=192.168.0.20
NAT_HOST_IP[1]=0