UVAQ
UNIVERSIDAD VASCO DE QUIROGA
Repositorio Digital Institucional

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Firewall de alta disponibilidad

Autor: Francisco Javier Ferreyra López

Tesis presentada para obtener el título de:
Lic. En Sistemas Computarizados [Sic]

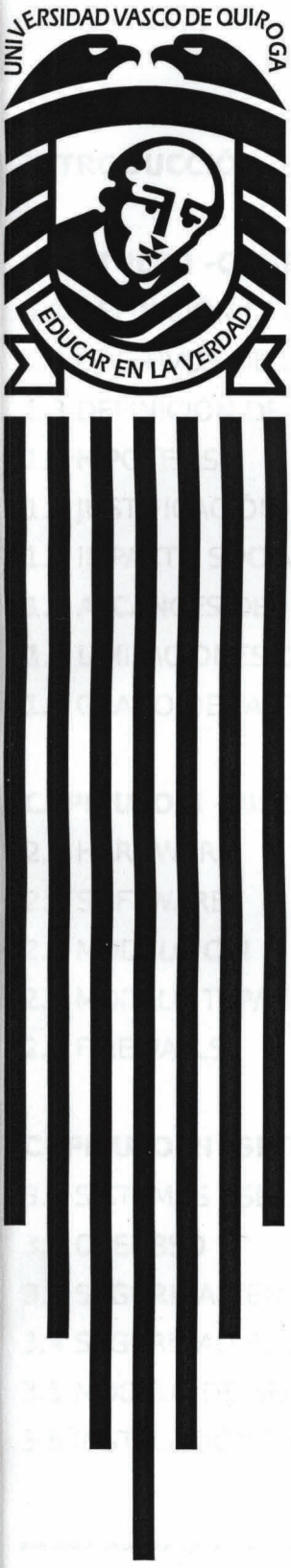
Nombre del asesor:
Gabriel Nava Vázquez

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.

Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.



UVAQ M.R.

UNIVERSIDAD VASCO DE QUIROGA

ESCUELA DE LICENCIATURA EN SISTEMAS COMPUTARIZADOS

N° DE ACUERDO:952006 CLAVE:16PSU0049F

Firewalls de Alta Disponibilidad

TESIS QUE PARA OBTENER EL GRADO DE

Licenciado en Sistemas Computarizados

PRESENTA:

Francisco Javier Ferreyra López

ASESOR:

M.C. Gabriel Nava Vázquez

MORELIA, MICH. JULIO 2007

INDICE

	PÁG.
INTRODUCCIÓN	1
CAPITULO I -GENERALIDADES-	
1.1 DEFINICIÓN DEL PROBLEMA	3
1.2 OBJETIVOS DE LA TESIS	3
1.3 DEFINICIÓN DE VARIABLES	4
1.4 HIPOTESIS	5
1.5 JUSTIFICACIÓN	6
1.6 IMPACTO SOCIAL	8
1.7 ALCANCES DEL PROYECTO	8
1.8 LIMITACIONES DEL PROYECTO	9
1.9 GRADO DE FACTIBILIDAD	9
CAPITULO II -DISEÑO DE RED-	
2.1 HARDWARE	10
2.2 SOFTWARE	12
2.3 MODELO OSI	12
2.4 MODELO TCP/IP	17
2.5 FIREWALLS	20
CAPITULO III -SISTEMA OPERATIVO OPENBSD-	
3.1 SISTEMAS BSD	26
3.2 OPENBSD	28
3.3 SEGURIDAD EN SISTEMA OPERATIVO	30
3.4 SEGURIDAD PROACTIVA DENTRO DEL PROYECTO OPENBSD	31
3.5 MODELO DE SEGURIDAD DE OPENBSD	32
3.6 INSTALACIÓN DE OPENBSD	32

	PÁG
CAPITULO IV -ALTA DISPONIBILIDAD-	
4.1 DEFINICIONES	48
4.2 CONCEPTOS	49
4.3 PROTOCOLOS	52
4.4 ESCENARIO	55
4.5 IMPLEMENTACIÓN	56
CAPITULO V -PACKET FILTER-	
5.1 INTRODUCCIÓN	66
5.2 SERVICIOS	69
5.3 EXPLICACIÓN DE LAS REGLAS	69
5.4 SCRIPT FINAL	71
CAPITULO VI -CONCLUSIONES Y RECOMENDACIONES-	
6.1 SEGURIDAD EN LOS SERVICIOS	78
6.2 PLAN DE CONTINGENCIA Y RECUPERACIÓN	80
6.3 CONCLUSIONES	83
GLOSARIO DE TERMINOS	85
REFERENCIAS	101

Las sociedades avanzadas de principios de este siglo son denominadas con frecuencia “sociedades de la información“, pues el volumen de datos que es procesado, almacenado y transmitido es inconmensurablemente mayor que en cualquier época pretérita.

Además, no sólo el volumen, sino la importancia de esta información para el desarrollo económico y social, no tienen ninguna comparación con la que tuvo en el pasado. De hecho, en la actualidad, las organizaciones consideran que la información es un bien más de su activo y, en muchos casos, prioritario sobre los restantes.

Así mismo todo lo que conlleva a almacenar, procesar y enviar esa información, como son medios de comunicación, las respectivas aplicaciones que se utilizan y los medios físicos de los que se valen comúnmente llamadas tecnologías de información, adquieren la misma relevancia que la información misma ya que no existiría una sin las demás.

No hay que dejar de mencionar el inminente crecimiento de Internet y todo el impacto social que esta causando en nuestra vida diaria, paulatinamente está cambiando la forma de vida de la sociedad, todo es más práctico y las distancias no parecen existir, haciendo nuestra vida mucho más sencilla y práctica quitando de nuestro desarrollo la limitante económica que significaba trasladarte a otros lugares. En nuestros días es muy sencillo desde comprar un articulo semi nuevo a algún desconocido en Argentina, hasta estudiar una carrera profesional a distancia en Instituciones educativas de México o Estados Unidos.

Dentro de esta gama de servicios y productos que nos ofertan, para cada una de las instituciones que lo hacen es muy importante el manejo de la información y aunado a todas las posibilidades que nos oferta Internet, está la seguridad de la información y el peligro latente que sufren nuestras computadoras a ser violada su seguridad, ver comprometida nuestra información y documentos. Dese nuestra pc de casa hasta la información confidencial de los clientes de una Institución Bancaria, es aquí donde la Seguridad Informatica tiene su campo de acción para ayudarnos a protegernos de ataques a la información, perpetrados por terceras personas ó sistemas.

Existe un acuerdo y conciencia general sobre la importancia de la Seguridad de los Sistemas de Información (SSI). La SSI está relacionada con la disponibilidad, confidencialidad e integridad de la información *(Figura 1)* tratada por las computadoras y las redes de comunicación. Se usan comúnmente otros términos que en esencia tienen el mismo significado, tales como seguridad de la información, seguridad de las computadoras, seguridad de datos o protección de la información, pero se orientan a la Seguridad de los Sistemas de Información, como un todo, HARDWRE, SOFTWARE y USUARIOS.

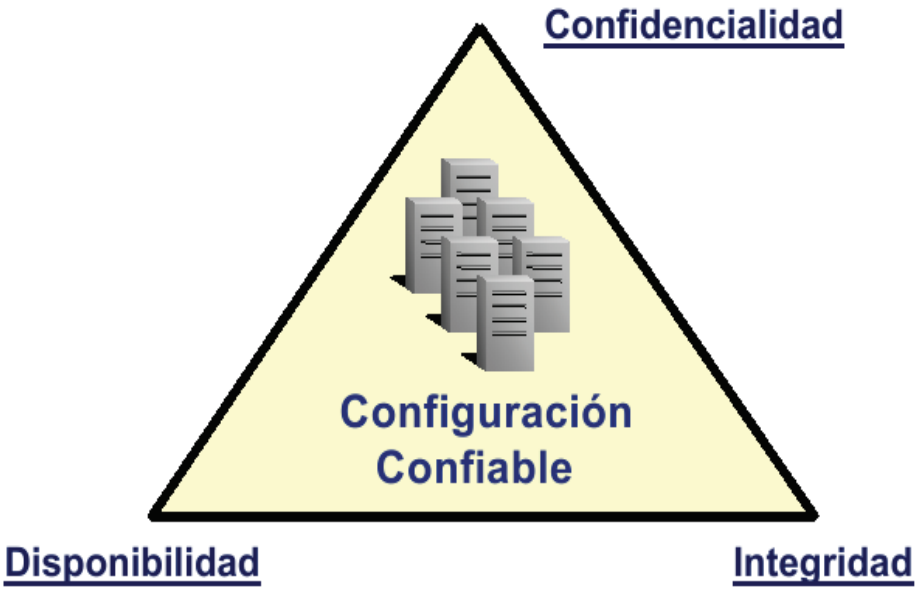


Figura 1

Dentro de la amplio horizonte que abarca el tema de la Seguridad informática, el tema de los cortafuegos (firewall) toma una importancia relevante ya que es el recurso de seguridad que nos protege de ataques cibernéticos para ser aplicado en todo tipo de entorno, más en nuestros días que no sabemos aveces quién, mucho menos cómo nos pueden tratar de invadir o robar desde INTERNET a nuestra casa o dentro de las oficinas donde tenemos nuestros servidores, así que es fundamental contar con este tipo de protección y saber como manejarla, de manera menos relevante pero real están los virus que pueden destruir nuestra información, para lo cual ya existen una diversa cantidad de anti-virus en el mercado.

1.1 Definición del problema.

La demanda de servicios en nuestros días aunado a la ampliación de mercados y la posibilidad de incursionar en ellos, demanda a las empresas públicas y privadas que quieran ofertar servicios o productos dentro de Internet a preguntarse lo siguiente:

- ¿Que tan seguro será el manejo del intercambio de información con mis clientes o proveedores?
- ¿Como puedo asegurar que las personas que accedan a esa información sean las correctas?.

Esto no es fácil de responder y peor aún cuando no depende de los administradores ya que no conocen del campo y no cuentan con un profesional del área que los asesore, por lo que en la mayoría de los casos lleva a los altos mandos de las instituciones a contratar servicios que no necesitan o un mal servicio que inevitablemente culmina con un dolor de cabeza cuando han sido víctimas de un fraude o les han robado información interna.

En nuestro caso en particular no vamos a referirnos a ningún tipo servicio en particular ya que la arquitectura propuesta a continuación pretende dar un enfoque en general de cómo se puede construir un perímetro de alta disponibilidad para la protección de nuestros servidores, aunado a esto ejemplificaremos el préstamo de servicios mediante un servidor web, un servicio de **security shell** y un servidor de correo, esto nos permitirá ejemplificar el como se podrían proteger dichos servicios en cuanto a integridad y confidencialidad de los mismos.

De esta manera estaremos cubriendo los tres tópicos que cubre la seguridad informática en general, no dejando de lado el mencionar que dentro de estos 3 grandes campos hay muchas particularidades que no podemos abarcar en este documento, por ser estas muy extensas.

1.2 Objetivos de la tesis.

a. Objetivo General de la tesis.

Informar, compartir y aprender creando una arquitectura perimetral robusta y segura de alta disponibilidad para los servicios que se ofrecen en cualquier empresa o institución basándose en

estándares y software de código abierto "**OpenSource**", tomando como base la disponibilidad, Integridad y confidencialidad de los mismos.

b. Objetivos Específicos de la tesis.

- Diseñar una red orientada a la alta disponibilidad pensando en aplicaciones futuras.
- Difundir el uso y todo lo que conlleva el sistema operativo de libre distribución **OpenBSD** para tareas de carácter específico y crítico, como lo es un perímetro de seguridad en nuestra red.
- Explicar y difundir el termino Alta Disponibilidad en informática.
- Crear una protección básica funcional para nuestros servicios por medio de la implementación de un firewall.
- Mencionar los servicios más comunes y los puntos esenciales que se deben de revisar desde el punto de vista de la seguridad.
- Hacer conciencia en la comunidad informática de la relevancia que debe de tener para nosotros los temas de seguridad informática que cubren desde nuestra PC en el hogar hasta los servidores que administramos en nuestro trabajo.

1.3 Definición de variables.

Teniendo en cuenta lo antes comentado, tenemos que cumplir con las siguientes variables para la resolución del problema.

- Servicios a Implementar.
- Equipo de cómputo destinado a proporcionar el servicio.

- Mejoras en la implementación de alta disponibilidad mediante la actualización de funcionalidades en los protocolos que sean necesarios.

1.4 Hipótesis.

La seguridad absoluta no existe, pero no por esto nos vamos a quedar de brazos cruzados y ver como los incidentes empiezan a suceder a nuestros colegas y esperar pacientemente a que no nos ocurra a nosotros.

Una solución en seguridad mal diseñada o mal implantada puede darnos peores resultados de los que nos pudiera haber causado algún ataque informático.

En una solución de este tipo se tiene que pensar desde el tipo de cableado que se tiene que implementar con su respectiva topología hasta el tipo de aplicaciones que se deben de implementar para ofrecer nuestros servicios en base a las características de la misma.

El uso de software libre para la mayoría de las soluciones nos da cierta base en seguridad ya que como el código fuente puede ser auditado por cualquiera para la búsqueda de vulnerabilidades haciéndolas públicas y solucionadas casi de manera inmediata, implícitamente se nos esta dando un servicio de respuesta a incidentes que cualquier compañía comercial envidiaría tener.

Esto nos lleva a pensar que por la falta de publicidad y conocimiento en productos de software libre, aunado al sentido de inseguridad en no tener una empresa que nos respalde al momento de cualquier incidente y al desconocimiento de los alcances del mismo, a que tengamos cierta desconfianza de las tecnologías sin un respaldo contratado corporativamente, por ende y aunado a las deficiencias de cultura informática que sufre nuestro país casi no se implementan soluciones de seguridad con software libre.

Aunque en los últimos años gracias a portales informativos, comunidades de usuarios en las que se incluye además de estudiantes, gente que labora en la iniciativa privada y en dependencias de gobierno; congresos y demás actividades que empiezan a tomar relevancia dentro de la vasta comunidad de profesionales en sistemas de información y por ende estos conocimientos se pasan a

la comunidad en general, dentro de esto hay que tener muy en cuenta la ya probada y aceptada seguridad que se puede implementar en los sistemas, lo anterior con base en el gran numero de empresas públicas y privadas alrededor del mundo que usan tecnologías libres.

Aquí no solamente se esta hablando de proteger a las máquinas que prestan los servicios claves en nuestra empresa, ya que aunque ésto es de suma importancia, nosotros nos avocaremos a dar una seguridad extra a estos servicios poniendo una barrera que examine y filtre la información que entra y sale en nuestros servidores.

1.5 Justificación.

Desde hace unos 6 años las empresas que se dedican a la creación de software empezaron a tomar más en serio y como un estándar en todos sus productos el concepto de seguridad. En México, la sociedad en general que hace uso de software comercial apenas hace 6 años se comenzaron a escuchar estos términos, gracias a las múltiples vulnerabilidades descubiertas y aprovechadas por hackers y crackers, terminando estas en los sistemas con virus tan efectivos como el "*test de inteligencia*", "*I love you*", y más recientemente "*Blaster*" y "*Sasser*" con todos sus derivados, estos provocaron que la sociedad se diera cuenta de todos estos términos y lo más importante, que los comprendiera, lo que conllevó a que desde un usuario en particular hasta las empresas más grandes empezaran a invertir además de en licencias de sistemas operativos y software de ofimática, en antivirus y más recientemente en firewalls y antispyware para protegerse de las miles de amenazas a las que nos exponemos cada que nos conectamos a Internet.

Cuando pensamos que ya vimos todo se descubren nuevas vulnerabilidades en nuestro software lo que implica nuevas ventanas para un posible aprovechamiento de las mismas y causar estragos en sistemas que no estén protegidos debidamente, esto en un ambiente de prestación de servicios informáticos nos lleva a pensar el planteamiento y desarrollo del proyecto, qué tipo de equipos se van a utilizar y el sistema operativo que los va a administrar; además del software de gestión que prestara el servicio deseado, conllevando en todo esto y con una etiqueta primaria en nuestros días el concepto de seguridad informática.

Por estas razones se han empezado a desarrollar nuevos conceptos en seguridad como los que vamos a tratar en este documento, que son el perímetro de seguridad aunado a una alta disponibilidad que nos permite asegurar que nuestro o nuestros servicios funcionen de manera amplia y continua, de forma segura tanto para nosotros como prestadores de servicio como para nuestros usuarios, todo este concepto bajo software libre.

Un concepto básico dentro de este documento es el cortafuegos o firewall *(Figura 1.2)*, que es el equipo destinado a dar protección a las máquinas que estén dentro de nuestra red local.

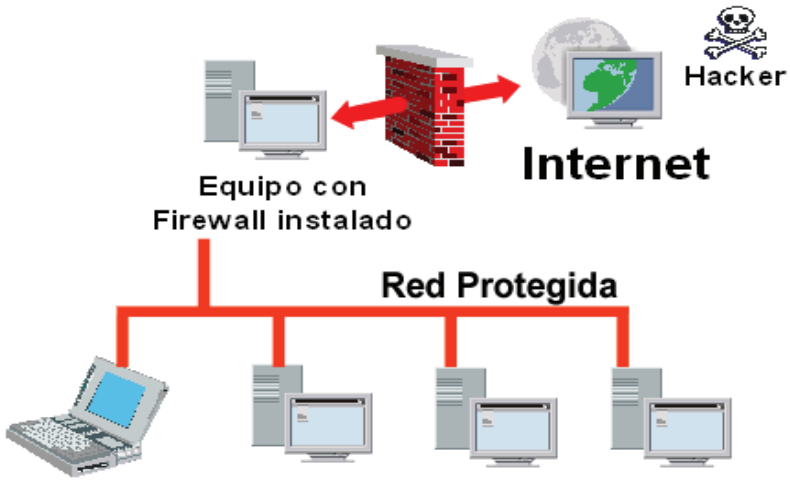


Figura 1.2

Con esto nos estamos protegiendo del 90% de los posibles intrusos y usuarios mal intencionados de nuestro sistema ya que la mayoría de los ataques son ya conocidos y por ende los firewalls están preparados para protegernos ante ellos, y se está justificando plenamente el gasto tanto en equipo como personal humano capacitado, que nos pueda prevenir y rescatar de cualquier futura contingencia en nuestros sistemas de cómputo que además de proveer servicios son la herramienta de trabajo diaria, de muchas personas en general.

1.6 Impacto Social y Tecnológico

En el aspecto social el beneficio se verá reflejado en los bajos costos en cuanto a la inversión se refiere, ya que la compra de los equipos que tenga la memoria suficiente para nuestras aplicaciones, que manejen tecnologías de velocidades GigaBit en sus tarjetas de red, y con redundancia en las fuente de poder es mucho menos costoso que un firewall de fábrica programado por terceros, ya que estos por más flexibles que sean, siempre tendremos alguna limitante por parte del fabricante que caiga en una dependencia tecnológica ya que a final de cuentas es un producto comercial, lo cual no se puede comparar con las limitantes que nos anteponga uno programado por nosotros ya que ahí la limitante es casi siempre nuestro conocimiento.

Tecnológicamente tenemos la gran ventaja de acabar con un único punto de falla en nuestra red que es lo que representa el firewall en si, dándole mayor robustez al mismo.

1.7 Alcances del Proyecto

En cuanto a protección técnicamente se podría decir que tendremos toda la posible en nuestros días, ya que se aplicará el concepto de redundancia tanto en el hardware como en el software contando con 2 equipos con fuente de poder redundante cada uno, además de que estarán sincronizados en el estado y tráfico de los paquetes que pasen y sean recibidos por cualquiera de los 2.

Tendrán una topología de estrella, utilizaremos Ethernet GigaBit sobre cable UTP categoría 6 como medio físico de transmisión.

El firewall estará configurado con el sistema operativo OpenBSD 4.1 inicialmente y básicamente se configuraran los protocolos CARP[1], PFSYNC[2] además de las reglas de PACKET FILTER[3] para la construcción de nuestro firewall.

1.- Ver capitulo III
2.- Ver capitulo IV
3.- Ver capitulo V

1.8 Limitaciones del Proyecto

El esquema de alta disponibilidad por medio de los protocolos mencionados solo puede trabajar bajo la plataforma OpenBSD. Lo anterior se basa en una búsqueda de soluciones a la alta disponibilidad de esta tesis, ya que por ejemplo en otros sistemas operativos como Linux o Windows no existe algún software que nos pueda sincronizar paquetes para proveer una arquitectura de alta disponibilidad.

En cuanto a temas de seguridad nos limitaremos a describir el concepto de alta disponibilidad por medio de los protocolos CARP y PFSYNC y la protección de los servicios prestados por nuestros servidores por medio de nuestro cortafuegos “firewall” en base a reglas de PF “Packet Filter”.

1.9 Grado de Factibilidad

Al observar que la tecnología en los últimos años avanza a una velocidad increíble, aunado al deseo de las instituciones y empresas por contar con lo último de la misma, la mayoría de las veces ésta es adquirida sin ser aprovechada en su totalidad durante la vida útil de la misma, esto debido principalmente a la incorrecta utilización de estas o a la falta de capacitación de los empleados y/o usuarios finales tanto en el software como en el hardware.

Como consecuencia esta solución se crea basada en una herramienta de muy alta utilidad e importancia, que dependerá al 100% de nosotros en su instalación y mantenimiento, además a el respaldo que se tenga por parte de la garantía del equipo hablando del hardware, siendo ésta transparente para nosotros pero de muchísima ayuda, ya que el grado de protección que tengamos en nuestros servicios será el óptimo y la disponibilidad de los mismos envidiable.

Podemos decir en cuanto al software que no tendría ningún costo y al estar bajo la licencia Berkeley que pertenece al grupo de licencias de software libre y es menos restrictiva que algunas otras de dicho grupo ya que por ejemplo esta licencia permite el uso del código fuente en software no libre.

Con una infraestructura mínima y el personal capacitado, la tecnología presentada en esta tesis se puede implementar en cualquier institución por lo cual es un proyecto factible, viable y novedoso.

2.1 Hardware

En esta parte dejaremos un poco de lado las aplicaciones y servicios, y nos abocaremos a los aspectos y problemas técnicos que implica la elección del diseño de una red según los fines que ésta persiga, de ésta manera y como bien se sabe, por múltiples razones no existe una taxonomía generalmente aceptada dentro de la cual quepan todas las redes de computadoras pero 3 de ellas son las que sobresalen como importantes, las cuales son:

- Redes punto a punto

Las redes Punto a Punto básicamente consisten en muchas conexiones entre pares individuales de máquinas, de esta manera, para ir del origen al destino, en este tipo de red un paquete tiene que visitar primero una o más máquinas intermedias, siendo los algoritmos de ruteo básicos en este tipo de red ya que suele haber múltiples rutas de diferentes longitudes para establecer una conexión.

- Redes de difusión

Estas redes tienen un solo canal de comunicación compartido por todas las máquinas de la red, los paquetes enviados dentro de la red son escuchados por todas las máquinas que la comprenden, cada paquete contiene un campo de dirección que especifica a que máquina va dirigido siendo esta la única en hacer caso del mismo al recibirlo al verificar que el campo de dirección esté dirigido a ella.

- Redes conmutadas o por switch

Básicamente trabaja de la misma forma que las de difusión solamente que en este tipo de red se conoce la ubicación de las máquinas así que se lee el campo de dirección del paquete y son enviados a la máquina que contenga esa misma dirección evitando así tráfico en la red.

Comprendiendo lo anterior nos damos cuenta de que la que más se apega y nos conviene a nosotros es la ultima, esto es las redes conmutadas, en nuestros días, el costo de los switches ha disminuido considerablemente, y la gran ventaja que nos ofrece este en respecto a los concentradores o hubs es la simple inteligencia de estos, es decir estas redes constituyen un ruteo de segunda generación que las lleva a ser más modernas y eficientes, siendo así una elección elegante de nuestra arquitectura de red.

Un criterio alterno para clasificar las redes es por su escala, a continuación presentamos un esquema *(tabla 2.1)* de este criterio el cual analizaremos para ver donde encaja nuestro sistema de seguridad que implementaremos.

Distancia entre Procesadores	***Procesadores Ubicados en el (la) mismo(a)***	***Ejemplo***
0.1 m	Tarjeta de circuitos	Máquina de flujo de datos
1 m	Sistema	Multi-Servidor (Cluster) – Bus debidamente acoplado
10 m	Cuarto	Red de Área Local - LAN
100 m	Edificio	Red de Área Local - LAN
1 km	Campus	Red de Campus - CAN
10 km	Ciudad	Red de Área Metropolitana - MAN
100 km	País	Red de Área Amplia - WAN
1,000 km	Continente	Red de Área Amplia - WAN
10,000 km	Planeta	Red de Ârea Global - GAN

Tabla 2.1

Bajo ésta clasificación nuestro sistema de seguridad entra en el casillero de Una LAN o red de Área Local ya que el espacio físico que requiere nuestro sistema para funcionar no va más allá de los 10 metros.

Vale la pena especificar que este criterio se aplica para el espacio físico mínimo necesario para la interconexión de nuestro sistema aunque los servicios que preste en mismo son para protegernos en Internet.

Antes de proseguir es necesario comentar la necesidad de revisar la compatibilidad del hardware en los equipos de cómputo que tenemos con el sistema operativo a instalar para prever posibles conflictos de hardware en la instalación.

Por otro lado en estándar de comunicación de red será Ethernet Giga bit, también conocida como GigE, es una ampliación del estándar Ethernet (concretamente la versión 802.3ab y 802.3z del IEEE) que consigue una capacidad de transmisión de 1 giga bit por segundo que en la práctica se convierten en unos 100 mega bytes útiles, Funciona sobre cables de cobre (par trenzado) del tipo UTP y categoría 6, y por supuesto sobre fibra óptica. Se decidió que esta ampliación será idéntica al Ethernet normal desde la capa de enlace de datos hasta los niveles superiores. Mientras que para el resto es deseable que el estándar sea tomado del ANSI X3T11, Fiber Channel, FDDI o Fibra Óptica, lo que otorga al sistema compatibilidad hacia atrás con Ethernet y el aprovechamiento de las bondades de la fibra óptica como lo es el ancho de banda que transitaría en la misma aparte de la velocidad en las conexiones entre equipos de cómputo y periféricos de red, esto depende básicamente de los componentes físicos de la misma, estamos hablando de cableado, switches y tarjetas de red las cuales deben de soportar esta velocidad, para poder ser usadas en el esquema propuesto.

2.2 Software

Ya tenemos bien definidos los requerimientos de los componentes físicos que van a ser parte del proyecto, ahora debemos ver la parte lógica, esto es, el como configurar nuestros dispositivos y las aplicaciones que trabajen en ellos para poder comunicarse entre sí y al exterior, es básico mencionar los modelos de comunicación base en nuestro proyecto como son el OSI y TCP/IP.

2.3 Modelo OSI

Este modelo se basa en una propuesta que desarrolló la Organización Internacional de Normas ISO, como primer paso para la estandarización internacional de los protocolos que se usan en las diferentes capas, el modelo es llamado de referencia OSI *(Figura 2.1)* que quiere decir Interconexión de Sistemas Abiertos, esto es, la conexión de varios sistemas entre si no importando la

tecnología con que hallan sido desarrollados, cabe mencionar y recalcar el adjetivo de modelo de referencia, lo cual quiere decir que todos los protocolos no están obligados a tener la misma estructura, nada más se deben de basar en ella para que no se pierda la compatibilidad.
Este modelo consta de 7 capas las cuales de describen a continuación:

Figura 2.1

- Capa Física.

  Esta capa tiene que ver con la transmisión de voltajes y señales por un canal de comunicación, las consideraciones de diseño tienen que ver con el asegurarse de que cuando un lado envié un nivel alto (1), se reciba en el otro lado como un nivel alto (1) y no como un nivel bajo (0).

- Capa de Enlace de Datos.

  La tarea principal del enlace de datos es Recibir peticiones de la capa de red y utiliza los servicios del la capa física. El objetivo del nivel de enlace es conseguir que la información fluya, libre de errores, entre dos máquinas que estén conectadas directamente (servicio orientado a conexión). Para lograr este objetivo tiene que montar bloques de información (llamados tramas en este nivel), dotarles de una dirección de capa de enlace, gestionar la detección o corrección de errores, y ocuparse del control de flujo entre equipos (para evitar

que un equipo más rápido desborde a uno más lento). Cuando el medio de comunicación está compartido entre más de dos equipos es necesario arbitrar el uso del mismo. Esta tarea se realiza en el subnivel de acceso al medio.

- Capa de Red.

  Esta capa se ocupa de conseguir que los datos lleguen desde el origen al destino aunque no tengan conexión directa. Ofrece servicios a la capa superior (capa de transporte) y se apoya en la capa de enlace, es decir, utiliza sus funciones.

  Para conseguir este objetivo tiene que realizar ciertas tareas:

  1. Asignación de direcciones de red únicas.
  2. Interconexión de subredes distintas.
  3. Encaminamiento de paquetes.
  4. Control de congestión.

- Capa de Transporte.

  La función básica de la capa de transporte es aceptar datos de la capa de sesión, dividirlos en unidades más pequeñas si es necesario, pasarlos a la capa de red y asegurar que todos los pedazos lleguen correctamente al otro extremo, es importante que esto se haga de manera eficiente y además que aísle a las capas superiores de los cambios inevitables que seguidamente sufren las inferiores debido a los cambios en el Hardware.

- Capa de Sesión.

  proporciona los mecanismos para controlar el diálogo entre las aplicaciones de los sistemas finales. En muchos casos, los servicios de la capa de sesión son parcialmente, o incluso, totalmente prescindibles. No obstante en algunas aplicaciones su utilización es ineludible.

La capa de sesión proporciona los siguientes servicios:

Control del Diálogo: Éste puede ser simultáneo en los dos sentidos (full-duplex) o alternado en ambos sentidos (half-duplex).

Agrupamiento: El flujo de datos se puede marcar para definir grupos de datos.

Recuperación: La capa de sesión puede proporcionar un procedimiento de puntos de comprobación, de forma que si ocurre algún tipo de fallo entre puntos de comprobación, la entidad de sesión puede retransmitir todos los datos desde el último punto de comprobación y no desde el principio.

Todas estas capacidades se podrían incorporar en las aplicaciones de la capa 7. Sin embargo ya que todas estas herramientas para el control del diálogo son ampliamente aplicables, parece lógico organizarlas en una capa separada, denominada capa de sesión.
La capa de sesión surge como una necesidad de organizar y sincronizar el diálogo y controlar el intercambio de datos.

La capa de sesión permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Una sesión permite el transporte ordinario de datos, como lo hace la capa de transporte, pero también proporciona servicios mejorados que son útiles en algunas aplicaciones. Se podría usar una sesión para que el usuario se conecte a un sistema remoto de tiempo compartido o para transferir un archivo entre dos máquinas.

- Capa de presentación.

El objetivo de la capa de presentación es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres (ASCII, Unicode, EBCDIC), números (little-endian tipo Intel, big-endian tipo Motorola), sonido o imágenes, los datos lleguen de manera reconocible.
Esta capa es la primera en trabajar más el contenido de la comunicación que cómo se establece la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas.
Por lo tanto, podemos resumir definiendo a esta capa como la encargada de manejar las

estructuras de datos abstractas y realizar las conversiones de representación de datos necesarias para la correcta interpretación de los mismos.
Esta capa también permite cifrar los datos y comprimirlos. En pocas palabras es un traductor

- Capa de Aplicación.
  Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.
  Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición "HTTP/1.0 GET index.html" para conseguir una página en html, ni lee directamente el código html/xml.
  Entre los protocolos (refiriéndose a protocolos genéricos, no a protocolos de la capa de aplicación de OSI) más conocidos destacan:
  - HTTP (HyperText Transfer Protocol) el protocolo bajo la www.
  - FTP (File Transfer Protocol) transferencia de ficheros.
  - SMTP (Simple Mail Transfer Protocol) envío y distribución de correo electrónico.
  - POP (Post Office Protocol)/IMAP: reparto de correo al usuario final.
  - SSH (Secure SHell) principalmente terminal remoto, aunque en realidad cifra casi cualquier tipo de transmisión.
  - Telnet otro terminal remoto, ha caído en desuso por su inseguridad intrínseca, ya que las claves viajan sin cifrar por la red.

Ahora veremos el modelo que sigue TCP/IP y su comparación con el modelo OSI

2.4 Modelo TCP/IP

Es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

En ocasiones se la denomina *conjunto de protocolos* ***TCP/IP****(figura 2.2)*, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra el popular HTTP (HyperText Transfer Protocol), que es el que se utiliza para acceder a las páginas web, además de otros como el ARP (Address Resolution Protocol) para la resolución de direcciones, el FTP (File Transfer Protocol) para transferencia de archivos, y el SMTP (Simple Mail Transfer Protocol) y el POP (Post Office Protocol) para correo electrónico, TELNET para acceder a equipos remotos, entre otros.

El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes Sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN). TCP/IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en ARPANET, una red de área extensa del departamento de defensa, enseguida veremos un .

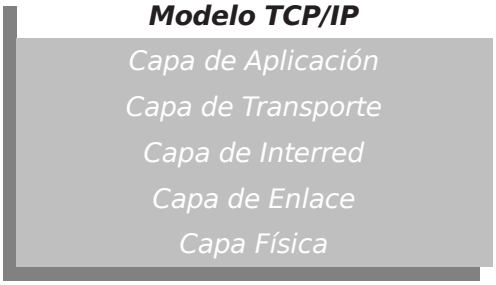


Figura 2.2

- Capa Física.

  Describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado para la comunicación (como las comunicaciones por cable, fibra óptica o radio), y todo lo relativo a los detalles como los conectores, código de canales y modulación, potencias de señal, longitudes de onda, sincronización y temporización y distancias máximas. La familia de protocolos de Internet no cubre el nivel físico de ninguna red; véanse los artículos de tecnologías específicas de red para los detalles del nivel físico de cada tecnología particular.

- Capa de Enlace.

  Especifica como son transportados los paquetes sobre la capa física, incluido los delimitadores (patrones de bits concretos que marcan el comienzo y el fin de cada trama). Ethernet, por ejemplo, incluye campos en la cabecera de la trama que especifican que máquina o máquinas de la red son las destinatarias de la trama. Ejemplos de protocolos de nivel de enlace de datos son Ethernet y Wireless entre otros.

- Capa de Interred.

  Soluciona el problema de conseguir transportar paquetes a través de una red sencilla. Ejemplos de protocolos son X.25 y *Host/IMP Protocol* de ARPANET.
  Con la llegada del concepto de Interred, nuevas funcionalidades fueron añadidas a este nivel, basadas en el intercambio de datos entre una red origen y una red destino. Generalmente esto incluye un enrutamiento de paquetes a través de una red de redes, conocida como Internet.

  En la familia de protocolos de Internet, IP realiza las tareas básicas para conseguir transportar datos desde un origen a un destino. IP puede pasar los datos a una serie de protocolos superiores; cada uno de esos protocolos es identificado con un único "Número de protocolo IP". ICMP y IGMP son los protocolos 1 y 2, respectivamente.

Algunos de los protocolos por encima de IP como ICMP (usado para transmitir información de diagnóstico sobre transmisiones IP) e IGMP (usado para dirigir tráfico multicast) van en niveles superiores a IP pero realizan funciones del nivel de red e ilustran una incompatibilidad entre los modelos de Internet y OSI. Todos los protocolos de enrutamiento, como BGP, OSPF, y RIP son realmente también parte del nivel de red, aunque ellos parecen pertenecer a niveles más altos en la pila.

- Capa de Transporte.

Los protocolos de la capa de transporte pueden solucionar problemas como la fiabilidad ("¿alcanzan los datos su destino?") y la seguridad de que los datos llegan en el orden correcto. En el conjunto de protocolos TCP/IP, los protocolos de transporte también determinan a que aplicación van destinados los datos.

TCP (protocolo IP número 6) es un mecanismo de transporte fiable y orientado a conexión, que proporciona un flujo fiable de bytes, que asegura que los datos llegan completos, sin daños y en orden. TCP realiza continuamente medidas sobre el estado de la red para evitar sobrecargarla con demasiado tráfico. Además, TCP trata de enviar todos los datos correctamente en la secuencia especificada. Esta es una de las principales diferencias con UDP, y puede convertirse en una desventaja en flujos en tiempo real (muy sensibles a la variación del retardo) o aplicaciones de enrutamiento con porcentajes altos de pérdida en el nivel de interred.

UDP (protocolo IP número 17) es un protocolo de datagramas sin conexión. Es un protocolo no fiable (*best effort* al igual que IP) - no porque sea particularmente malo, sino porque no verifica que los paquetes lleguen a su destino, y no da garantías de que lleguen en orden. Si una aplicación requiere estas características, debe llevarlas a cabo por sí misma o usar TCP. UDP es usado normalmente para aplicaciones de streaming (audio, video, etc) donde la llegada a tiempo de los paquetes es más importante que la fiabilidad, o para aplicaciones simples de tipo petición/respuesta como el servicio DNS, donde la sobrecarga de las cabeceras que aportan la fiabilidad es desproporcionada para el tamaño de los paquetes.

- Capa de Aplicación.

  Esta es la capa que los programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en ésta capa son aplicaciones específicas que pasan los datos a la capa de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar.

  Algunos programas específicos se considera que se ejecutan en ésta capa. Proporcionan servicios que directamente trabajan con las aplicaciones de usuario. Estos programas y sus correspondientes protocolos incluyen a HTTP (*HyperText Transfer Protocol*), FTP (*Transferencia de archivos*), SMTP (*correo electrónico*), SSH (*login remoto seguro*), DNS (*Resolución de nombres de dominio*) y a muchos otros.

A continuación se muestra una gráfica de la relación del los modelos OSI y TCP/IP *(Figura 2.3)*, en la cual se pueden comparar las equivalencias de dichas capas.

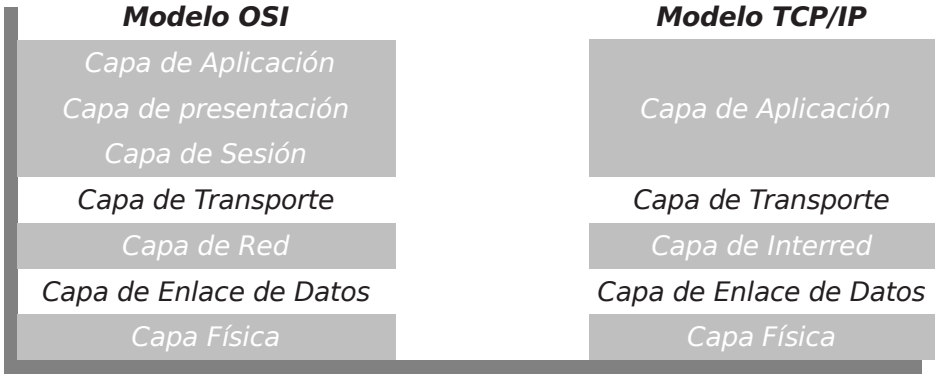


Figura 2.3

2.5 Firewalls

Como ya se menciono anteriormente la seguridad en los sistemas de información ha ido tomando más importancia dentro de nuestra vida diaria, una parte fundamental de la misma y de nuestro proyecto es los Firewalls, en ellos radica la seguridad perimetral de nuestros sistemas de cómputo ya sea que se encuentre en la misma máquina que ofrece nuestros servicios o en una

tercera que sea dedicada exclusivamente a tal fin, a continuación se muestra un diagrama básico *(Figura 2.4)* que explica el funcionamiento y ubicación del firewall.

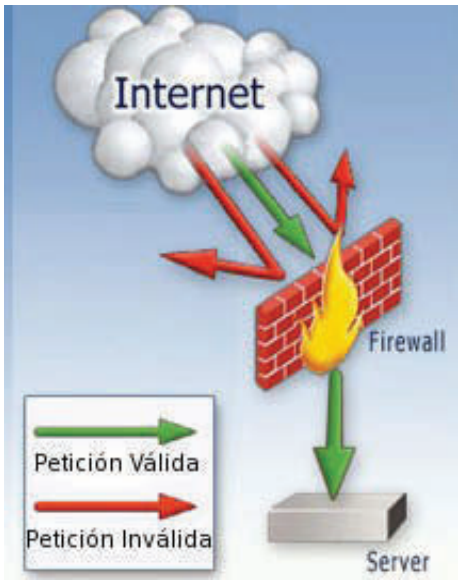


Figura 2.4

El objetivo de la figura anterior es mostrar gráficamente cómo por medio del firewall se crea lo que llamamos el perímetro de seguridad que en este caso esta protegiendo a los equipos que compongan la red interna dentro de la cual estarán los servidores de nuestra empresa, así todo lo que llegue del exterior es analizado por el firewall decidiendo éste si pasan o no los paquetes, de esta manera se conforma nuestra primera barrera de seguridad y muchas veces la más importante o la única.

Un firewall es un sistema o conjunto de sistemas que imponen unas políticas de seguridad entre nuestra red y el mundo exterior, básicamente Internet, el Firewall determina cual de los servicios de red internos ofrecidos por nuestra empresa pueden ser accesados desde el exterior, por quién y de que manera pueden hacerlo.

Existen muchos tipos de firewalls en la actualidad que veremos más adelante, independientemente de eso, existen dos políticas básicas para la configuración de cualquiera de ellos y a continuación se mencionan.

- Desechar o rechazar todo el tráfico, excepto lo que este explícitamente permitido.
- Permitir todo el tráfico, excepto lo que esta explícitamente rechazado o prohibido.

La segunda opción no es muy recomendable por obvias razones pero por lo general es la que se utiliza cuando no se tienen muchos conocimientos en la materia o se tienen políticas pobres de seguridad, de esta manera tenemos que la primera opción es la adecuada, dado el principio básico de todo administrador de red, que entre menos cosas ofrezcas o tengas disponibles en tu red, menos cosas debes de cuidar y por ende menos posibilidades de compromiso tienes dentro de tu sistema de seguridad.

La otra clasificación depende en un 100% de como hace cumplir las reglas el firewall, dentro de esta clasificación tenemos las siguientes opciones:

- De filtraje de paquetes

  Este es el más simple y antiguo de todos, se basa en la examinación de los encabezados de cada paquete y con base en un conjunto de reglas, decide si el paquete es aceptado y reenviado a su destino, o bien. si el paquete es desechado o rechazado
  Las reglas pueden estar basadas en :

  - Dirección de origen (Dirección IP o un rango de estas).
  - Dirección destino del paquete.
  - Protocolo de transporte (TCP, UDP, ICMP, ...).
  - Puerto origen.
  - Puerto Destino.
  - Sentido del paquete (entrante o saliente).

  Unas de las ventajas de este tipo de Firewall son la de un bajo impacto en el desempeño además de bajos costos, cabe mencionar que la mayoría de los ruteadores suelen contener este tipo de firewalls.
  Por otro lado las desventajas que tenemos son que sólo examinan encabezados de capas de red y transporte, no hay verificación de contenido, vulnerables a ataques de ***spoofing***, No ofrecen autenticación de usuarios.

- De estado del sistema

  Este tipo registra el estado de las conexiones en la capa de transporte conforme los paquetes pasan a travéz de él, la decisión de permitir o negar el acceso se toma con base en:

  - Si el paquete es parte de una conversación previamente iniciada, el acceso es permitido.
  - En caso contrario, se evalúa un conjunto de reglas similar al de un cortafuegos de filtraje de paquetes para determinar la acción correspondiente.

  Este tipo de firewall ha evolucionado en diferentes tipos del mismo por mencionar algunos de ellos tenemos los siguientes:

  - Proxy

    También conocido como **Application gateway**, **proxy gateway** o **proxy server**, Provee un mayor nivel de seguridad que el de un filtro de paquetes ya que es posible tomar decisiones con base en la información de la capa de aplicación de los paquetes.
    Un cortafuegos proxy provee los siguientes beneficios:

    - Invisibilidad de los clientes
    - Filtraje de contenido
    - Un punto único para el monitoreo de actividad y registro de Bitácoras.

    Este tipo de firewall comúnmente utilizado para almacenar en caché las paginas web más utilizadas, con el objeto de mejorar la navegación en internet, aligerando el tráfico.
    Las desventajas de este tipo de firewall están en que tenemos un punto de falla único, además de que la mayoría de ellos tienen configuraciones por defecto austeras.

- Nivel de circuito

  Esta tecnología valida que los paquetes pertenezcan ya sea a una solicitud de conexión o bien a una conexión entre dos computadoras. Aplica mecanismos de seguridad cuando una conexión *TCP* o *UDP* es establecida. Una vez que la conexión se establece, los paquetes pueden ir y venir entre las computadoras sin tener que ser revisados cada vez.
  El firewall mantiene una tabla de conexiones válidas y permite que los paquetes de la red pasen a través de ella si corresponden a algún registro de la tabla. Una vez terminada la conexión, la tabla se borra y la transmisión de información entre las dos computadoras se cierra.

Nuestro proyecto depende en su programación de firewall, de un software llamado Packet Filter que cae dentro de la categoría de análisis de estados del sistema antes mencionada que a su vez y en base a nuestros objetivos es la mejor.

Cabe mencionar que dentro de esta categoría existe una subcategoría que nos divide las arquitecturas de los firewall, en este caso solo mencionaremos las más importantes como son:

- Dual-Homed Gateway[1]
- Screened Host Gateway[1]
- Screened Subnet Firewall[1]

Dada la arquitectura de red en la que se va a poner el marcha el proyecto y la arquitectura de alta disponibilidad que vamos a aplicar no cae estrictamente en alguna de ellas, por lo mismo vamos a omitir su explicación.

Nuestra arquitectura esta dentro de una red clase “B”, dentro de la cual nosotros que estamos ubicados dentro de un segmento de la misma y crearemos nuestra DMZ o Zona Desmilitarizada para nuestros servidores, de acuerdo a esto nuestro esquema o arquitectura de red quedaría como el siguiente diagrama *(Figura 2.5)*:

1.- Ver definición en el glosario

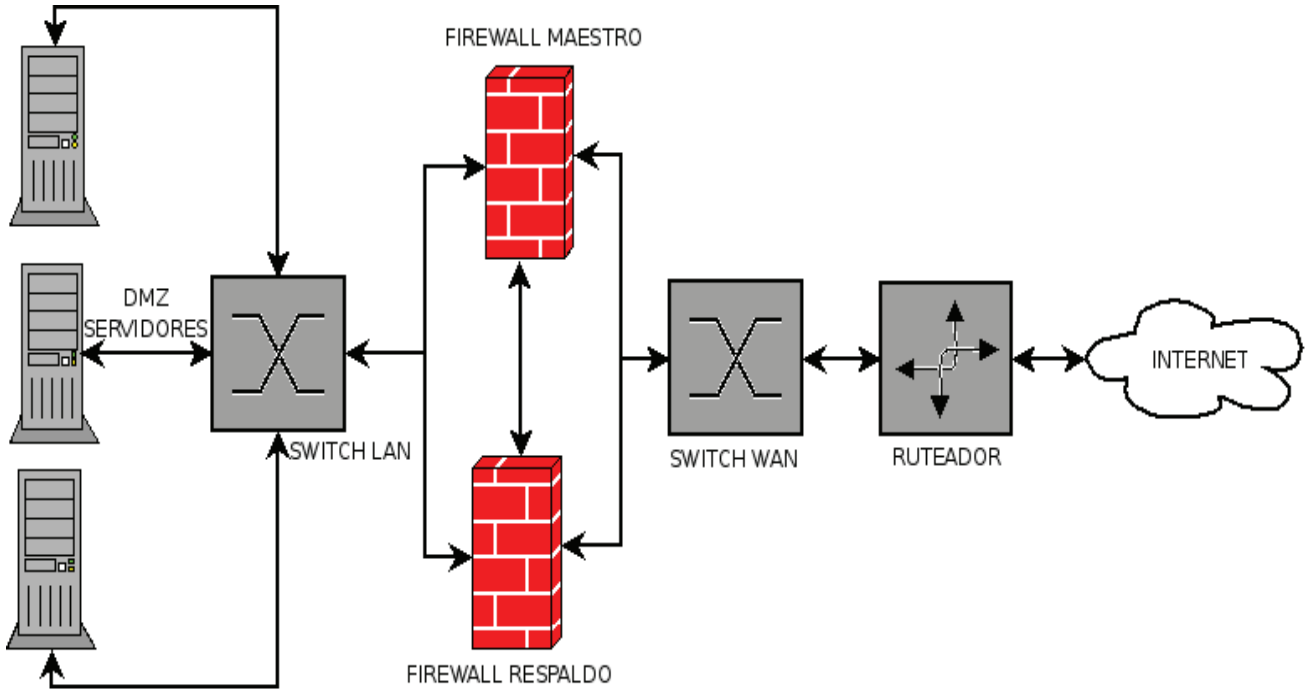


Figura 2.5

De esta manera crearemos un perímetro de seguridad entre nuestros servidores y el exterior, esto por medio de 2 servidores que fungirán como firewalls y estarán comunicados entre ellos aportando redundancia al perímetro de seguridad.

Dentro de este capitulo conoceremos el proyecto OpenBSD de la misma manera que estaremos justificando el porqué se ha elegido el mismo para que sea la base del desarrollo de este proyecto.

3.1 Sistemas BSD

A continuación el por que en mayúsculas las siglas BSD y algunos otros tópicos de interés sobre este proyecto de software libre que beneficia a millones de personas en todo el mundo.

- ¿Que es BSD?

  Las siglas BSD significan "Berkeley Software Distributions". Este software de Berkeley fue realizado por el grupo de desarrolladores de la Universidad de California de Berkeley, aparte de la colaboración del grupo de AT&T y de los laboratorios Bell.

- Historia de BSD

  En 1979, William Joy y Ozalp Babaoglu publicaron la primera versión de UNIX procedente de Berkeley, a la que llamaron 3BSD, el cual convenció a DARPA para que Berkeley desarrollara un sistema operativo con soporte para TCP/IP; asi se creo el estándar 4BSD.
- Con el lanzamiento de 4.2BSD en 1983, termino el proyecto "Berkeley DARPA UNIX". A partir de entonces BSD fue evolucionando hacia nuevas versiones. 4.3BSD Net1 y 4.3BSD Net2 fueron distribuidos en forma de código fuente y binarios. El proyecto seguía avanzando y en la siguiente versión, la 4.4BSD, se mejoraron muchos aspectos (algoritmos de gestión de memoria, mejora de la *suite IP*, compatibilidad con POSIX, etc). La última versión publicada por el Computer Systems Research Group (CSRG) fue 4.4BSD-Lite release 2.

- Sistemas Basados en BSD

- Aunque el Computer Systems Research Group (CSRG) cayera, el desarrollo de 4.4BSD continúa, tanto en sistemas comerciales como en proyectos de software libre. Actualmente los sistemas BSD son utilizados fundamentalmente:

    - En servidores de Internet.
    - Como modelo de estudio en universidades.
    - Como entorno de desarrollo con numerosas y potentes utilerías y compiladores.
    - En computadoras personales como máquinas de escritorio.

  Los sistemas operativos más famosos basados en BSD son los siguientes:

    - **FreeBSD** - Ofrece altísimas prestaciones en unas pocas arquitecturas soportadas, siendo un excelente servidor. Es una de las más populares debido a que también es bastante sencilla de instalar y configurar.

    - **NetBSD** - Sistema ampliamente compatible entre diferentes arquitecturas (más de 52), con un buen diseño. Normalmente es el pionero en la aplicación de nuevas ideas.

    - **OpenBSD** - Se centra tanto en crear el sistema operativo más seguro como en ofrecer una excelente plataforma de desarrollo.

    - **BSD/OS** - Sistema comercial producido por la compañía comercial Berkeley Software Design Inc.(BSDi)

    - **Darwin** - Sistema operativo Open Source, basado en BSD y producido por Apple.

3.2 OpenBSD

Dados las especificaciones más básicas sobre los sistemas BSD ahora nos centraremos en el que usaremos dentro de nuestro proyecto que es el OpenBSD empezando por su historia y objetivos.

- ¿Que es OpenBSD?

  Es un sistema operativo libre tipo UNIX, multiplataforma y basado en 4.4BSD su mayor interés esta centrado en la corrección de código así como la portabilidad, la seguridad proactiva y la criptografía integrada.

- Historia de OpenBSD

  OpenBSD nació en 1995 en razón de la expulsión de Theo de Raadt del grupo de desarrollo de NetBSD, a finales de 1994. Dicho grupo decidió revocarle el acceso a las fuentes debido a que consideraba que no era lo suficientemente colaborador con la gente que pretendía contribuir en el proyecto, además de que pretendía implantar esquemas de seguridad dentro de NetBSD que el resto del grupo no consideraba como importantes dentro del sistema.
  Theo, tras casi un año de negociaciones con el núcleo para que le dejaran incluir en las fuentes todo el trabajo que había estado desarrollando para SPARC, que era la arquitectura de la que el había estado encargado antes de su salida, decidió crear su propio BSD, cuyo principal objetivo fuera la seguridad y la corrección de código.

  En el año de 1996, apareció el nuevo sistema operativo OpenBSD 2.0, la primera versión pública de este sistema concebido por Theo re Raadt y que gracias a sus fuertes objetivos y a la ayuda de las personas que colaboran con el proyecto, ha conseguido alzarse como el sistema operativo más seguro del mercado.

  Años después, aprovechando el lanzamiento de OpenBSD 2.6, decidieron incluir una versión libre del protocolo SSH (Secure Shell), que bautizaron como OpenSSH, basado en las partes libres de los paquetes SSH de Tatu Ylonen y OSSH de Bjorn Gronvall.

Durante todo este tiempo, el equipo de desarrolladores de OpenBSD ha buscado no sólo vulnerabilidades en su propio código, si no que han descubierto fallos en la programación que no se pensaba que pudieran desembocar en fallos explotables, realizando de este modo una labor de investigación aplicada en el campo de la seguridad informática.

Un acontecimiento con bastante eco dentro de la comunidad BSD, fue la decisión de Darren Reed de modificar la licencia de su paquete de software más importante, IPFilter, debido a la cual no se podían realizar cambios a los fuentes sin que el lo autorizara. Por culpa de este cambio de licencia, Theo de Raddt se vio obligado a retirarlo de la distribución.

Mientras esto sucedía un colaborador de OpenBSD, Daniel Hartmeier, comenzó a elaborar un nuevo filtro de paquetes que fue incorporado en la versión 3.0. este nuevo filtro fue bautizado como PF (Packet Filter), el cual ha impactado a muchos por la rápida incorporación de código para permitir Balanceo de Carga y la posibilidad de controlar ancho de banda, a travéz de las reglas de este software.

- Objetivos de OpenBSD
    - Proporcionar una buena plataforma de desarrollo. Dar acceso a las fuentes tanto a desarrolladores como a usuarios, permitiendo observar los cambios directamente del sistema de control de versiones o Concurrent Versions System (CVS).

    - Integrar código de cualquier fuente siempre y cuando sea lo suficientemente bueno y posea un derecho de autor no muy restringido.

    - Se pretende que el código fuente esté disponible para cualquiera y para cualquier propósito.

    - Prestar especial atención en los problemas de seguridad y tratar de solucionarlos antes que ningún otro, de esta manera se intenta que sea el sistema operativo más seguro.

    - Gran interacción de software orientado a criptografía. Tales como IPSec, Ipv6, así como motores de claves como Kerberos, free-AFS entre otras.

- Seguir e implementar estándares ANSI, POSIX, partes de X/Open, entre otros.

- Trabajar en un código lo más independientemente posible de la máquina, de esta manera brindar soporte a tantos sistemas como hardware disponible exista.

- Tener una política lo más libre posible.

- No dejar ningún problema sin solucionar.

- Proporcionar una buena plataforma para el desarrollo y compilación otorgándoles cross- compile o compilación cruzada, la cual consiste en compilar en una máquina, binarios para otra. Como ejemplo podríamos poner el recompilar en una máquina x86, código para una máquina que corra un power-pc, un sistema ARM, o un sparc. Todas ellas, arquitecturas incompatibles entre si.

- Permitir la importación de paquetes de software externos con la menor modificación posible, haciendo la actualización más sencilla.

- Publicar distribuciones en CD-ROM cada 6 meses como una manera de financiar el proyecto.

3.3 Seguridad en sistema operativo

El enfoque del proyecto en el sentido de la seguridad ha sido claro desde las primeras versiones hasta la actual, mejorando "***release***" a "***release***" e incluyendo mejoras como la generación aleatoria de PIDs (***Process ID o ID de proceso***, que es el número con el que podemos referenciar a cada uno de los procesos del sistema), la generación aleatoria de los números de secuencia iniciales de las conexiones TCP, la integración de algoritmos de criptografía en el kernel, la separación de privilegios para la mayoría de ***daemons*** del sistema, el uso de políticas de acceso para llamadas al sistema (**systrace)** o protección de pila (**propolice)**, entre muchos más.

Decir que un sistema operativo es seguro por defecto puede sonar un poco arrogante, esto ha llevado a la comunidad “underground” (Hackers y Crackers básicamente), a enfocar sus esfuerzos en sistemas basados en BSD concretamente OpenBDS para demostrar a sus desarrolladores que no existe la seguridad perfecta.

Estamos de acuerdo con la expresión de que la seguridad total no existe y menos tratándose de software en constante desarrollo y evolución, a pesar de esto, en el contexto del sistema operativo OpenBSD, significa que un sistema recién instalado, sea lo más seguro posible sin necesidad de configurar el software para endurecer su seguridad por defecto.

Un sistema OpenBSD recién instalado no sólo no tiene apenas servicios corriendo por defecto, si no que los que tiene están configurados de forma que no supongan peligro de ataques externos. Los “daemons” que corran por defecto en el sistema, así como la mayoría de los que el usuario/administrador encargado del sistema pueda activar/instalar, poseen pocos privilegios, esto unido a que muchos de los servicios están configurados para trabajar por defecto en entornos fuera del ROOTDIR ó chroot hace que los problemas en caso de producirse algún fallo se reduzcan drásticamente.

Estos son tan solo ejemplos de la innumerable cantidad de razones que hacen que los desarrolladores del proyecto OpenBSD puedan anunciar su sistema como seguro por defecto.

3.4 Seguridad Proactiva dentro del proyecto OpenBSD

Seguridad proactiva, siempre hablando dentro del contexto del proyecto OpenBSD, significa que se intenta buscar fallos, auditar todo el código posible del sistema siempre buscando nuevos fallos, nuevos ***bugs***, explorando con técnicas que se van descubriendo en el campo de la seguridad informática, tratando de descubrir esos ***bugs***, entiendo como se pueden originar y aplicando esos conocimientos al resto del código.

De esta forma se consigue tener un sistema en constante fase de auditoría, aumentando la posibilidad de ser los primeros en encontrar un fallo en cualquier parte del sistema y no sólo corrigiéndola para OpenBSD, si no ayudando a otros proyectos a corregir esos fallos y mejorar la

seguridad de sus sistemas en general.

3.5 Modelo de Seguridad de OpenBSD

Como ya hemos comentado antes, el desarrollo de OpenBSD continuo y se publica una nueva versión cada 6 meses. Además del desarrollo en sí del proyecto, una parte de los desarrolladores se encargan únicamente de auditar el código, como el que se va aportando de versión en versión. Esta auditoría de código es llevada por diferentes desarrolladores a lo largo de la vida del proyecto.

Todo lo anterior, comentado en este capítulo, es una de las 2 grandes razones por las que OpenBSD fue el sistema operativo indicado para incluirlo dentro de este proyecto de tesis. (cabe mencionar que en los 11 años que lleva de vida solamente se han encontrado 2 ***bugs*** en su instalación y eso es mucho que decir comparándolo con otros sistemas operativos más populares basados en UNIX u otros como Windows de Microsoft con cientos de ***bugs***)

3.6 Instalación de OpenBSD

Ahora vamos a ver como instalar un sistema OpenBSD bajo una arquitectura i386 ya que los procesadores para nuestro proyecto son de esta característica, el método que vamos a utilizar será por medio de un archivo de imagen ISO de la versión 4.1, aunque El proyecto OpenBSD no hace públicos los archivos de imágenes ISO que usa para producir los CD oficiales. El motivo es que les gustaría que los usuarios adquirieran los CD, ayudando de este modo a recolectar fondos para el desarrollo de OpenBSD. Los derechos sobre la composición del CDROM oficial de OpenBSD pertenecen a Theo de Raadt. Theo no permite la redistribución de los CD oficiales de OpenBSD por parte del público. A modo de incentivo para que los usuarios compren los CD, también se incluyen algunos pequeños extras en el paquete.

Estos derechos sólo son aplicables a la composición del CD, mientras que el sistema OpenBSD en sí mismo es de libre distribución. Nada impide que alguien se baje OpenBSD y componga un archivo de imagen para grabarlo en un CD. Cualquier imagen ISO de OpenBSD que pueda existir en Internet es

un archivo de imagen no oficial, o una violación de los derechos de Theo de Raadt. La fuente de un archivo de imagen no oficial puede o no ser de fiar, y por lo tanto corresponde al usuario el determinar el grado de fiabilidad, en nuestro caso usaremos una imagen de los CDs Oficiales.

OpenBSD dispone de un robusto procedimiento de instalación de gran adaptabilidad, basado en texto, es decir sin interfaz gráfica. El procedimiento de instalación es muy parecido en la mayoría de las plataformas; sin embargo existen algunos detalles en los que se diferencian. En cualquier caso es altamente recomendable leer el documento INSTALL específico para cada plataforma, que se encuentra en el directorio correspondiente a la plataforma del CD-ROM de los servidores de FTP.

- Preinstalación o ***checklist***

  Antes de comenzar la instalación hay que tener una idea clara de lo que se desea hacer que en nuestro caso es un firewall, asi que como mínimo hay que conocer la información siguiente (*Tabla 3.1*):

PREGUNTA	RESPUESTA
¿Nombre de la máquina?	fw1 para el firewall maestro y fw2 para el firewall de respaldo.
¿Componentes instalados y disponibles en la computadora?	Procesador, Tarjeta madre, memoria, tarjetas de red que es lo más importante e indispensable en nuestro caso.
verificar que el hardware de la plataforma se encuentre en la página de compatibilidad correspondiente.	Todo el hardware esta debidamente soportado.
Si hay componentes ISA, hay que saber la configuración de hardware de esos componentes y asegurarse de que cumplen con las que requiere OpenBSD.	No tenemos componentes ISA.
¿Método de instalación a utilizar? (CD-ROM, FTP, etc.).	Imagen ISO original en CD-ROM

¿Cómo se va a actualizar el sistema?	Se hará de forma local y consideramos que para la función especifica de nuestro servidor un disco SCSI de 36 Gb. es más que suficiente para esta tarea.
¿Va a coexistir OpenBSD con otro sistema operativo en la misma máquina?	No, ya que su función como firewall será permanente.
¿Cómo queremos subdividir la parte de OpenBSD de nuestro disco?	Se dividirá en 2 particiones una para ***root*** (/) y otra para intercambio (***swap***).
¿Nombre de dominio?	midominio.org
¿Dirección IP del Servidor de Nombres de Dominio ó DNS?	192.168.1.254
● ¿Dirección IP del fw1 en su interfaz hacia la WAN o Internet?	192.168.1.101
● ¿Dirección IP del fw2 en su interfaz hacia la WAN o Internet?	192.168.1.102
● ¿Dirección IP del fw1 en su interfaz hacia la LAN ó red interna?	192.168.252.101
● ¿Dirección IP del fw2 en su interfaz hacia la LAN ó red interna?	192.168.252.102
● ¿Dirección IP del fw1 en su interfaz conectada directamente con la interfaz del fw2?	172.16.147.101
● ¿Dirección IP del fw2 en su interfaz conectada directamente con la interfaz del fw1?	172.16.147.102
¿Dirección IP de la pasarela o gateway?	192.168.1.254
¿Se va a usar el sistema gráfico X Window System?	No.

Tabla 3.1

Una vez conociendo las respuestas estamos listos para introducir el CD-ROM y arrancar el sistema, como comentario adicional, es posible cancelar el intento de instalación desde casi cualquier punto del proceso de instalación de OpenBSD, pulsando CTRL-C, y volver a empezar sin tener que reiniciar, ejecutando install desde el punto de inserción del intérprete.

Cabe mencionar que los datos mencionados en los apartados de nombre de la máquina y configuración de red son meramente descriptivos, y representan los de una típica LAN.

De esta manera arrancamos con las imágenes de instalación dando una breve explicación de la misma apoyándonos con imágenes:

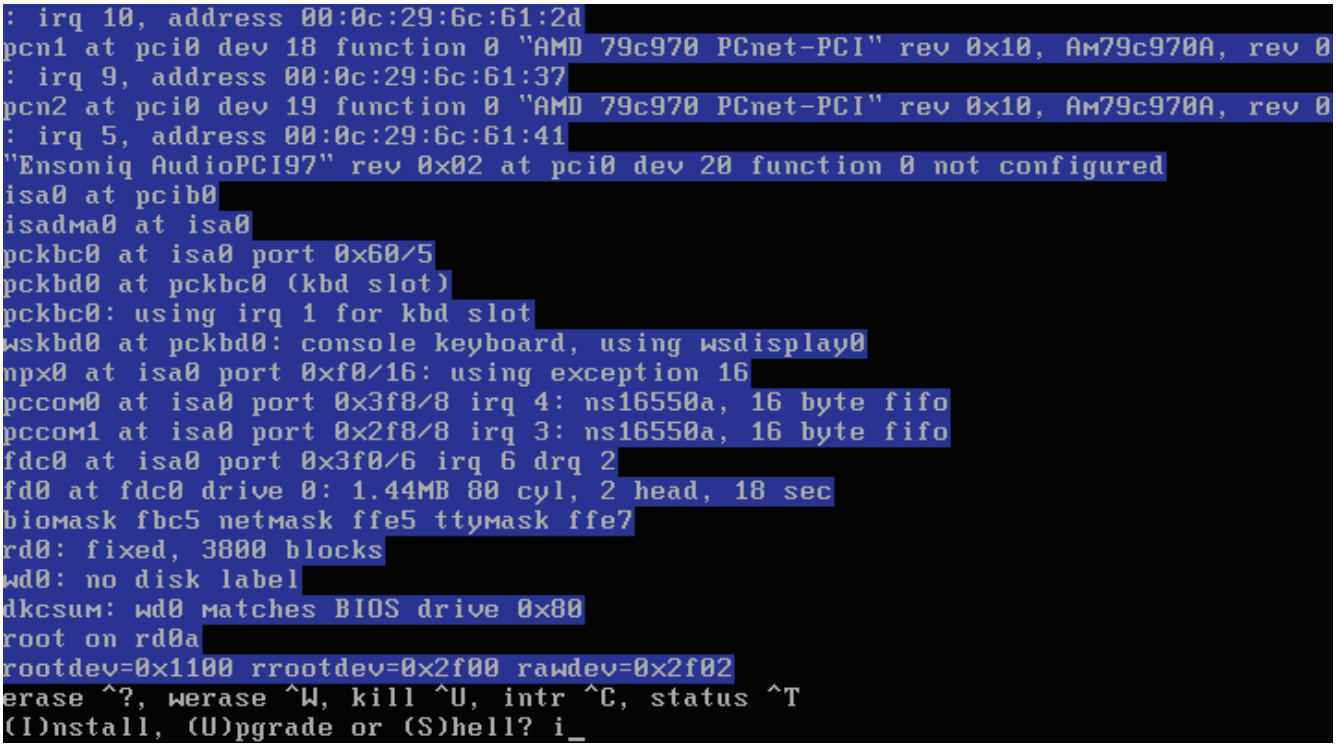


Figura 3.1

Después de haber realizado un arranque con éxito, se verá muchas líneas de texto de un mensaje pasando por la pantalla (*figura 3.1*). Este texto, que en muchas arquitecturas es en blanco sobre fondo azul, es la salida de "***dmesg***", el núcleo del sistema informando sobre qué dispositivos se han encontrado y dónde han sido encontrados. No hay que preocuparse por recordar estas líneas, ya que una copia del texto se guarda en el archivo /var/run/dmesg.boot. En la mayoría de las arquitecturas, las teclas SHIFT+RePág permite examinar el texto que ya ha pasado por la pantalla.
En la parte final de la pantalla se puede observar la primera pregunta del asistente de instalación del sistema la cual básicamente nos solicita el estado de nuestra instalación o arranque desde cd, dicha pregunta nos da 3 opciones las cuales son:

- **Install**: instalar, cargar OpenBSD en el sistema, sobreescribiendo cualquier otra cosa que pueda existir en él. Nótese que se puede dejar algunas particiones sin tocar durante este proceso, como /home, pero en cualquier otro caso hay que asumir que se sobreescribirá cualquier otra, esta el la opción que seleccionaremos en nuestro caso ya que se trata de una instalación a bajo nivel en un servidor nuevo, así que presionando la letra I seleccionaremos esta opción.

- **Upgrade**: actualizar, instalar un nuevo grupo de archivos de instalación en la máquina, pero sin sobrescribir ninguna información de configuración, datos de usuario o programas adicionales. En esta opción no se lleva a cabo ningún formateo del disco, ni se sobreescriben los directorios /etc o /var. Unas notas importantes:

    - No se ofrece la opción de instalar el archivo etc39.tgz. Después de la instalación es necesario fusionar los cambios en etc39.tgz a mano en el sistema para que éste sea completamente funcional. Este paso es muy importante, y se debe realizar ya que sin él algunos servicios clave como pf(4) podrían no iniciarse.

    - El proceso de actualización no está diseñado para saltarse versiones. Aunque esto suele funcionar, no existe soporte para ello. Para OpenBSD 4.1, actualizar 4.0 a 4.1 es la única actualización soportada. Si tiene que actualizar desde una versión más antigua, se recomienda una reinstalación completa.

- **Shell**: (el intérprete de comandos) a veces puede ser necesario llevar a cabo reparaciones o mantenimiento en un sistema que no arranca, o que no debería arrancar, con un núcleo normal. Esta opción permite invocar al intérprete para realizar estas operaciones al sistema.

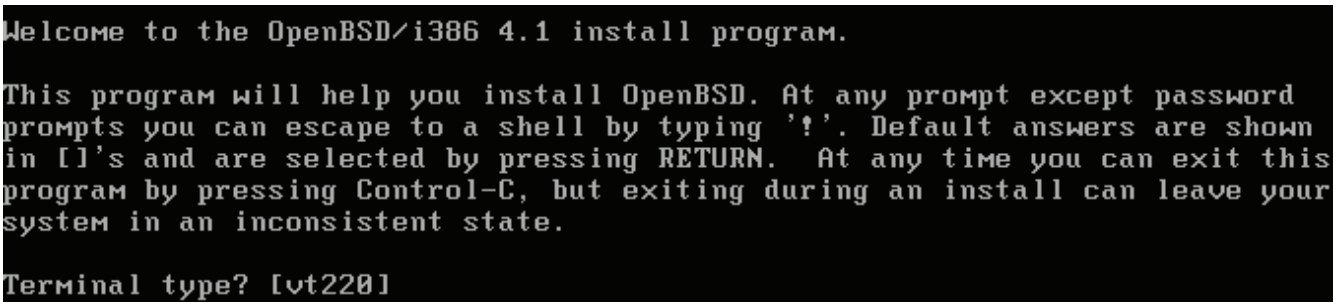


Figura 3.2

Después de nos da una leyenda de bienvenida y nos da la explicación del ayudante en la instalación (*figura 3.2*), seguido de esto nos pregunta sobre el tipo de terminal que utilizaremos para la instalación, en la mayoría de los casos el terminal predeterminado es el apropiado; sin embargo, si se está usando una consola serie tipo tty para la instalación, no se debe escoger el predeterminado sino responder con el terminal que corresponda, en nuestro caso tomaremos el predeterminado que nos marca que es el vt220, y como la opción esta preseleccionada solamente es cuestión de pulsar "enter" para confirmarla.

```
kbd(8) mapping? ('L' for list) [none] L
Major tables: be br cf de dk es fr hu it jp la lt nl no pl pt ru sf sg si sv tr
ua uk us
kbd(8) mapping? ('L' for list) [none] es
kbd: keyboard mapping set to es
```

Figura 3.3

En este paso nos toca seleccionar la configuración de nuestro teclado (*figura 3.3*), tecleando la letra "L", nos despliega los mapas de caracteres que tenemos disponibles, en nuestro caso seleccionaremos "es" que es el correspondiente al idioma español.

```
IS YOUR DATA BACKED UP? As with anything that modifies disk contents, this
program can cause SIGNIFICANT data loss.

It is often helpful to have the installation notes handy. For complex disk
configurations, relevant disk hardware manuals and a calculator are useful.

Proceed with install? [no] yes
Cool! Let's get to it.
```

Figura 3.4

En esta parte (*figura 3.4*) nos advierte que debemos tener un respaldo de los datos contenidos dentro de nuestro disco duro ya que estos son susceptibles a borrarse en el transcurso de este proceso de instalación, así que ya estando enterados de lo anterior nos pregunta si podemos proceder con la misma.

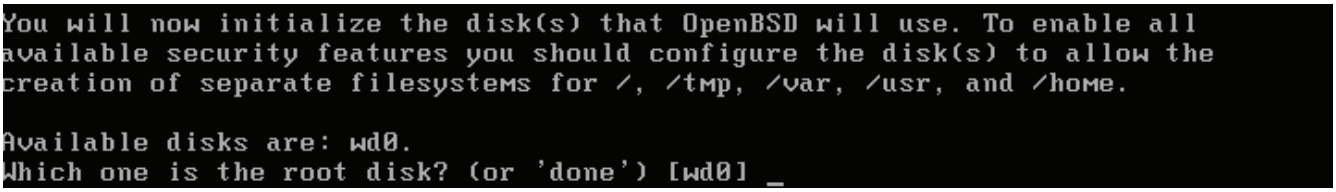


Figura 3.5

Después en esta imagen (*figura 3.5*)nos indica que para habilitar más seguridad al sistema creemos particiones separadas para diferentes partes del sistema como son / (raíz), /home (usuarios), etc. Seguido de esto nos indica cuantos discos duros fueron encontrados en el sistema y nos pregunta que cual de ellos deberemos utilizar para la instalación.

Figura 3.6

Una vez seleccionado el disco duro que vamos a utilizar nos pregunta si usaremos todo el espacio en el disco duro seleccionado para instalar OpenBSD (*figura 3.6*).

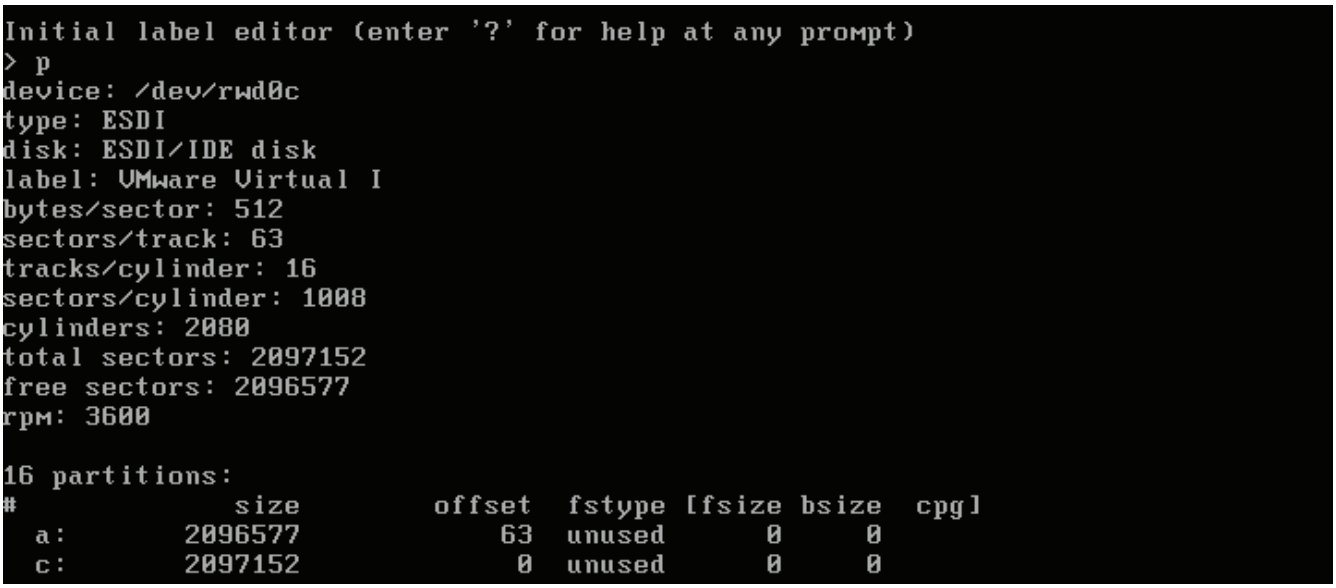


Figura 3.7

Al seleccionar la opción anterior, nos abré una sesión de la herramienta ***disklabel*** que nos ayuda a particionar nuestro disco a nuestra entera voluntad.

Como se muestra en el anterior ***screenshot*** (*figura 3.7*) el comando “p” dentro del ***disklabel***, nos muestra las características del disco duro así como las particiones actuales, en este caso son 2 declaradas por las letras “a” y “c”; la “c” es una partición que simboliza todo el disco duro y no debemos tocarla y la “a” es la partición que se nos creó por defecto al decirle al sistema que íbamos a utilizar todo el disco duro para nuestro OpenBSD.

Figura 3.8

Pero ya que la partición a no tiene ni sistema de archivos ni punto de montaje la borraremos por medio de la opción “d” seguido de la letra de la partición que queremos borrar (*figura 3.8*).

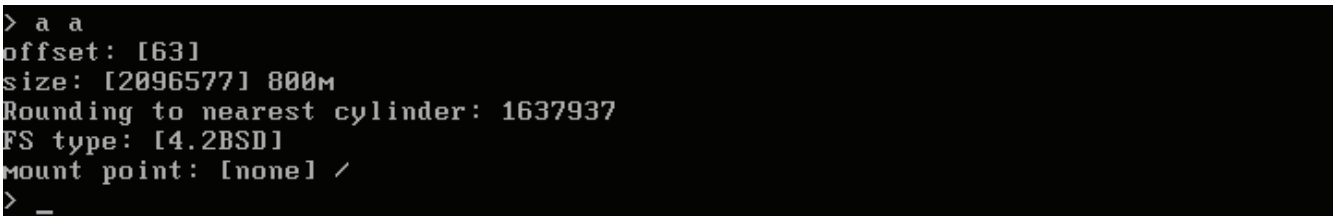


Figura 3.9

Después la volveremos a crear con el comando “a” seguido de la letra que le queremos dar a la partición, de esta manera el ***disklabel*** nos preguntara las opciones necesarias para que este correctamente creada, como son el tamaño, el tipo de sistema de archivos y el punto de montaje(*figura 3.9*), con los datos en el ejemplo hemos creado nuestra partición raíz (/ o root).

Figura 3.10

Ahora creamos de la misma manera la partición “b” (*figura 3.10*), que se utilizara como espacio de intercambio (***swap***).

De esta manera tendremos las particiones básicas para el funcionamiento del sistema operativo que son la raíz (/) y la de intercambio (swap), cabe enfatizar el particionamiento hecho para nuestro sistema es el adecuado ya que no tenemos por que separar en diferentes particiones los

componentes del sistema como /tmp, /home /usr y demás ya que nuestro sistema será dedicado a ser firewall de alta disponibilidad únicamente, por lo tanto sólo será necesario crear una cuenta de usuario para el administrador ya que por razones de seguridad se recomienda evitar el uso de la cuenta ***root*** para acceder al sistema.

```
> p
device: /dev/rwd0c
type: ESDI
disk: ESDI/IDE disk
label: VMware Virtual I
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 2080
total sectors: 2097152
free sectors: 0
rpm: 3600

16 partitions:
#             size        offset  fstype [fsize bsize  cpg]
  a:       1637937            63  4.2BSD   2048 16384   16 # /
  b:        458640       1638000    swap
  c:       2097152             0  unused      0     0
> _
```

Figura 3.11

Una vez terminado el particionamiento podemos dar el comando “p” de nuevo para ver como quedó al final repartido nuestro disco duro (*figura 3.11*).

```
> q
Write new label?: [y] y
No more disks to initialize.

OpenBSD filesystems:
wd0a /
```

Figura 3.12

Ahora procedemos a salir de la utilidad ***disklabel*** con el comando “q” y nos preguntará si queremos escribir las particiones que acabamos que crear a lo que respondemos “y” (*figura 3.12*) seguido de esto nos indica el disco duro que contiene particiones para la instalación del sistema.

Figura 3.13

```
The next step *DESTROYS* all existing data on these partitions!
Are you really sure that you're ready to proceed? [no] yes
/dev/rwd0a:     1637936 sectors in 1625 cylinders of 16 tracks, 63 sectors
        799.8MB in 5 cyl groups (328 c/g, 161.44MB/g, 20608 i/g)
/dev/wd0a on /mnt type ffs (rw, asynchronous, local, ctime=Mon May 22 20:29:21 2
006)
```

Seguido de esto nos indica que el siguiente paso destruirá todos los datos de las particiones y nos da una última advertencia preguntándonos si estamos completamente seguros de lo que vamos a hacer (*figura 3.13*), contestando con la palabra “yes” le damos permiso de formatear nuestras particiones.

```
System hostname? (short form, e.g. 'foo') fw1_
```

Figura 3.14

Ahora nos pregunta en asistente por un nombre que debemos darle al sistema (*figura 3.14*), en nuestro caso le daremos por nombre la abreviatura “*fw1*” que nos indica que es el firewall numero 1 de nuestro sistema de alta disponibilidad.

```
Configure the network? [yes] yes
```

Figura 3.15

Aquí nos esta preguntando si queremos configurar la red de nuestro sistema (*figura 3.15*), a lo que nosotros responderemos “yes”.

```
Available interfaces are: pcn0 pcn1 pcn2.
Which one do you wish to initialize? (or 'done') [pcn0]
Symbolic (host) name for pcn0? [fw1]
The media options for pcn0 are currently
        media: Ethernet autoselect (autoselect)
Do you want to change the media options? [no]
IPv4 address for pcn0? (or 'none' or 'dhcp') 192.168.1.101
Netmask? [255.255.255.0]
IPv6 address for pcn0? (or 'rtsol' or 'none') [none]
```

Figura 3.16

Ahora nos muestra las interfaces disponibles en nuestro sistema en este caso son 3 las que tenemos disponibles (pcn0, pcn1 y pcn2), por que las necesitamos para nuestro sistema de alta disponibilidad (*figura 3.16*), la nomenclatura “pcn” es lo equivalente a la “eth” en distribuciones Linux.

Cabe mencionar que en este paso solo nos limitaremos a dar la explicación de la configuración de

las mismas ya que el porque de todo esto lo veremos a detalle en el siguiente capítulo.

En este caso escogemos la pcn0 que es la que nos da por defecto, seguido de esto nos pregunta por un nombre simbólico para este "***host***", podemos ponerle algún nombre "x", pero en este caso dejaremos el que nos marca el sistema que es el mismo que el nombre del equipo.

Seguido nos pregunta si queremos modificar las opciones de medio, esto tiene que ver con el controlador que le haya asignado el sistema al hardware con el que contamos, por lo regular nunca da problemas y siempre es correcto por esto responderemos que no queremos modificarlo.

Dado lo anterior nos pregunta si configuraremos esta interfaz por medio de DHCP o le daremos una dirección estática, nuestra opción es la segunda por lo que procedemos a darle la dirección IP de dicha interfaz, enseguida nos pregunta qué máscara de subred le corresponde a dicha dirección IP, que en este caso es 255.255.255.0, después nos pregunta si le asignaremos alguna dirección IPV6 a lo que respondemos que no.

De esta manera tendremos configurada nuestra primera interfaz de red.

```
Available interfaces are: pcn1 pcn2.
Which one do you wish to initialize? (or 'done') [pcn1]
Symbolic (host) name for pcn1? [fw1]
The media options for pcn1 are currently
        media: Ethernet autoselect (autoselect)
Do you want to change the media options? [no]
IPv4 address for pcn1? (or 'none' or 'dhcp') 172.16.147.101
Netmask? [255.255.255.0]

IPv6 address for pcn1? (or 'rtsol' or 'none') [none] Available interfaces are: p
cn2.
Which one do you wish to initialize? (or 'done') [pcn2]
Symbolic (host) name for pcn2? [fw1]
The media options for pcn2 are currently
        media: Ethernet autoselect (autoselect)
Do you want to change the media options? [no]
IPv4 address for pcn2? (or 'none' or 'dhcp') 192.168.252.101
Netmask? [255.255.255.0]

IPv6 address for pcn2? (or 'rtsol' or 'none') [none] No more interfaces to initi
alize.
```

Figura 3.17

Ahora nos quedan 2 interfaces por configurar, lo haremos de la misma manera que la primera, lo único que cambia será la dirección IP ya que es importante que las interfaces no estén en el mismo segmento de red (*figura 3.17*).

```
No more interfaces to initialize.
DNS domain name? (e.g. 'bar.com') [my.domain] midominio.com
DNS nameserver? (IP address or 'none') [none] 192.168.1.254
Use the nameserver now? [yes] yes
Default IPv4 route? (IPv4 address, 'dhcp' or 'none') 192.168.1.254
add net default: gateway 192.168.1.254
Edit hosts with ed? [no]
Do you want to do any manual network configuration? [no]
```

Figura 3.18

Ahora nos indica que no hay más interfaces que inicializar o configurar, de esta manera completaremos la configuración de la red (*figura 3.18*) al configurar el nombre de dominio de nuestra máquina que en nuestro caso particular es “midominio.com” (este nombre de dominio es solamente demostrativo), el servidor de nombres o DNS que en este caso es 192.168.1.254 y nuestra salida o gateway que es 192.168.1.254.

```
Password for root account? (will not echo)
Password for root account? (again)
```

Figura 3.19

Ahora nos pregunta por la contraseña para la cuenta de súper usuario o ***root*** (*figura 3.19*). Después de teclearla nos pide que la repitamos para comprobarla, y que quede de esta manera configurada.

```
Let's install the sets!
Location of sets? (cd disk ftp http or 'done') [cd]
Available CD-ROMs are: cd0.
Which one contains the install media? (or 'done') [cd0]
Pathname to the sets? (or 'done') [4.1/i386] i386
The directory 'i386' does not exist.
Pathname to the sets? (or 'done') [4.1/i386] _
```

Figura 3.20

Estamos listos para proceder a la instalación del sistema en nuestro disco duro, primero nos pregunta el tipo de medio utilizaremos para la instalación (*figura 3.20*), por defecto la opción es cd que es la que corresponde a la unidad de CD de nuestro equipo, el asistente prosigue a detectar

nuestras unidades de CD-ROM disponibles, a lo que nos dice que tenemos una llamada cd0 por lo cual le indicamos que ésta es la que contiene el medio de instalación del sistema.
Después nos pregunta por la ruta en donde están los archivos de instalación del sistema que en nuestro caso es la carpeta /4.1/i386.

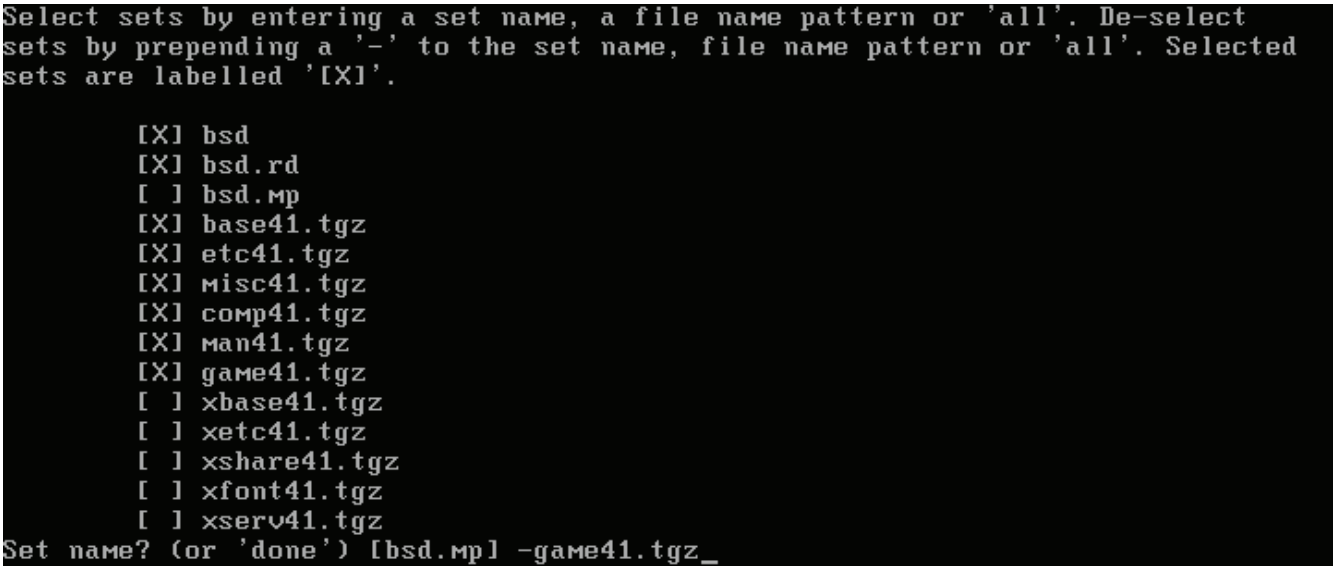


Figura 3.21

Ahora nos muestra una lista de paquetes de software que podemos instalar, por defecto ya vienen algunos seleccionados. Ya que no tendremos sistema gráfico no es necesario seleccionar los de la parte inferior, pero el último seleccionado de la lista tampoco nos interesa por que son los juegos que trae el sistema, de esta manera procedemos a quitarlo de la lista seleccionada mediante el comando “-” seguido del nombre de paquete que no deseamos instalar (*figura 3.21*).

Figura 3.22

Ahora nos vuelve a mostrar la lista de paquetes de software pero vemos que ya no está seleccionado el que eliminamos por lo que procedemos a decirle que ésa es la lista definitiva de paquetes a instalar mediante el comando “done”, enseguida nos pregunta si estamos listos para la instalación a lo que respondemos “yes” (*figura 3.22*).

Figura 3.23

En este ***screenshot*** vemos cómo ha instalado los paquetes previamente indicados (*figura 3.23*).

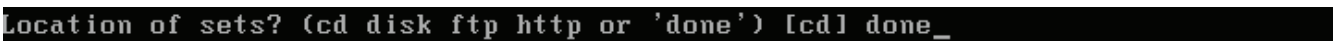


Figura 3.24

Al final nos vuelve a preguntar por el medio de instalación, esto en caso de que queramos instalar alguna otra cosa que se nos haya pasado en el mismo medio o desde otro medio distinto como

puede ser http o ftp (*figura 3.24*).

```
Start sshd(8) by default? [yes] yes
Start ntpd(8) by default? [no] yes
Do you expect to run the X Window System? [yes] no
Change the default console to com0? [no]
Saving configuration files...done.
```

Figura 3.25

Ahora nos pregunta si puede configurar los "***daemons"*** SSH (acceso remoto) y NTPD (Sincronización del reloj) a lo que respondemos "yes" en ambas.
Después nos pregunta si esperamos utilizar en sistema gráfico en esta máquina y como anteriormente lo comentamos la respuesta es no (*figura 3.25*).
El ayudante nos pregunta si queremos cambiar la consola por defecto a lo que también nos negamos, dicho todo lo anterior procede a guardar los cambios en los archivos de configuración.

```
What timezone are you in? ('?' for list) [Canada/Mountain] Mexico
What sub-timezone of 'Mexico' are you in? ('?' for list) General
Setting local timezone to 'Mexico/General'...done.
```

Figura 3.26

Por último configuramos nuestra zona horaria en este caso por la ubicación física en la que nos encontramos la zona horaria primaria sería "México" y la zona interna seria "General" (*figura 3.26*).

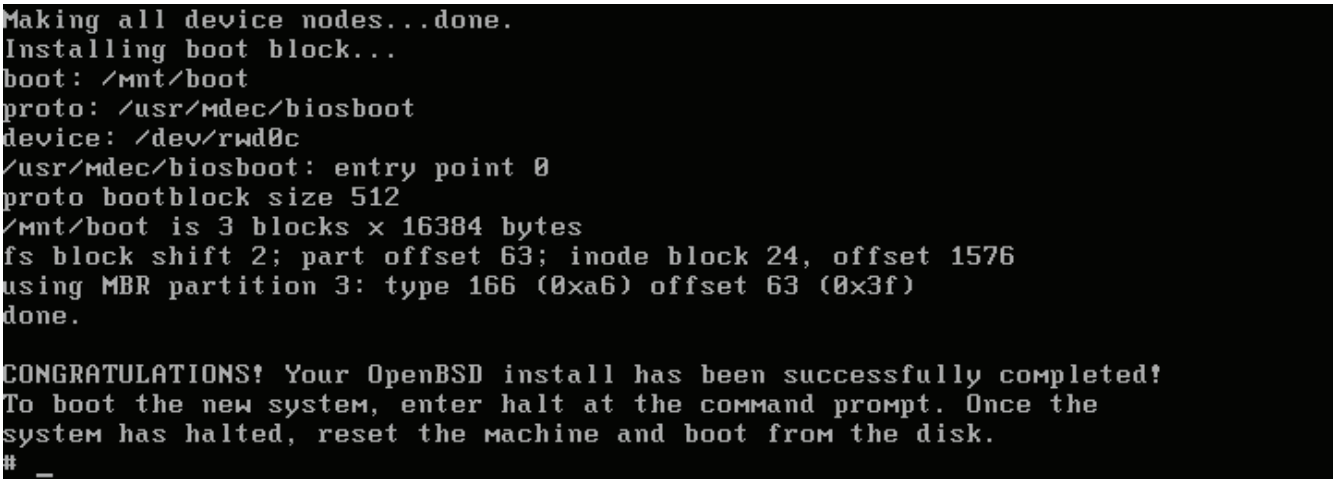


Figura 3.27

Después de algunos ajustes y configuraciones automáticas finales nos felicita por haber terminado exitosamente la instalación y nos invita a reiniciar el sistema por medio del comando "***halt***" (*figura 3.27*) para que todo quede debidamente finalizado. Después de reiniciar, nuestro sistema OpenBSD esta listo para ser configurado como firewall y proveerlo de alta disponibilidad, para ésto es necesario realizar una instalación idéntica a esta en otra computadora la cual fungirá como el otro firewall, las únicas diferencias en la instalación serán:

- El nombre de la máquina, ya que no deben de tener el mismo nombre debido a que físicamente son máquinas diferentes.
  Que en la segunda se llamara "*fw2*".

- Las direcciones IP en cada una de las 3 interfaces, ya que estarán conectadas al mismo segmento de la máquina principal y no se pueden duplicar las direcciones en un mismo segmento.
  En estas solamente el ultimo dígito cambiara del 101 al 102, conservando los 3 primeros dígitos iguales.

4.1 Definiciones.

Antes de empezar a configurar la alta disponibilidad dentro de nuestros firewalls, debemos de aclarar un poco este concepto para después poder proseguir.

- Alta disponibilidad.

  En tecnología informática este término se utiliza para indicar un sistema o un componente que está operacionalmente apto durante un largo período e tiempo.

- Medición de la Alta Disponibilidad.

  La disponibilidad se mide relativa a un 100% operacional o "cero fallas". Un estándar de amplia aplicación pero de difícil logro para tener como objetivo de disponibilidad se conoce como los "cinco nueves" (99.999%) del tiempo que marquemos.

- Redundancia.

  Básicamente esto lo definiremos como la capacidad del hardware para poder reponerse a fallas que se presenten dentro del entorno en que trabajan, como son las fallas eléctricas o de calentamiento, en pocas palabras, eliminando lo más que se pueda cualquier punto único de falla que pueda interrumpir la disponibilidad del servicio.

- Restablecimiento.

  Se refiere a la habilidad para sobreponerse a una falla momentánea, de tal manera que no haya impacto en la disponibilidad para el usuario final. Puede ser tan pequeño como una pequeña porción de la memoria recuperándose de un error insignificante, o algo tan grande como un sistema de servidores que decida invernar sin razón alguna, sin perdida de

información transaccional.

- Robustez.

  Esta característica de alta disponibilidad describe el diseño general del proceso de disponibilidad. Un proceso robusto resistirá una variedad de ataques, tanto internos como externos, que podrían fácilmente interrumpir y dañar la disponibilidad en un ambiente más débil. Robustez implica un alto nivel de documentación y entrenamiento para absorber cambios técnicos a las plataformas, productos, servicios y clientes; cambios de personal cuando hay rotación y expansión, y cambios en los negocios cuando hay nuevos objetivos, adquisiciones, y fusiones.

4.2 Conceptos

Dado que un sistema de cómputo o una red requieren muchos elementos operativos para que el todo funcione, mucha de la planeación para alta disponibilidad se centra alrededor de sistemas de respaldo y procesamiento, almacenamiento y acceso de contingencia.

En nuestro caso , tomando en cuenta que nuestro sistema será dedicado a una sola tarea la cual consiste en revisar todo el tráfico que se dirija a nuestros servidores y eliminar el que no se considere seguro, dichos tópicos se verían reflejados en lo siguiente (*tabla 4.1*).

Tópico	***Relación***
Respaldo	● El respaldo se orientaría a la presencia de un segundo firewall ya que este tendrá las mismas reglas de filtrado que el primero.
Almacenamiento y acceso de contingencia	● El almacenamiento se podrá entender como los ***logs*** de nuestro sistema de firewalls. ● La contingencia la podríamos cubrir con una tercera máquina con configuración igual a las que estén en productivo esto en caso de una falla grave de hardware.

Tabla 4.1

Para que un sistema tenga una alta disponibilidad, es esencial que todos sus componentes estén adecuadamente diseñados y probados exhaustivamente antes de ser utilizados, ya que dentro de nuestro ***checklist*** que realizamos antes de la instalación, éste fue un punto de revisión que da por hecho que todo esta debidamente probado y funciona a la perfección; ahora bien en el caso de que nos estemos refiriendo a la aplicación, si ésta no ha sido probada en su totalidad puede convertirse fácilmente en el eslabón más delgado del sistema y ser el punto de rompimiento frecuente en un sistema de protección.

En nuestro caso nos estaríamos refiriendo a los protocolos CARP y PFSYNC además del el modulo que nos permite poner las reglas de filtrado que es denominado PACKET FILTER, los dos primeros aparecieron dentro de la versión 3.5 del sistema operativo OpenBSD, dado que la ultima versión estable de dicho sistema es la 4.1 y las declaraciones de la comunidad que lo usa y desarrolladores que lo programan, indican que puede considerarse estable y sin errores o ***bugs*** conocidos hasta el día de hoy, por el lado del sistema de filtrado de paquetes éste tiene aún más robustez ya que nació junto con el sistema operativo.

No debemos dejar de lado el mencionar otros tópicos de la alta disponibilidad como serian:

- La seguridad Física.

  Aquí nos estamos refiriendo Básicamente al control de acceso físico que se tendrá para con los servidores, con esto estamos controlando quién y a qué entran al centro de operaciones de una red, que es donde normalmente se encuentran los servidores dentro de una empresa o institución, de esta manera podemos restringir el acceso a personas que por malicia quieran hacer algún tipo de daño al equipo, desde desconectarlo de la red o electricidad, hasta dañar físicamente algún componente de nuestro sistema.

- Redundancia en las Instalaciones eléctricas.

  Esta medida se refiere a la capacidad que tengan los sistemas de alimentarse de 2 fuentes eléctricas distintas a la vez, ya que si tenemos una sola fuente de energía y ésta falla, estamos dependiendo mucho de que ésto se solucione y en la mayoría de los casos no

depende de nosotros el arreglarlo, dentro de este mismo tópico también podemos mencionar la necesidad de tener un sistema eléctrico de respaldo ya que en caso de una contingencia en la falta de corriente eléctrica podamos hacer los respaldos y movimientos necesarios para no permitir la pérdida de datos de nuestro sistema.

En la comunidad de tecnologías de la información, la métrica empleada para medir la disponibilidad es el porcentaje de tiempo que un sistema es capaz de realizar las funciones para las que está diseñado. La disponibilidad es el porcentaje de tiempo que un servicio está activo y en funcionamiento. Se emplea la fórmula siguiente para calcular los niveles de disponibilidad(*Tabla 4.2*):

Formula para obtener el porcentaje de disponibilidad según articulo de TecNet de Microsoft
Porcentaje de disponibilidad = (tiempo total transcurrido – suma de tiempo no disponible)/tiempo total transcurrido

Tabla 4.2

La disponibilidad suele medirse en “nueves” como lo comentamos anteriormente. Por ejemplo, una solución cuyo nivel de disponibilidad sea de “tres nueves” es capaz de realizar su función prevista el 99,9 por ciento del tiempo, lo que equivale a un tiempo de inactividad anual de 8,76 horas por año sobre una base de 24x7x365 (24 horas al día, siete días a la semana, 365 días al año). ahora mostraremos los niveles de disponibilidad(*tabla 4.3*) frecuentes que muchas organizaciones intentan conseguir.

Porcentaje de Disponibilidad y Tiempo de Inactividad Anual	
Porcentaje de Disponibilidad	**Inactividad en Día de 24 horas**
90%	876 horas (36.5 días)
95%	438 horas (18.25 días)
99%	87.6 horas (3.65 días)
99.9%	8.76 horas
99.99%	52.56 minutos
99.999% (“cinco nueves”)	5.256 minutos
99.9999%	31.536 segundos

Tabla 4.3

Desgraciadamente, calcular la disponibilidad no es tan sencillo como seleccionar uno de los porcentajes de disponibilidad que se muestran en la tabla anterior(*tabla 5*). Debe decidir primero qué métrica desea utilizar para calificar el tiempo de inactividad. Por ejemplo, una organización puede considerar que se produce tiempo de inactividad cuando una base de datos no está montada. Otra organización puede considerar que sólo se produce tiempo de inactividad cuando más de la mitad de sus usuarios se ven afectados por una interrupción del servicio, en nuestro caso podremos tomar en cuenta como tiempo de inactividad solamente el requerido para cambios necesarios que se tengan que hacer en las reglas de filtrado y las actualizaciones que requieran un reinicio del sistema como podría ser un cambio de kernel, así que podremos llegar a los cinco nueves deseados con nuestro proyecto.

4.3 Protocolos

- CARP

Existe el protocolo virtual de ruteo redundante ó VRRP, “Virtual Router Redundancy Protocol” por sus siglas en ingles, elimino el punto único de falla en una red estática por medio de la asignación de una salida (***gateway***) virtual entre muchos ruteadores físicos, en nuestro caso y de una manera más simple se trata de asignar una dirección IP única, fija y virtual que se levante mediante varias interfaces físicas de diferentes equipos, esto les permite a varios equipos comportarse como una sola máquina donde una de ellas trabaja como maestro mientras que las demás están en espera como esclavos, de esta manera si la máquina maestro tiene alguna falla o no ésta disponible, una de las que están como reservas se empezará a anunciar como maestro, de esta manera se permite que el tráfico continué fluyendo de manera ininterrumpible sobre la nueva interfaz maestra, desafortunadamente aunque este protocolo es estándar, pesa sobre él una patente puesta por su autor Cisco Systems Inc. Que al mismo tiempo que publican no estar interesados en demandar a nadie que ponga en funcionamiento dicho protocolo, se reservan el derecho sobre levantar demandas sobre la patente de dicho protocolo. OpenBSD necesitó de esta funcionalidad para poder proveer su sistema de una herramienta de alta disponibilidad, y de esta manera dicha patente hizo empobrecer la solución proporcionada por Cisco.

Basados en la dedicación del software libre, el equipo de OpenBSD se puso a trabajar sobre una patente libre que remplazara al protocolo desarrollado por Cisco.

Esto fue realizado con la aparición el Protocolo de Dirección Común Redundante o Common Address Redundancy Protocol por sus siglas en ingles (***CARP***) a finales del 2003, este protocolo opera en las capas de enlace de datos y de red del modelo OSI, usando una dirección Física de control de acceso al medio o Media Access Control Address por sus siglas en ingles (***MAC***), y una o más direcciones IP virtuales, el sistema que trabaja como maestro en el grupo de máquinas dentro del sistema CARP responde a una petición de tipo del protocolo de resolución de direcciones o Address Resolution Protocol por sus siglas en ingles (ARP) que trabajan como su nombre lo indica para identificar máquinas dentro de una red, dichas peticiones se ven como sigue(*tabla 4.4*):

FORMATO DE MENSAJES ARP
18:33:49.908612 arp who-has 192.168.1.2 tell 192.168.1.1
18:33:49.908691 arp reply 192.168.1.2 is-at 0:2:a5:ee:ec:10

Tabla 4.4

En este caso, la máquina *192.168.1.1* pregunta por la dirección ethernet *192.168.1.2* (suponemos ambas máquinas en la misma subred). Como vemos la *192.168.1.2* responde identificándose a sí misma ante las demás.

De esta manera el maestro responde a una petición ARP de la dirección MAC virtual compartida por varias interfaces con IP real, permitiendo así a los switches una rápida obtención del número de puerto al cual mandar el tráfico recibido.

La máquina maestra manda un mensaje de aviso CARP vía multicast usando el mismo protocolo CARP (Protocolo IP numero 112) en un tiempo regular, y la máquina o máquinas esclavo escuchan dichos avisos. Si estos avisos dejan de ser enviados, en caso de ser una sola máquina esclava, ésta empieza a mandar estos avisos como respaldo, siendo la frecuencia de éstos configurable; en caso de ser más de una las máquinas esclavas, la que tenga configurado el tiempo de los avisos más corto, será la que se convierta en maestro, cuando el servidor maestro configurado en el inicio se detiene. En el caso de ser una sola máquina la que se configure como esclavo, será la única que mande avisos y se convertirá en maestro independientemente de la frecuencia de los mismos ya

que no hay otra con la cual compita por convertirse en maestro.
Esto parecerá muy familiar si hemos leído o escuchado acerca del protocolo VRRP de Cisco, pero ahora comentaremos algunas grades diferencias:

- El protocolo CARP es independiente de las familias de direcciones que maneje ya que soporta direcciones IPV4 e IPV6, como transporte de los paquetes CARP, así como las direcciones que se compartirá.

- El protocolo CARP tiene la característica de balanceo sobre ARP, que permitirá que múltiples máquinas compartan una sola dirección IP, de esta manera hay una dirección MAC por cada tarjeta de red pero solo una sola IP en común.

- El protocolo CARP usa un fuerte algoritmo criptográfico, el SHA1-HMAC para proteger cada aviso de la máquina maestro.

Dejando aparte las diferencias técnicas mencionadas anteriormente existe otra que por su contenido podría ser más importante que las anteriores, esta diferencia reside en que este protocolo no tiene una patente que limite su implementación y uso.

- PFSYNC

Este protocolo PFSYNC viene del al abreviación de synchronization of “PF” o Packet Filter, en este caso podríamos tomar como definición que este protocolo es usado por Packet Filter para manejar y actualizar las tablas de estado del firewall, estos mensajes son enviados vía multicast (multidifusión) en una interfaz especifica, usando el protocolo PFSYNC (Protocolo IP 240) y de la misma manera escucha por dicha interfaz mensajes provenientes de la otra máquina importándolos en su tabla local de estados.

Una debilidad de este protocolo es que no incorpora una autenticación en el intercambio de paquetes entre ambas máquinas, por lo que es extremadamente recomendable no usar esta configuración en una red con direcciones IP homologadas, es decir que tengan acceso directo a ellas desde Internet, por lo que se recomienda implementarlo bajo una red LAN privada y en la que

tengamos total control. Para nuestra configuración en la que únicamente 2 máquinas participaran en nuestra implementación es recomendable que dichas máquinas se conecten entre sí para intercambiar paquetes PFSYNC por medio de un cable cruzado sin un HUB o SWITCH de por medio, de esta manera aseguraríamos que los paquetes viajen sin intervención de terceros, anulando un poco así la debilidad mencionada anteriormente.

4.4 Escenario.

Ahora ya habiendo explicado los conceptos de PFSYC y CARP vamos a explicar la configuración de alta disponibilidad que desarrollaremos para nuestro objetivo que es implementar una configuración de alta disponibilidad en firewalls.

Describiendo un poco lo anterior nosotros tendremos 2 máquinas que actuarán como firewalls, cada una con 3 interfaces de red, una de ellas estará trabajando bajo el protocolo CARP para levantar la interfaz o IP por la cual saldrá al exterior o Internet, otra de estas estará conectada por un cable cruzado directamente a la otra máquina o firewall, por esta interfaz se implementará la comunicación de estado de los paquetes por medio de PFSYC, y la última de éstas trabajará el protocolo CARP levantando una IP que será la salida de nuestra LAN y la conexión de la misma hacia nuestros firewalls, quedando como sigue:

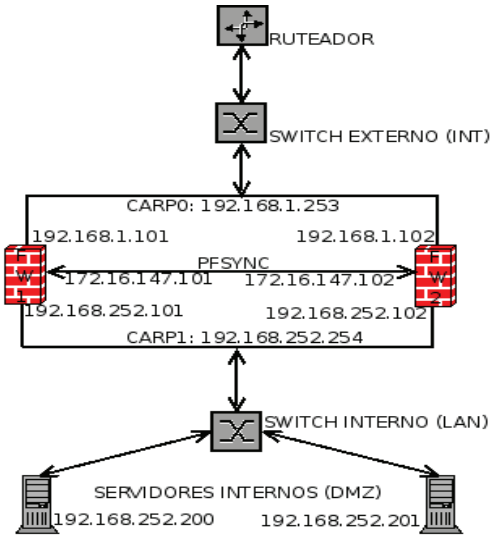


Figura 4.1

De esta manera vemos(*figura 4.1*) que levantaremos 2 IPS por medio de CARP para nuestra comunicación tanto al mundo exterior por ejemplo Internet y al interior que será nuestra DMZ en este caso, ya que en dicho segmento estarán nuestros servidores, también se levantará una interfaz de cada firewall para la interacción del estado de paquetes por medio del protocolo PFSYNC. Obsérvese cómo las direcciones IP reales de cada interfaz de nuestros firewalls son del mismo segmento que las levantadas por medio del protocolo CARP aunque esto no es necesario y en el caso de nuestra LAN nuestros servidores también están con direcciones IP del mismo segmento de la CARP interna.

4.5 Implementación.

Cada grupo de redundancias o alta disponibilidad es representado por una CARP o interfaz virtual de red mediante un IP, y la configuración de éstas se hace por medio del comando *ifconfig*, que tiene la siguiente sintaxis:

> *ifconfig carpN vhid vhid [pass password] [carpdev carpdev] \ [advbase N] [advskew N] [state state] ipaddress mask*

donde:

- carpN

  en este caso “carp” es el nombre de la interfaz virtual y “N” es un entero que representa el número de interfaz, por ejemplo (ejemplo carp1).

- vhid

  Esto representa el identificador del host o IP virtual. Este es un número único que es usado para identificar el grupo de máquinas dentro de la configuración de alta redundancia o alta disponibilidad a otros equipos o nodos dentro de la red, los valores aceptados van del 1 al

255.

- pass

  Esta palabra nos indica que posterior a la misma vendrá la contraseña o password.

- password

  Esta es la contraseña de autenticación que es usada para comunicarse con las otras máquinas dentro del grupo de alta disponibilidad o CARP, la única restricción es que este password debe de ser el mismo para todas las máquinas dentro del grupo de alta disponibilidad.

- carpdev

  Este parámetro es opcional y especifica la interfaz de red física del servidor que pertenece a la CARP o grupo de redundancia o alta disponibilidad, por omisión la CARP tratará de determinar que interfaz utilizar, buscando una interfaz física que se encuentre dentro del mismo segmento que la IP y la máscara de subred dadas a la interfaz virtual de la CARP.

- advbase y advskew

  Estos valores determinan el intervalo entre 2 mensajes de aviso a la CARP, dicho intervalo que ésta descrito en segundos, ésta dado por la siguiente formula (advbase + (advskew / 255)); cuando aumentamos el valor del advase que por omisión es de 1 segundo, haremos menos pesado el tráfico de la red pero se hará mayor el tiempo de elección de un nuevo maestro en caso de una falla en el maestro actual.
  Valores más pequeños en el advskew permiten a una máquina o host dentro de la CARP mandar mensajes de aviso más frecuentemente a la misma, incrementando así su probabilidad de convertirse en maestro.

Los valores del parámetro advbase deben de estar entre 1 y 255, los del advskew entre 1 y 254.

- state

  Por medio de este valor podemos forzar una interfaz dentro de la CARP a adoptar un cierto estado, los estados válidos son ***init***, ***backup*** y ***master***.

- ipaddress

  Aquí especificamos la IP virtual que se dará de alta para que sea mantenida por todo el grupo de redundancia o alta disponibilidad, no es necesario que esta IP esté en el mismo rango que las dadas de alta en las interfaces físicas, pero las direcciones IP en cada una de las interfaces físicas dentro del grupo de la CARP sí deben de estar dentro del mismo segmento.

- mask

  Aquí indicamos la máscara de subred de la IP compartida.

Ahora veamos unos parámetros del sistema que es necesario conocer para el funcionamiento de esta configuración.

- net.inet.carp.allow
  Este parámetro define si la máquina o host puede manejar paquetes de la CARP, por omisión este parámetro esta activado.

- net.inet.carp.arpbalance
  Este parámetro es usado para el balanceo de carga, si esta característica es usada para el balanceo de carga, (si esta característica es activada) la CARP genera firmas de la dirección IP fuente de una petición y después las utiliza para seleccionar el host que manejara dicha petición; esto esta deshabilitado por omisión.

- net.inet.carp.log
  Aquí se define si se registran los errores que genere la CARP, este parámetro está desactivado por defecto.

- net.inet.carp.preempt
  Este parámetro permite a las máquinas dentro de la CARP tener la preferencia para convertirse en el maestro ya que esta opción les permite pasar por arriba de los demás hosts, en el caso de que hubiera una falla pondría el valor de 240 en todas las demás interfaces de la CARP menos en ésta, permitiéndole de manera segura convertirse en maestro, esta característica esta deshabilitada por defecto con un valor de 0.

Es necesario hacer unas verificaciones previas en el sistema operativo de cada uno de los firewalls para poder empezar a configurar las CARP, en primera instancia verificaremos que los números de IP asignados a las interfaces físicas estén debidamente configurados, esto lo haremos por medio de la salida del comando ifconfig que nos dará una lista de las interfaces de red configuradas en el sistema, lo cual nos dará un resultado como el que sigue:

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
        groups: lo
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x6
pcn0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:0c:29:94:79:e8
        groups: egress
        media: Ethernet autoselect (autoselect)
        inet 192.168.1.101 netmask 0xffffff00 broadcast 192.168.1.255
        inet6 fe80::20c:29ff:fe94:79e8%pcn0 prefixlen 64 scopeid 0x1
pcn1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:0c:29:94:79:f2
        media: Ethernet autoselect (autoselect)
        inet 172.16.147.101 netmask 0xffffff00 broadcast 172.16.147.255
        inet6 fe80::20c:29ff:fe94:79f2%pcn1 prefixlen 64 scopeid 0x2
pcn2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:0c:29:94:79:fc
        media: Ethernet autoselect (autoselect)
        inet 192.168.252.101 netmask 0xffffff00 broadcast 192.168.252.255
        inet6 fe80::20c:29ff:fe94:79fc%pcn2 prefixlen 64 scopeid 0x3
```

Figura 4.2

como vemos están las 3 interfaces que vamos a utilizar (pcn0, pcn1 y pcn2), además de una cuarta (lo0) que es el ***loopback*** o interfaz interna (*figura 4.2*), donde se puede apreciar que los números de IP asignados a cada una son los correctos, de esta misma manera tendremos que revisar los de nuestro segundo firewall.

Ahora tendremos que habilitar el ***forward*** en nuestros equipos para que nos permita rutear paquetes dentro de nuestras CARP, esto se hace mediante el comando siguiente.

```
# sysctl net.inet.ip.forwarding=1
net.inet.ip.forwarding: 0 -> 1
# _
```

Figura 4.3

esto nos habilitará de manera temporal esta opción dentro del sistema(figura *4.3*), la cual es necesaria para poder mandar o redireccionar paquetes que lleguen del exterior hacia máquinas que estén dentro de la LAN, para hacer un cambio definitivo, tendremos que editar el archivo sysclt.conf dentro de la carpeta */etc*, por medio del editor de texto *vi* (*figura 4.4*).

```
# vi /etc/sysctl.conf _
```

Figura 4.4

ya dentro del editor hay que quitar el comentario (#) a la línea que indica la habilitación de esta característica (*figura 4.5*).

```
net.inet.ip.forwarding=1        # 1=Permit forwarding (routing) of IPv4 packets
```

Figura 4.5

Es obligatorio hacer los pasos de revisión anteriores deben de hacerse en los 2 firewalls.

Visto lo anterior ahora especificaremos los parámetros con que configuraremos nuestros firewalls para la alta disponibilidad, todos quedaran descritos en la siguiente tabla (*tabla 4.5*).

	FW1	FW2	CARP
INTERNET	192.168.1.101	192.168.1.102	192.168.1.253
LAN (DMZ)	192.168.252.101	192.168.252.102	192.168.252.254
PFSYNC	172.16.147.101	172.16.147.102	

Tabla 4.5

Teniendo los datos anteriores, procederemos a crear las 2 CARP dentro de nuestro primer firewall llamado "*fw1*", bajo las siguientes instrucciones.

```
#ifconfig carp0 create
#ifconfig carp0 192.168.1.253 netmask 255.255.255.0 vhid 1 pass hola advbase 1 advskew 0
#ifconfig carp1 create
#ifconfig carp1 192.168.192.254 netmask 255.255.255.0 vhid 2 pass hola2 advbase 1 advskew 0
```

de la misma manera creamos las CARP en el segundo firewall conocido como "*fw2*"

```
#ifconfig carp0 create
#ifconfig carp0 192.168.1.253 netmask 255.255.255.0 vhid 1 pass hola advbase 1 advskew 100
#ifconfig carp1 create
#ifconfig carp1 192.168.192.254 netmask 255.255.255.0 vhid 2 pass hola2 advbase 1 advskew 100
```

Antes de continuar, es prudente comentar que en las anteriores dos instrucciones no se presentaron las capturas de pantalla del sistema por que dichas instrucciones sobrepasaban el ancho de pantalla y no podían presentarse completas, de esta misma manera se representarán algunas otras que cumplan esta misma característica.

En este paso solamente hemos creado las interfaces virtuales asignándoles una IP, un identificador virtual (*vhid*) para cada CARP, 1 en el caso de la conexión externa o a internet y 2 a nuestra LAN o DMZ, y una contraseña o password (pass) para su autenticación, y en cuanto respecta al primer firewall (*fw1*), lo hemos hecho maestro y siempre que sea posible o que esté activa dicha característica se la damos al ponerle el valor más pequeño en el parámetro advskew, ya que en el *fw1* los mensajes de aviso tendrán un tiempo más corto entre ellos que los del fw2, dicho valor entre

mensajes se da por la formula fw1=(1+0)/255 que sería igual a 1 y el el fw2=(1+100)/255 que es igual a 1.3921.

Ahora para poder configurar el fw1 para que cada vez que esté activo se convierta en maestro, tendremos que modificar el parámetro net.inet.carp.preempt a 1 por medio del siguiente comando (*figura 4.6*).

```
# sysctl net.inet.carp.preempt=1
net.inet.carp.preempt: 0 -> 1
# _
```

Figura 4.6

Ahora para hacer este cambio permanente tenemos que revisar dentro del fw1 que en cada una de las CARP de dicho firewall el parámetro "*advskew*" sea igual a "0" esto dentro de los archivos *hostname.carp0* y *hostname.carp1* para que queden de la siguiente manera.

```
#cat /etc/hostname.carp0
inet 192.168.1.253 255.255.255.0 192.168.1.255 vhid 1 pass hola advbase 1 advskew 0
#cat /etc/hostname.carp1
inet 192.168.252.254 255.255.255.0 192.168.252.255 vhid 2 pass hola2 advbase 1 advskew 0
```

también debemos de agregar la una línea que diga *net.inet.carp.preempt=1* dentro del archivo con ruta */etc/sysctl.conf* (*Figura 4.7*) para que quede de la siguiente manera.

```
#       $OpenBSD: sysctl.conf,v 1.40 2006/01/28 18:22:43 brad Exp $
#
# This file contains a list of sysctl options the user wants set at
# boot time.  See sysctl(3) and sysctl(8) for more information on
# the many available variables.

#
net.inet.carp.preempt=1_
net.inet.ip.forwarding=1        # 1=Permit forwarding (routing) of IPv4 packets
```

Figura 4.7

ya que estamos seguros de tener estas configuraciones dentro del fw1 ahora dentro del fw2 haremos en la revisión del los archivos hostname.carp0 y hostname.carp1 para cerciorarnos de que en el parámetro "*advskew*" tengan el valor de "100" así su intervalo entre los avisos que manden será más largo que el fw1, de esta manera nos aseguramos que siempre que el fw1 esté activo, éste nunca pase a ser maestro.

```
#cat /etc/hostname.carp0
inet 192.168.1.253 255.255.255.0 192.168.1.255 vhid 1 pass hola advbase 1 advskew 100
#cat /etc/hostname.carp1
inet 192.168.252.254 255.255.255.0 192.168.252.255 vhid 2 pass hola2 advbase 1 advskew 100
```

Después de tener ya nuestras 2 interfaces virtuales funcionando de manera correcta debemos de ver que los paquetes que entran y salen estén sincronizados para que ambos equipos conozcan la misma información sobre el estado de los mismos, esto lo haremos por medio del protocolo "*pfsync*" antes mencionado, este es usado por "*packet filter*" para actualizar y manejar las tablas de estado, por defecto los mensajes de cambio de estado son enviados por la interfaz de sincronización usando paquetes IP multicast o multidifusión, el protocolo usado es el IP numero 240 y el grupo o segmento de difusión es 224.0.0.240, y en consecuencia de esta sincronización de estado en los paquetes que pasan por nuestros servidores, en caso de una falla del maestro (fw1) nos permitirá que el tráfico de red fluya sin interrupciones por el nuevo maestro (fw2).

Pfsync representa el nombre del supuesto dispositivo por el cual los cambios en las tablas de estado son trasmitidas, excepto las que generan reglas del script de ***packet filter*** marcadas con el prefijo "*no-sync"*, pfsync puede ser configurado con una interfaz física dedicada a la sincronización de paquetes, esto se utiliza para combinar las tablas de estado de varios firewalls. La sincronización física de interfaces puede ser configurada por medio del comando **ifconfig** añadiéndole el parámetro syncdev, y como argumento el nombre de la interfaz seleccionada para este propósito, por esto mismo la recomendación de que este tráfico transite por una red segura, ya que carece de métodos de autenticación o criptografía para protegerlo, por ende se recomienda un switch dedicado a esta interconexión y más seguro aún un cable cruzado entre las tarjetas de red, que sería nuestro caso ya que es lo más seguro por ahora; cabe mencionar que como una alternativa podemos usar el parámetro syncpeer para especificar la dirección IP del firewall con el que vamos a sincronizar, de esta manera el sistema utilizaría dicha dirección en vez del broadcast o difusión

como el destino de los mensajes pfsync y también podemos usar el parámetro enc seguido de un numero que nos representaría una interfaz virtual por la cual el tráfico se mandaría encapsulado por medio de *ipsec* lo cual protegería la comunicación, para poner un ejemplo de esto último en nuestro caso sería de la siguiente manera.

```
ifconfig pfsync0 syncpeer 172.16.147.102 syncdev enc0
```

de esta manera le estamos diciendo que mande el tráfico a la IP 172.16.147.102 encapsulada por la interfaz enc0.

En nuestro caso como las máquinas se conectarán por medio de un cable cruzado, no es muy necesaria la encriptación, lo que podríamos hacer sería decirle a pfsync por cual interfaz física queremos que mande el tráfico para que no utilice la difusión y sea directamente por medio de dicha interfaz minimizando el tráfico, por lo que proseguimos con el siguiente comando (*figura 4.8*), esto es dentro del fw1.

```
# ifconfig pfsync0 syncdev pcn1
```

Figura 4.8

y para que este cambio sea definitivo tendremos que editar el archivo /etc/hostname.pfsync0 y agregar el siguiente código (*figura 4.9*).

```
# cat hostname.pfsync0
up syncdev pcn1
```

Figura 4.9

De esta misma manera realizamos estos 2 pasos en el fw2(*Figura 4.10 y 4.11*).

```
# ifconfig pfsync0 syncdev pcn1
```

Figura 4.10

```
# cat hostname.pfsync0
up syncdev pcn1
```

figura 4.11

ahora ya tenemos configuradas nuestras interfaces pfsync, y con esto ya hemos proporcionado la característica de alta disponibilidad a nuestra arquitectura por lo cual nos abocaremos a crear las reglas de nuestro firewall que es el ultimo paso para tener completa nuestra arquitectura.

5.1 Introducción.

Packet Filter es el equivalente a **iptables** en todos los sistemas Linux, esto es que dicho paquete de software o programa es el encargado de filtrar el tráfico TCP/IP y llevar a cabo la Traducción de Direcciones de Red o ***NAT*** por las siglas en inglés de "*Network Address Translation*", también es capaz de normalizar y acondicionar el tráfico TCP/IP y de proveer control del ancho de banda y la priorización de paquetes TCP/IP. *Packet Filter* ha formado parte del núcleo GENERIC del sistema OpenBSD desde la versión 3.0 del sistema. El creador original de Packet Filter es Daniel Hartmeier, y en el momento actual lo mantiene y desarrolla Daniel junto con el resto del equipo de desarrolladores de OpenBSD.

Dejando un poco atrás la historia y datos generales de Packet Filter, necesitamos conocer antes que nada la manera de controlarlo, primeramente tendremos que habilitarlo dentro del sistema esto se puede hacer mediante el siguiente comando pfctl -e y para deshabilitarlo con el comando pfctl -d, pero en nuestro caso para habilitarlo de manera definitiva necesitamos editar el archivo /etc/rc.conf.

```
# set the following to "YES" to turn them on
rwhod=NO
nfs_server=NO           # see sysctl.conf for nfs client configuration
lockd=NO
amd=NO
pf=YES_                 # Packet filter / NAT
portmap=NO              # Note: inetd(8) rpc services need portmap too
inetd=YES               # almost always needed
check_quotas=YES        # NO may be desirable in some YP environments
```

Figura 5.1

dentro de dicho archivo encontraremos una sección donde aparece el parámetro pf, por defecto dicho parámetro viene deshabilitado por medio de la palabra "NO", tendiéndola que cambiar a "YES" como lo indica en la figura 5.1, ahora que ya sabemos como habilitar Packet Filter dentro de nuestro sistema operativo necesitamos conocer algunos comandos para manipularlo, a continuación mostraremos en la tabla 5.1 con el comando pfctl y algunos parámetros del mismo que nos servirán para esta situación, para una lista completa de opciones de este comando podremos consultar la página del manual del mismo.

COMANDO	DESCRIPCIÓN
pfctl -f /etc/pf.conf	Carga las reglas que están en el archivo pf.conf.
pfctl -nf /etc/pf.conf	Analiza las reglas del archivo pf.conf pero no las carga.
pfctl -Nf /etc/pf.conf	Carga solamente las reglas “NAT” del archivo pf.conf.
pfctl -Rf /etc/pf.conf	Carga solamente las reglas de filtrado del archivo pf.conf.
pfctl -sn	Muestra en pantalla las reglas “NAT” que en ese momento están activas.
pfctl -sr	Muestra en pantalla las reglas de filtrado que en ese momento están activas.
pfctl -ss	Muestra en pantalla las tablas de estado que en ese momento están activas en el sistema.
pfctl -si	Muestra las estadísticas y contadores de filtrado del sistema.
pfctl -sa	Muestra todo lo posible sobre el estado de Packet Filter en el sistema.

Tabla 5.1

Ahora sabemos que Packet Filter por defecto carga las reglas del archivo pf.conf que esta dentro del directorio /etc, pero necesitamos describir un poco de la estructura de dicho archivo para poder comprender como trabaja Packet Filter y saber interpretarlo, en este sentido solamente diremos que dicho fichero consta de siete partes o secciones, las cuales describiremos a continuación.

- Macros

  Variables definidas por el usuario que pueden contener direcciones IP, nombres de interfaces, etc.

- Tablas

  Una estructura que se utiliza para contener listas de direcciones IP.

- Opciones

  Varias opciones para el control del funcionamiento de PF.

- Normalización o Scrub

  Reprocesamiento de paquetes para su normalización y desfragmentación.

- Formación de Colas

  Provee control del ancho de banda y priorización de paquetes.

- Traducción de Direcciones

  Controla la Traducción de Direcciones de Red (NAT) y el redireccionamiento de paquetes.

- Reglas de Filtrado

  Permite el filtrado selectivo o el bloqueo de paquetes según van pasando a través de cualquiera de las interfaces de red.

Es necesario mencionar que no siempre el archivo pf.conf deberá de tener todas las secciones o partes dentro de él, esto depende mucho del objetivo que persigamos dentro de nuestro firewall y la experiencia del la persona que lo configure, así mismo las secciones deben de aparecer en el orden en el que las describimos anteriormente con excepción de macros y tablas, también podemos mencionar que como políticas de lectura, Packet Filter no interpreta las líneas en blanco y las líneas que comiencen con el carácter “#” las tomará como comentarios, en el apartado 5.4 tenemos un ejemplo completo de un script pf.conf.

5.2 Servicios.

Nuestro objetivo primario es el brindar alta disponibilidad a nuestros firewalls pero un firewall debe de tener una razón de ser o algo que proteger dicho de otra manera algo a lo que le pueda proporcionar dicha propiedad, que en nuestro caso serán los servicios que ofrecen nuestros servidores que estén por detrás de los firewalls, en nuestro caso como lo mencionamos anteriormente los servicios que prestaremos son.

- Servidor de Paginas Web.
- Servicio de Shell Seguro.
- Servicio de Correo Electrónico

Por lo que abocaremos a packet filter a proteger dichos servicios.
En este contexto podemos agregar que se decidieron dichos servicios por que son los más comunes dentro de las entidades públicas o privadas que presten un servicio por medio de internet. Es prudente comentar que no se verán aquí cuestiones del sistema operativo ni instalación de servicios que correrá nuestro servidor interno, dentro de la DMZ, abarcando solamente algunos tips de seguridad para protegerlos por sí mismos, y esto se verá en el capítulo siguiente.

5.3 Explicación de las reglas.

Dado lo anterior empezaremos a describir cada una de las reglas a implementar según los objetivos que tenemos para que quede claro la sintaxis y el objeto de la misma, después se mostrara como queda el script de Packet Filter de manera integral.

Cabe mencionar que el impacto que tiene CARP y PFSYNC dentro de Packet Filert es mínimo, ya que no influye en el funcionamiento del mismo, por ende sólo tenemos que dejar pasar estos protocolos sin añadirles nada para que funcionen a la perfección, en el apartado 5.4 se muestra un ejemplo.

En nuestro script no utilizaremos las 7 secciones que comprenden un archivo de reglas de Packet Filter, usaremos solo 6 de ellas dejando fuera la formación de colas, ya que para nosotros no es necesario controlar el ancho de banda por causa de algún servicio ya que no es nuestra intención

dentro de esta tesis y por ende tampoco priorizaremos paquetes, la política por defecto de nuestro firewall será la de cerrar todo por defecto y solamente abrir lo necesario para cumplir nuestros objetivos. Dicho lo anterior pasaremos a la construcción de nuestras reglas.

- Secciones de Macros y Listas.

  Aquí pondremos todas nuestras interfaces lógicas y físicas agregándoles un alias para su mejor manejo dentro del script, de la misma manera definiremos a nuestros servidores dentro de la DMZ y los puertos de los servicio que presten, los tipos de paquetes ICMP que dejaremos pasar además de definir nuestras redes internas, cabe mencionar que no separamos estas 2 secciones ya que ambas se emplean para definir tópicos de nuestra arquitectura ya sea de manera individual o por grupo.

- Sección de Opciones.

  Dentro de las opciones de nuestro firewall debemos poner la política a seguir que en nuestro caso será la de denegar todo y guardaremos en bitácora todo lo que pase por nuestra interfaz externa.

- Sección de Normalización.

  Habilitaremos la normalización todos los paquetes con el con el objetivo de que no existan ambigüedades de interpretación en el destino final del paquete y realiza el reensamblaje de paquetes fragmentados, protegiendo a algunos sistemas operativos de ciertos tipos de ataques y bloqueando los paquetes TCP que lleven una combinación no válida del indicador de TCP.

- Sección de Traducción de Direcciones.

  En esta sección debemos de redireccionar los servicios que ofrecemos para que lleguen al servidor que los ostenta dentro de la DMZ.

- Sección de Reglas de Filtrado.

  En esta sección lo primero es definir la política por defecto que en este caso es negar o cerrar todo, ahora iremos definiendo lo que esta permitido dándole el respectivo permiso o paso, de ésta a lo que se le dará permiso será al tráfico interno o en la interfaz Loopback, a los Protocolos PFSYNC Y CARP en las interfaces en las cuales trabajan, dejaremos pasar y redireccionaremos todo el tráfico referente a los servicios que prestaremos que son el WWW, SSH y MAIL además de filtrar y permitir el tráfico referente a la resolución de nombres o DNS y a la de ICMP.

5.4 Script final.

De esta manera podemos crear nuestras reglas cubriendo todas las necesidades de esta tesis, así el scritp completo quedaría como se describe a continuación.

```
#############################################################################################
                                        INICIO
#############################################################################################
#SECCION DE MARCOS Y LISTAS#
#Definición de Interfaces
ext_if  = pcn0                                  #Interfaz externa
int_if  = pcn1                                  # Interfaz DMZ
pfs_if  = pcn2                                  #Interfaz PFSYNC
carp_if = "{ carp0, carp1 }"                    #Interfaces CARP
```

```
#Definición de servidores
mail_srv = "servidor.midominio.com"                    #Correo
web_srv  = "{ servidor.midominio.com }""               #Web
dns_srv  = "{ servidor.midominio.com }"                #DNS
ssh_srv  = “{servidor.midominio.com}”                  #SSH
#Definición de puertos
mail_ports = "{ smtp, imap, imaps }"                   #Mail
web_ports  = "{ www, https }"                          #Web
#Tipos de ICMP Permitidos
icmp_types = "{ echoreq, timex, paramprob, unreach code needfrag }"
Definicion de mis redes privadas
priv_nets = "{ 127.0.0.0/8, 172.16.147.0/8, 192.168.0.0/16 }"

#############################################################################################
# SECCIÓN DE OPCIONES#
#Opciones por defecto
set block-policy drop                                  #Bloquear todo
set loginterface $ext_if                               #Bitácora ext.

#############################################################################################
#SECCIÓN DE NORMALIZACIÓN#
#Normalizamos el trafico
scrub in all                                           #Normalizar todo

#############################################################################################
#SECCIÓN DE TRADUCCIÓN DE DIRECCIONES#
# Nateamos las conexiones salientes
nat on $ext_if from $int_if to any -> $ext_if
#Redireccionamos el servicios Web
rdr on $ext_if inet proto tcp from any to $carp_if port $web_ports -> $web_srv
#Redireccionamos el servicio de Mail
rdr on $ext_if inet proto tcp from any to $carp_if port $mail_ports -> $mail_srv
```

```
#############################################################################################
#SECCIÓN DE FILTRADO
#Política por defecto
block all                                                    #Denegar
pass quick on lo0 all                                        #Loopback
pass quick on $pfs_if proto pfsync                           #PFSYNC
pass quick on { $int_if, $ext_if } proto carp keep state     #CARP
#Permitir conexiones a SSH
pass in on $ext_if proto tcp to ($ext_if) port ssh keep state
#Servidor de Mail
pass in  on $ext_if inet proto tcp from any to $mail_srv port $mail_ports \ flags S/SA keep state
pass out on $int_if inet proto tcp from any to $mail_srv port $mail_ports \ flags S/SA keep state
pass in  on $int_if inet proto tcp from $mail_srv to any port smtp \ flags S/SA keep state
pass out on $ext_if inet proto tcp from $ext_if to any port smtp   \ flags S/SA modulate state
# Servidor Web
pass in  on $ext_if inet proto tcp from any to $web_srv port $web_ports \ flags S/SA synproxy state
pass out on $int_if inet proto tcp from any to $web_srv port $web_ports \ flags S/SA keep state
# Protocolo ICMP
pass in  inet proto icmp all icmp-type $icmp_types keep state
pass out inet proto icmp all keep state
# Consultas al servidor DNS
pass in  on $int_if inet proto { tcp, udp } from $int_if to $dns_srv \ port domain keep state
pass out on $ext_if inet proto { tcp, udp } from $ext_if to $dns_srv \ port domain keep state
# Permitir trafico hacia el servidor interno
pass in  on $int_if inet proto tcp from $int_fw to any port $web_ports \ flags S/SA keep state
pass out on $ext_if inet proto tcp from $ext_if to any port $web_ports \ flags S/SA modulate state

#############################################################################################
                                        FINAL
#############################################################################################
```

Como comentarios, dentro de la sección de filtrado debemos de dar permiso a los servidores internos que se conecten al exterior para permitir la actualización de los sistemas operativos en línea, en nuestro caso esto lo dejaremos abierto para no mencionar algún otro sistema operativo en especial, y dentro de la sección de traducción de direcciones en su apartado de redireccionamiento a los servicios internos podríamos habilitar el balanceo de carga mediante el método *round-robin* en el cual se envían a un mismo servidor web las conexiones de un mismo origen mientras existan estados mencionando a esa conexión; una vez que estas conexiones expiren el comportamiento del direccionamiento también expira, la próxima vez que vuelva a establecerse una conexión desde el mismo origen los paquetes serán redirigidos al siguiente servidor Web en el *round-robin*, ésto nos sirve cuando tengamos varios servidores web y la carga de peticiones a nuestro sitio sea de alto impacto y se implementa poniendo el siguiente parámetro al final del direccionamiento.

\ round-robin sticky-address

al final de la línea en donde redireccionamos los paquetes a nuestro servidor en la sección de traducción de direcciones. Por ultimo indicamos que el parámetro.

synproxy state

El cual colocamos en la sección de filtrado de paquetes para dejar pasar el tráfico hacia nuestro servidor web nos permite proporcionarle una seguridad extra a nuestro servidor web, la anterior protección se da cuando un cliente inicia una conexión TCP a un servidor, Packet Filter pasa los paquetes del saludo inicial "***handshake***" entre los dos extremos según llegan. Sin embargo, Packet Filter también puede hacer de proxy para el saludo inicial. Con el modo proxy, Packet Filter completará el saludo inicial con el cliente, iniciará un saludo inicial con el servidor, y pasará los paquetes entre los dos. La ventaja de este proceso es que no se enviará ningún paquete al servidor antes de que el cliente complete el saludo inicial. Esto elimina la amenaza de que desbordamientos TCP SYN falseados puedan afectar al servidor, debido a que una conexión de un cliente falseado no podrá completar el saludo inicial.
Por otra parte también tenemos que Packet Filter tiene la funcionalidad de mantenimiento de estados o inspección completa de estados en los paquetes, esta funcionalidad la usamos en nuestro scritp en la sección de filtrado con el parámetro.

keep state

La inspección del estado se refiere a la capacidad de Packet Filter de llevar un seguimiento del estado, o del progreso, de una conexión de red. Almacenando información sobre cada conexión en una tabla de estado, Packet Filter puede determinar rápidamente si un paquete que está pasando a través del cortafuegos pertenece a una conexión ya establecida. Si es así, se le permite pasar a través del cortafuegos sin tener que pasar a través de la evaluación del grupo de reglas.

Por otra parte el mantenimiento del estado tiene muchas ventajas, entre otras que los grupos de reglas son más simples y se obtiene un rendimiento más alto del filtrado de paquetes. Packet Filter puede hacer que los paquetes que vayan en cualquier dirección concuerden con entradas en la tabla de estado, lo que quiere decir que no es necesario escribir reglas de filtrado que permitan el paso del tráfico de vuelta. Y, como los paquetes que concuerdan con conexiones "***stateful***" no pasan a través de la evaluación del grupo de reglas, el tiempo que tarda PF en procesarlos puede reducirse considerablemente.

Cuando una regla tiene la opción ***keep state***, el primer paquete que concuerda con ella crea un estado entre el remitente y el destinatario. A partir de ahí, los paquetes que vayan desde el remitente hacia el destinatario no serán los únicos que concuerden con la entrada de estado y que circunvalen la evaluación de las reglas, sino que también lo harán los paquetes de respuesta desde el destinatario hacia el remitente.

La opción de Modulación del Estado, funciona como ***keep state***, con la diferencia que sólo es válida para paquetes TCP. Con ***modulate state***, el ISN de las conexiones salientes es aleatorio. Esta opción es útil para proteger conexiones que hayan sido iniciadas por ciertos sistemas operativos que realizan un pobre trabajo al escoger ISNs. A partir de OpenBSD 3.5, la opción ***modulate state*** puede usar usada en aquellas reglas que especifican protocolos diferentes de TCP, esta opción la usamos también en nuestro script en la sección de filtrado con el parámetro.

modulate state

también hacemos mención en nuestro script del siguiente parámetro.

flags S/SA

Esto hace referencia a los indicadores o banderas de los paquetes, la concordancia de paquetes TCP basada en indicadores es algo que se suele usar para filtrar paquetes TCP que estén intentando abrir una nueva conexión. Aquí se puede ver una lista de indicadores TCP y sus significados.

- F : FIN – Finish.
  Indica al otro extremo que la aplicación ya no tiene más datos para enviar. Se utiliza para solicitar el cierre de la conexión actual.

- S : SYN – Synchronize.
  Sincronización de los números de secuencia. Se utiliza al crear una conexión para indicar al otro extremo cual va a ser el primer número de secuencia con el que va a comenzar a transmitir (veremos que no tiene porqué ser el cero).

- R : RST – Reset.
  Interrupción de la conexión actual.
- P : PUSH – Push.
  La aplicación ha solicitado una operación *push* (enviar los datos existentes en la memoria temporal sin esperar a completar el segmento).

- A : ACK – Acknowledgement.
  El campo *Número de acuse de recibo* contiene información válida, es decir, el segmento actual lleva un ACK. Observemos que un mismo segmento puede transportar los datos de un sentido y las confirmaciones del otro sentido de la comunicación.

- U : URG – Urgent.
  El campo Puntero de urgencia contiene información válida.

Por último en esta misma sección hacemos referencia a el siguiente parámetro.

port domain

Esto lo único que hace es substituir el numero 53 que es el asignado para prestar el servicio de traducción de nombres “DNS”.

No vamos a profundizar más en estos conceptos por que se saldría del los objetivos de esta tesis pero con esto es más que suficiente para llegar a una compresión de los que esta pasando y aclarar la sintaxis de cada una de las reglas puestas en nuestro firewall.

6.1 Seguridad de los servicios.

Prácticamente ya tenemos trabajando nuestro firewall y le hemos dado la característica de Alta Disponibilidad y aunque no hemos dicho nada sobre el sistema operativo en el cual estarán funcionando los servidores dentro de nuestra DMZ, es por deducción que será un BSD, UNIX o LINUX por los servicios mencionados, no está por demás mencionar algunos consejos sobre como mejorar la seguridad en los mismos para darle un plus a nuestra arquitectura, esto es de cierta manera importante por que aunque tenemos una arquitectura de alta disponibilidad para los servicios y un filtrado de paquetes es necesario que no sean lo menos vulnerables por ellos mismos, ahora daremos dichos consejos para cada uno de los tres, pero empezamos por uno general.

Consejo de Seguridad para los 3 servicios.

- Mantener siempre los paquetes o programas que presten dichos servicios en la última versión estable para nuestro sistema operativo.

- Revisar progresivamente las bitácoras de cada servicio, estas se encuentran dentro del directorio /var/log y los nombres de los archivos cambian según el servicio que queramos revisar.

Para el servicio de SSH (Secure Shell) o Shell seguro.

- No permitir la entrada a el usuario “root”o administrador.

- Cambiar la autenticación al servicio de contraseña a llave criptografica.

- Cambiar el puerto por defecto de escucha al servicio.

- No permitir contraseñas en blanco.

- Restringir el servicio a los usuarios y/o grupos que sea necesario solamente.

Para el servicio Web

- Utilizar siempre que intercambiemos información importante el protocolo https.
- Verificar los permisos de manera que se evite la escritura y ejecución de archivos lo más posible, una medida muy buena para lo anterior es implementar este servicio en un entorno chroot.
- tener un buen control de acceso en cuanto a las páginas restringidas en nuestro sitio.
- No permitiendo la indexación de directorios públicos.
- Bajar el valor de tiempo de vida y respuesta de los paquetes (TTL).

Para el servicio Mail

- No trabajar con usuarios dados de alta directamente en el sistema, es mejor utilizar una base de datos para manejarlos.
- Instalar algún verificador de virus en nuestro sistema de correo.
- Instalar un antispam para el correo basura.
- Evitar el renvio o relay de correo.

Estas no son todas las medidas de seguridad que podríamos tomar pero sí las más comunes y que nos protegería de un mayor número de atacantes o usuarios mal intencionados.

6.2 Plan de contingencia y Recuperación.

Como mencionamos antes la seguridad absoluta no existe así que por más herramientas que implementemos para protegernos debemos tener en cuenta que nos pueden causar algún día un daño a nuestros servicios, por ende tanto el administrador de sistemas como toda aquella persona que tenga algo que ver con la prestación de dichos servicios se tendrá que reunir y estar consciente del peligro que siempre estará presente, de esta manera pensar en lo peor y ver los niveles de impacto que en este sentido pueden sufrir nuestros sistemas.

Por todo lo anterior de debe de crear un plan de contingencia en cuanto se detecte una intrusión o falla dentro de los sistemas, dándole tareas y responsabilidades a cada uno de los que estén involucrados y teniendo siempre a la mano una forma de comunicarse entre ellos.

Esto nos lleva a tener definidas las medidas de chequeo y monitoreo del los sistemas pertinentes para reaccionar de una manera pronta y correcta ante una ataque. Ahora, siendo más pesimistas y pensando en un ataque efectivo o una falla irrecuperable de hardware, tendremos que tener implementada una Política de respaldos totales que nos permita de una manera rápida poder restablecer los servicios de manera similar a la que se venían prestando.

Un ejemplo de normas a implementar para la recuperación de sistemas sería la siguiente.

1. Todo sistema deberá contar con la documentación de los procedimientos de resguardo y recuperación antes de entrar en producción. La misma será controlada por el área responsable de la Seguridad Informática o Administrador de Sistemas para verificar que es clara, completa y contempla como mínimo la recuperación de los siguientes elementos.

    - El reemplazo de los servidores críticos.
    - El sistema operativo y su configuración (parámetros, file systems, particiones, usuarios y grupos, etc.).
    - Las utilerías y paquetes de software necesarios para que la aplicación se ejecute.
    - Los programas que componen la aplicación.
    - Los archivos y/o bases de datos del sistema.
    - Horario de ejecución de la copia de resguardo.

No se pondrá en producción ningún sistema que no cumpla este requerimiento.

- Todas las copias de resguardo deberán estar claramente identificadas, con etiquetas que indiquen como mínimo.
    - Equipo al que pertenecen.
    - Fecha y hora de ejecución.
    - Frecuencia : anual, mensual, semanal y diaria.
    - Número de secuencia o lote.
    - Tipo de backup o respaldo.
    - Nombre del sistema y otros datos necesarios para su fácil reconocimiento.

- Se llevará un registro diario de las cintas en uso indicando al menos.
    - Fecha de ejecución del resguardo.
    - Qué cintas integran el backup de los equipos.
    - Cantidad de veces que se usó la cinta.
    - Lugares asignados para su guarda.

El área responsable de Seguridad Informática revisará periódicamente que se cumpla con este registro en tiempo y forma.

- Todos los procedimientos de respaldo deberán generar un log en el equipo que permita la revisión del resultado de la ejecución, y dentro de lo posible, se realizarán con la opción de verificación de integridad (lectura posterior a la escritura.)

- Los sitios donde se almacenen las copias de resguardo deberán ser físicamente seguros, con los controles físicos y ambientales según normas estándares; los soportes ópticos o magnéticos deben guardarse dentro de un armario o caja de seguridad.

- Se generarán en lo posible 2 copias de resguardo, guardando una de ellas en un edificio diferente al del ámbito de procesamiento, en un lugar que cumpla con los requerimientos mencionados en el punto 5) y a distancia tal que la ocurrencia de cualquier contingencia en uno no afecte al otro. En caso de tener solo una copia esta debe ser llevada fuera del ámbito de procesamiento de la forma anteriormente mencionada.

El traslado de las cintas debe ser realizado por personal debidamente autorizado, utilizando los accesos habilitados para movimiento de insumos.

- Se realizarán copias de resguardo del sistema completo de acuerdo a lo indicado en la frecuencia asignada a cada aplicación o sistema, previendo la conservación de estos backups por el período de tiempo también estipulado previamente conforme a la criticidad de la información.

En el caso de utilizar backups incrementales se deberá tener en cuenta lo siguiente.

- Se documentará la identificación de secuencia de los backups incrementales.

- Deberán existir controles para prevenir la carga de cintas en una secuencia equivocada.

- Se realizará un backup del sistema completo cada siete (7) días corridos.

- Se efectuarán pruebas de recuperación de las copias de resguardo al menos una vez cada treinta (30) días corridos. Estas pruebas servirán para constatar que se puedan obtener correctamente los datos grabados en la cinta al momento de ser necesarios, de forma de garantizar su propósito.

Las pruebas se deberán formalizar en un acta escrita y firmada por el responsable del sector técnico y el encargado de realizar la recuperación.
Eventualmente el área responsable de la Seguridad Informática presenciará las pruebas y firmará el acta.

- Los servidores críticos deberán contar con RAIDs de discos, a los efectos de que la información sensible no se vea afectada por potenciales desperfectos en los discos.

- Para el caso de aplicaciones críticas se implementarán técnicas de replicación automática, por hardware o software, de forma tal que si el equipo/base de datos principal deje de funcionar el equipo/base de datos espejo tome el control inmediatamente.

- Los períodos de retención de la información histórica son los siguientes.
    - Fuentes y base de datos: perpetuo
    - Actividades de los usuarios y pistas de auditoría: TRES (3) años.

- El resguardo de la información histórica se realizará utilizando soportes ópticos de referencia no reutilizables (CDs, etc).

- Los procedimientos de generación y grabación de estos archivos serán automáticos, a fin de evitar su modificación.

Cabe mencionar que estas normas no son tajantes y muy generales, siempre podrán ser mejoradas de acuerdo a los tiempos de creación, políticas de la empresa y las particularidades de nuestro entorno de trabajo. Esto es con el objetivo de que nos demos una idea de como implementar un plan de recuperación y contingencia.

6.3 Conclusiones.

De esta manera llegamos al final de este trabajo donde nos hemos dado cuenta de que la seguridad informática es muy extensa e interesante, que siempre hay que buscar alternativas en lo que se refiere a tecnología para poder llegar a cubrir una necesidad que tengamos y que la alta disponibilidad es un concepto que tenemos que tener presente en la implementación de servicios de alto impacto o demanda dentro de nuestro entorno.

La seguridad informática ya no es cosa de bancos o grandes empresas, tenemos que defender nuestro derecho a la privacidad, es cierto que la Internet es una ventana al mundo para nosotros pero es también una puerta abierta para los demás, por lo que tendremos que estar preparados y conocer de estos temas para poder estar más tranquilos.

Los sistemas de código abierto o libre son una buena alternativa para nuestras aplicaciones, tanto que los utilizan desde programadores principiantes hasta grandes empresas de diferente índole, la única objeción que ponen algunos para no utilizarlo es la falta de un respaldo seguro en cuanto a asesoría, esto se ha ido suplantando por grupos, listas y foros de discusión sobre diferentes temas

además de que el numero de posibles asesores es mucho más grande y especializado que en el software propietario ya que estas personas tienen acceso al código de dicho software.

Como lo mencionamos anteriormente, la única solución de alta disponibilidad conocida para firewalls es la de Cisco que no aporta mejorías ante esta y si implica un gasto económico ya que es una solución comercial.

- ANSI

  Instituto Nacional Estadounidense de Estándares.

- ANTISPAM

  Programa informático que nos protege contra mensajes no solicitados, habitualmente de tipo publicitario, enviados en cantidades masivas. Aunque se puede hacer por distintas vías, la más utilizada entre el público en general es la basada en el correo electrónico.

- ANTISPYWARE

  Programa informático que nos protege contra aplicaciones que recopilan información sobre una persona u organización sin su conocimiento.

- ANTIVIRUS

  Programa informático que nos protege contra aplicaciones de ordenador que puede infectar otros programas modificándolos para incluir una copia de sí mismo, esto puede implicar un mal funcionamiento de nuestro sistema.

- ARP

  Son las siglas en inglés de Address Resolution Protocol (Protocolo de resolución de direcciones), Es un protocolo de nivel de red responsable de encontrar la dirección física que corresponde a una determinada dirección lógica dentro de una red de cómputo.

- ARPANET

  Sus siglas en ingles son Advanced Research Projects Agency Network fue creada por encargo del departamento de defensa de los Estados Unidos como medio de comunicación para los diferentes organismos físicamente distantes del país. El primer nodo se creó en la Universidad de California.

- BACKUP

  Operación dentro de un sistema que consiste en guardar en un medio extraíble (para poder guardarlo en lugar seguro) la información sensible referida a un sistema.

- BASE DE DATOS

  Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

- BUG

  Es el resultado de un fallo de programación introducida en el proceso de creación de programas de ordenador o computadora.

- CD

  Es un soporte digital óptico utilizado para almacenar cualquier tipo de información, en español conocido como disco compacto.

- CHECKLIST

  Es una lista previa de acciones tomadas por alguna persona para corroborar o revisar que todo esta listo para proceder con algún procedimiento posterior.

- CONMUTADO

  Consisten en un conjunto de nodos interconectados entre sí, a través de medios de transmisión (cables), formando la mayoría de las veces una topología mallada, donde la información se transfiere encaminándola del nodo de origen al nodo destino mediante conmutación entre nodos intermedios.

- CONFIDENCIALIDAD

  Se define como la capacidad de proporcionar acceso a usuarios autorizados, y negarlo a no autorizados. Por desgracia en castellano esta palabra tiene la connotación de “secreto”, y muchas veces se confunde confidencialidad con secreto.

- COPYRIGHT

  El derecho de autor es una forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales entre otras.

- CRACKER

  Persona que se dedica a romper los sistemas de seguridad de las redes con la intención de ingresar para hacer daño.

- CROSS-COMPILE

  Funcionalidad de compilación en algún software que esta específicamente diseñado para alguna plataforma especifica y por medio de esta poder migrar el código fuente a otra plataforma.

- DAEMON O DEMONIO

  Clase especial de programa que corre en segundo plano en vez de ser controlado directamente por el usuario, vale decir, que funciona sin tener relación con una terminal o consola y, consecuentemente, sin interactuar con un humano.

- DESFRAGMENTACIÓN

  Es la utilidad que reordena todos los archivos dentro de una unidad de disco duro de manera que queden de forma físicamente continua, sin estar divididos.

- DIFUSIÓN

  En este tipo de redes no existen nodos intermedios de conmutación; todos los nodos comparten un medio de transmisión común, por el que la información transmitida por un nodo es conocida por todos los demás.

- DISKLABEL

  una tabla de particiones propia en los sistemas BSD, que se utiliza, precisamente para describir las particiones que tienen.

- DISPONIBILIDAD

  Se define como la capacidad de acceder a información o utilizar un servicio siempre que lo necesitemos.

- DMESG

  Propiedad en los sistemas UNIX, LINUX o BSD que permite desplegar y guardar los mensajes que manda el núcleo del sistema al arrancar, por medio del cual nos podemos dar cuenta de algún error en el sistema.

- DMZ

  En seguridad informática, una zona desmilitarizada o red perimetral es una red local o subred que se ubica entre la red interna de una organización y una red externa.

- DNS

  Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio.

- Dual-Homed Gateway

  Algunos firewalls son implementados sin un screening router poniendo un sistema entre la red privada y la Internet, y deshabilitando el recorrido del TCP/IP. El host en la red privada se puede comunicar con el gateway, asi como el host en el Internet, pero el tráfico directo entre las redes es blokeado. Por definición un Dual Homed Gateway es un bastion host.

- ENCRIPTACIÓN

  Proceso mediante el cual cierta información o "texto plano" es cifrado de forma que el resultado sea ilegible a menos que se conozcan los datos necesarios para su interpretación.

- ETHERNET

  Es el nombre de una tecnología de redes de computadoras de área local basada en tramas de datos.

- FDDI (Fiber Distributed Data Interface)

  Es un conjunto de estándares ISO y ANSI para la transmisión de datos en redes de computadoras de área local mediante cable de fibra óptica. Se basa en la arquitectura token ring y permite una comunicación tipo Full Duplex. Dado que puede abastecer a miles de usuarios, una LAN FDDI suele ser empleada como backbone para una red de área amplia.

- FIBRA OPTICA

  Es una guía de ondas en forma de filamento, generalmente de vidrio (en realidad, de polisilicio), aunque también puede ser de materiales plásticos, capaz de guiar una potencia óptica (lumínica), generalmente introducida por un láser.

- FILE SYSTEM

  Es una gran colección de directorios y archivos que guardan todo tipo de información. En sistemas de muchos usuarios se pueden tener cientos o miles de archivos.

- FILTRAJE

  Técnica informática por medio de la cual se puede regular el acceso a diferentes servicios en línea.

- FIREWALL

  Es un dispositivo físico o lógico utilizado en las redes para prevenir algunos tipos de comunicaciones prohibidos por las políticas de red, las cuales se fundamentan en las necesidades del usuario.

- FORWARD

  Capacidad que tienen un equipo de cómputo de reenviar información a través del entre equipos diferentes.

- FREE-AFS

  Sistema de archivos distribuido mundialmente, su característica principal el tener ventajas sobre el manejo de NFS como podrían ser una mejor administración y disponibilidad.

- FTP

  Es uno de los diversos protocolos de la red Internet, concretamente significa Protocolo de Transferencia de Ficheros y es el ideal para transferir grandes bloques de datos por la red.

- GATEWAY

  Es normalmente un equipo informático configurado para dotar a las máquinas de una red local conectadas a él de un acceso hacia una red exterior.

- Gigabit o Gb

  Es una unidad de información o de almacenamiento informático normalmente abreviada como Gbit o a veces Gb, 1 gigabit = 10^9 = 1,000,000,000 bits, que equivalen a 125 megabytes decimales.

- HACKER

  Trasciende a los expertos relacionados con la informática, para también referirse a cualquier profesional que está en la cúspide de la excelencia en su profesión, ya que en la descripción más pura, personalizando lo anterior decimos que es aquella persona que le apasiona el conocimiento, descubrir o aprender nuevas cosas y entender el funcionamiento de éstas.

- HALT

  Comando en algunos sistemas operativos que apaga nuestro sistema parando todos los procesos y servicios que este corriendo así, es sinomimo de apagar el sistema.

- HANDSHAKE

  Procedimiento dentro de los protocolos de red por medio del cual 2 nodos o equipos empiezan la comunicación entre ellos, es como el saludo a una persona antes de comenzar una conversación.

- HARDWARE

  Son los componentes físicos de una computadora.

- HTTP

  Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la Web.

- HTTPS

  Implementación de la versión segura de HTTP, mediante algoritmos de cifrado que encriptan las transacciones Web.

- IEEE

  Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

- IMAGEN ISO

  Es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco compacto (como un CD o un DVD).

- IFCONFIG

  Es un programa disponible en varias versiones del sistema operativo UNIX, que permite configurar o desplegar numerosos parámetros de las interfaces de red.

- INTERFACES, TARJETAS DE RED Ó NIC

  Network Interface Controller, Controlador de Interfaz de Red en español), es una tarjeta de expansión que permite a cualquier dispositivo de cómputo acceder a una red y compartir sus recursos.

- INTEGRIDAD

  Se define como la capacidad de garantizar que una información o mensaje no han sido manipulados.

- INTERNET

  Es una red mundial de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP. Aparece por primera vez en 1960.

- IP

  Es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.

- IPFILTER

  Antiguo filtro de paquetes dentro del sistema operativo OpenBSD que fue dado de baja en el sistema por problemas de licencia con su creador Darren Reed.

- IPSEC

  Es una extensión al protocolo IP que añade cifrado fuerte para permitir servicios de autenticación y cifrado y, de esta manera, asegurar las comunicaciones a través de dicho protocolo.

- IPTABLES

  Es el nombre de la herramienta del “user space” (espacio de usuario) por medio de la cual el administrador crea reglas para filtrado de paquetes y para hacer NAT. Mientras que técnicamente iptables es solamente la herramienta que controla estos componentes dentro del kernel, el nombre “iptables” se utiliza muchas veces para referirse a toda la infraestructura, incluyendo a netfilter, connection tracking y NAT, como también a la herramienta propiamente dicha. iptables es una parte standard de todas las distribuciones Linux actuales.

- IPV4

  Es la versión 4 del Protocolo IP. Esta fue la primera versión del protocolo que se implementó extensamente, y forma la base de Internet.

- IPV6

  Es la versión 6 del Protocolo de Internet, un estándar del nivel de red encargado de dirigir y encaminar los paquetes a través de una red.

- KERBEROS

  Es un protocolo de autentificación de red. Esta diseñado para suministrar una autentificación poderosa para aplicaciones cliente/servidor usando criptografía secret-key.

- KERNEL

  Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuanto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso para el programador.

- LAN

  Significa red local y la definición nos dice que es la interconexión de varios ordenadores y periféricos. Su extensión esta limitada físicamente a un edificio o a un entorno de unos pocos kilómetros. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo en oficinas, fábricas, etc; para compartir recursos e intercambiar datos y aplicaciones. En definitiva, permite que dos o más máquinas se comuniquen.

- LINUX

  Linux es el nombre de un núcleo, uno de los componentes de un sistema operativo, que pretende ser una implementación libre de UNIX, implementando estándares como POSIX ó Single Unix Specification. Para tener un sistema operativo útil, hace falta software adicional, ya sea en forma de herramientas para configurar y administrar el sistema mismo como en aplicaciones a ser usadas por los usuarios. Al paquete formado por el núcleo y software adicional capaz de funcionar se le suele llamar una distribución. Hay una gran variedad de empresas que proveen distribuciones de Linux, con distintas orientaciones, que van desde uso como servidor o estación de trabajo hasta aplicaciones empotradas en las cuales el usuario final quizá ni se entere que hay Linux dentro del producto. Pero el término Linux se ha popularizado para referirse a la distribución misma en detrimento de la correcta significación. Es decir que Linux en voz popular se refiere comúnmente para describir al sistema operativo completo que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros

proyectos/grupos de software (libre o no libre). El núcleo no es parte oficial del proyecto GNU (el cual posee su propio núcleo en desarrollo, llamado Hurd), pero es distribuido bajo los términos de la licencia GNU GPL.

- MAC ADDRESS

  En redes de computadoras la dirección MAC (Media Access Control address) es un identificador hexadecimal de 48 bits que se corresponde de forma única con una tarjeta o interfaz de red. Es individual, cada dispositivo tiene su propia dirección MAC determinada y configurada por el IEEE (los primeros 24 bits) y el fabricante (los 24 bits restantes).

- MULTICAST

  Es el envío de la información en una red a múltiples destinos simultáneamente, usando la estrategia más eficiente para el envío de los mensajes sobre cada enlace de la red sólo una vez y creando copias cuando los enlaces en los destinos se dividen.

- NAT

  Es un estándar creado por la Internet Engineering Task Force (IETF) el cual utiliza una o más direcciones IP para conectar varios computadores a otra red (normalmente a Internet).

- NNTP

  Protocolo de transferencia de noticias. Es el Protocolo de red utilizado por el Usenet internet service. Es un Protocolo de red basado en tiras de textos enviados sobre canales TCP de 7 bit ASCII . Es usado para subir y bajar así como para transferir artículos entre servidores.

- ISO

  Es una organización internacional no gubernamental, compuesta por representantes de los Organismos de Normalización (ONs) nacionales, que produce Normas Internacionales industriales y comerciales. Dichas normas se conocen como normas ISO.

- PARTICIÓN

  Son las divisiones lógicas en un disco duro que permite aplicar el formato lógico de un sistema operativo específico.

- PASSWORD

  Una contraseña o clave, es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso.

- PID

  Identificador de procesos, es un número usado por el kernel del sistema operativo (como el de Unix o el de Windows NT) para identificar un proceso.

- POSIX

  Acrónimo de Portable Operating System Interface, viniendo la X de UNIX con el significado de la herencia de la API (Se traduciría como Sistema Operativo Portable basado en UNIX).

- PROCESADOR

  Es un componente de un sistema o máquina que se encarga de convertir la materia prima de éste y dar un producto que puede ser sometido a otro procesamiento o ser el producto final del sistema o máquina.

- PROPOLICE

  Se trata de unas mejoras en el compilador GCC que hace que los binarios obtenidos estén protegidos contra los ataques de tipo buffer overflow.

- PROTOCOLOS

  conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o automatismos de otro tipo, tales y como dispositivos electrónicos capaces de interactuar en una red.

- PROXY

  Hace referencia a un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la del servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP.

- PUERTO

  Es una forma genérica de denominar a una interfaz por la cual diferentes tipos de datos pueden ser enviados y recibidos. Dicha interfaz puede ser física, o puede ser a nivel software.

- PUNTO A PUNTO

  Son aquellas redes en las que se usa cada canal de datos para comunicar únicamente a 2 nodos, en contraposición a las redes multipunto, en las cuales cada canal de datos se puede usar para comunicarse con diversos nodos.

- RAID

  Es un término inglés que hace referencia a un conjunto de discos redundantes independientes/baratos. Este tipo de dispositivos se utilizan para aumentar la integridad de los datos en los discos, mejorar la tolerancia a los fallos y errores y mejorar el rendimiento.

- REDES

  es un conjunto de computadoras y/o dispositivos conectados entre sí y que comparten información, recurso y servicios, etc.

- RELEASE

  Acrónimo de versión que hace alusión a nuevas mejoras o actualizaciones dentro de un producto, generalmente programas de cómputo o sistemas operativos.

- ROUND-ROBIN

  Es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

- RUTEADOR

  Es un dispositivo hardware o software de interconexión de redes de computadoras que opera en la capa tres (nivel de red) del modelo OSI. Este dispositivo interconecta segmentos de red o redes enteras. Hace pasar paquetes de datos entre redes tomando como base la información de la capa de red.

- Screened Host Gateway

  Posiblemente la configuración de firewall más común es un screened host gateway, es usada utilizando un screening router y un bastion host. Usualmente el bastion host se encuentra en la red privada, y el screening router esta configurado de tal manera que el bastion host es el único sistema de la red privada que puede ser alcanzado desde Internet. Frecuentemente el screening router esta configurado para blokear el tráfico al bastion host en algunos puertos específicos, permitiendo solo que un pequeño numero de servicios se comuniquen con el.

- Screened Subnet Firewall

  En algunas configuraciones Firewall, se crea una subred aislada, y se sitúa entre Internet y la red privada. Tipicamente, esta red es aislada usando screening routers, los cuales pueden implementar varios niveles de filtración. Generalmente una screened subnet es configurada para que ambos, tanto el Internet como la red privada tengan acceso al host en la screened subnet, pero el tráfico que pasa a traves de la subred es bloqueado. Algunas configuraciones de estas subredes tienen un host bastion en la red protegida, en ambos casos se soportan sesiones de terminal interactivas o niveles gateways de aplicación.

- SCREENSHOT

  Una captura de pantalla, es una imagen tomada por una computadora para registrar los elementos visibles en el monitor u otro dispositivo de salida visual.

- SCRIPT

  El guión o archivo de procesamiento por lotes (en inglés script) es un conjunto de instrucciones, sentencias de control, variables y demás elementos de programación generalmente almacenadas en un archivo de texto.

- SCRUB

  Directiva dentro de Packet Filter que permite la normalización de paquetes.

- SCSI

  Del acrónimo inglés Small Computer System Interface es una interfaz estándar para la transferencia de datos entre periféricos en el bus del ordenador (computadora).

- SHA1-HMAC

  Algoritmo Criptográfico de autenticación dentro de una red o servicio.

- SMTP

  Protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos.

- SOFTWARE

  Son todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

- SOFTWARE LIBRE

  Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios.

- SPOOFING

  En términos de seguridad de redes hace referencia al uso de técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación.

  Existen diferentes tipos de spoofing dependiendo de la tecnología a la que nos refiramos, los cuales se describirán más adelante, como el IP spoofing (quizás el más conocido), ARP spoofing, DNS spoofing, Web spoofing o e-mail spoofing, aunque en general se puede englobar dentro de spoofing cualquier tecnología de red susceptible de sufrir suplantaciones de identidad.

- SSH

  Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo el ordenador mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X arrancado.

- SWAP

  El espacio de intercambio es una zona del disco (un fichero o partición) que se usa para guardar las imágenes de los procesos que no han de mantenerse en memoria física.

- SWITCH

  Es un dispositivo de interconexión de redes de computadoras, interconecta dos o más segmentos de red, pasando datos de un segmento a otro, de acuerdo con la dirección MAC de destino de los datagramas en la red.

- SYSTRACE

  Sistema que hace cumplir ciertas políticas en las llamadas de procedimientos dentro del sistema, y si algo tratara de saltárselas emite una alerta para que el usuario tome una decisión al respecto.

- TCP

  Protocolo de control de transmisión, usado en las redes de cómputo para garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

- TELNET

  Sirve para acceder mediante una red a otra máquina, para manejarla como si estuviéramos sentados delante de ella. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

- UDP

  Es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación, ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco sabemos si ha llegado correctamente, ya que no hay confirmación de entrega o de recepción.

- UNDERGROUND

  En informática se aplica a un individuo o grupo que realizan tareas o funciones licitas o ilícitas pero de manera no reconocida y sin responsabilidad de las mismas.

- USUARIO

  Es la persona a la que va destinada un producto informático una vez ha superado las fases de desarrollo correspondientes.

- UTP

  Es un cable de cobre, y por tanto conductor de electricidad, que se utiliza para telecomunicaciones y que consta de uno o más pares, ninguno de los cuales está blindado “Unshielded”. Cada par “Pair” es un conjunto de dos conductores aislados con un recubrimiento plástico; este par se trenza “Twisted” para que la señales transportadas por ambos conductores (de la misma magnitud y sentido contrario) no generen interferencias ni resulten sensibles a emisiones.

- WEB

  Es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador web para extraer elementos de información (llamados "documentos" o "páginas web") de los servidores web (o "sitios") y mostrarlos en la pantalla del usuario.

- WINDOWS

  es el nombre de una familia de programas, y en la actualidad, sistemas operativos privativos desarrollados por la empresa de software Microsoft Corporation. Todos ellos tienen en común el estar basados en una interfaz gráfica de usuario basada en el paradigma de ventanas (de ahí su nombre en inglés) que en un principio se inspiro e imito la interfaz del MacOS de Apple.

- X/OPEN

  Es la denominación de un grupo consorciado de fabricantes y usuarios, cuya tarea es el desarrollo de productos software que cumplan la normativa CAE o ingeniería asistida por ordenador o computadora.

[1] Yanek Korff, Paco Hope. Bruce Potter, Mastering FreeBSD and OpenBSD Security. Editorial Orreilly. Capítulos 3 y 8.

[2] Michael W. Lucas. Absolute OpenBSD. Editorial Orreily. Pag. 56 – 135.

[3] Stephen J. Bigelow. Localización de averías, reparación, mantenimiento y optimización de redes. Editorial Mc Graw Hill. Capítulos 1,2,3 y 5.

[4] Neil Jenkins y Stan Schatt. Redes de área local. Editorial Prentice Hall. Pag. 22-28, 137-139.

[5] J.C. Daccach T. Alta disponibilidad. URL:
http://www.gestiopolis.com/delta/term/TER170.html Fecha de ultima consulta. 23/06/06

[6] Van TI Advanced bussines. Alta disponibilidad. URL:
http://www.vanti.com.mx/con_infra_tec_3.htm Fecha de ultima consulta 23/06/07.

[7] Derek J. Hunt. Creating the Ultimate Home Firewall and Intrusion Detection System.... in under an hour. URL: http://derek.uberh4x0r.org/ Fecha de ultima consulta 15/05/07.

[8] Juan Pedro Paredes. Alta disponibilidad para Linux. Contacto juampe@retemail.es Fecha de ultima consulta 01/02/06.

[9] Jacek Artymiak. Changes in pf: Packet Filter. URL:
http://www.onlamp.com/pub/a/bsd/2003/06/26/ssn_openbsd.html Fecha de ultima consulta 27/05/07

[10] Equipo de desarrollo de OpenBSD. Filtro de paquetes de OpenBSD. URL:
http://www.openbsd.org/faq/pf/es/index.html Fecha de ultima visita 27/06/07.

[11] Lizardo Desilos. Alta disponibilidad en servicios dentro de OpenBSD. URL:
http://www.openbsd.org.mx/lizardo/carp/ Fecha de ultima consulta 05/03/07.

[12] Jason Dixon. Failover Firewalls with OpenBSD and CARP. URL:

http://www.samag.com/documents/s=9658/sam0505e/0505e.htm Fecha de ultima consulta 18/04/07.

[13] Daniele Mazzocchio. Redundant Firewalls with OpenBSD, CARP and PFSYNC. URL: http://www.kernel-panic.it/openbsd/carp/ Fecha de ultima consulta 05/06/07.

[14] Ryan McBride. Firewall Failover with PFSYNC and CARP. URL: http://www.countersiege.com/doc/pfsync-carp/ Fecha de ultima consulta 03/04/07.