ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA SEGURIDAD PARA LA RED DE VOZ Y DATOS DE LA CORPORACIÓN MACHANGARASOFT, UTILIZANDO EL SISTEMA OPERATIVO LINUX

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

JOSÉ ANTONIO ESTRADA JIMÉNEZ

DIRECTOR: ING. FERNADO FLORES

Quito, Mayo de 2007

DECLARACIÓN

Yo, José Antonio Estrada Jiménez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por José Antonio Estrada Jiménez, bajo mi supervisión.

Ing. Fernando Flores
DIRECTOR DE PROYECTO

AGRADECIMIENTO

A mi familia, mis padres, mi abuela, mis hermanos, a todos mis amigos que, como siempre, confiaron en mi más que yo mismo, por estar seguros de que no fallaría, por su aliento, por su abrigo, Gracias.

Al Ing. Fernando Flores por su acertada guía, comprensión y sobre todo por su tiempo.

Finalmente, y de manera muy especial, un agradecimiento a todas las comunidades de *software* libre que defienden y pregonan en esencia la libertad del conocimiento y la independencia tecnológica, porque esa es la vía hacia el desarrollo: el trabajo en comunidad, el conocimiento compartido. Gracias por esa filosofía, gracias por este nuevo modelo de vida.

DEDICATORIA

Por enseñarme a no rendirme nunca y a alcanzar todo lo que soñaba, este prólogo de todo lo que vendrá está enteramente dedicado a mis padres.

ÍNDICE de Contenidos

ÍNDICE de Anexos

ÍNDICE de Figuras

ÍNDICE de Tablas

RESUMEN

El presente proyecto detalla el Diseño e Implementación del prototipo de un sistema de voz sobre IP y de un sistema de seguridad en base a *firewall*, para la Corporación MachángaraSoft. todo esto mediante la utilización del sistema operativo Linux.

El sistema de voz involucra una central telefónica con capacidad para voz sobre IP e integración con la red de datos existente, a través de una gran variedad de aplicaciones y una flexibilidad que cualquier central propietaria podría envidiar. La central telefónica basada en la aplicación de código abierto *Asterisk* tiene capacidad de interacción con redes analógicas como la de Andinatel y con la red Internet.

El sistema de seguridad consiste en la generación de un *script* de aplicación de reglas de *firewall* mediante la herramienta *iptables* de Linux. Este *script* define un conjunto de políticas que regulan el acceso a aplicaciones a Internet de acuerdo a los estrictos requerimientos de la corporación.

A continuación, un breve resumen del contenido de cada capítulo:

Capítulo 1: Fundamentos Teóricos

En este capítulo se realiza un estudio de la voz sobre IP, detallando las tecnologías, protocolos y métodos de conexión de este tipo de transmisión de voz. Además se estudian algunos lineamientos para la determinación del hardware necesario para implementarse en una solución, dependiendo de los requerimientos de una organización.

Asimismo, en lo referente al sistema de seguridad, se realiza un estudio de la tecnología de *firewall* y los mecanismos de seguridad de una red para su conexión a Internet y la prestación de servicios DNS, correo, Proxy, VPN, etc.

Se parte del estudio del sistema operativo Linux y los componentes que posee para el manejo, filtrado y control del tráfico de información, el conjunto de protocolos TCP/IP que nos permiten la administración, el control y las herramientas disponibles para ello en el sistema operativo Linux.

Capítulo 2: Diseño y configuración del Prototipo de Voz y datos sobre Linux

En este capítulo se detalla todo el proceso de diseño y configuración del prototipo, tomando en cuenta los requerimientos de conexión a nivel de voz y seguridad de datos en Internet de la corporación. A partir de este capítulo se explica paso a paso el proceso de configuración de las aplicaciones de voz y del *script* que aplicará las reglas de filtrado de tráfico desde y hacia Internet.

Capítulo 3: Implementación, Pruebas y Costos del Prototipo

A partir del proceso de configuración, se detalla en este capítulo el proceso de puesta en funcionamiento del prototipo, de manera que puedan ejecutarse las aplicaciones y enrutamiento de voz junto con el adecuado filtrado de tráfico de datos desde la *intranet* hacia Internet y viceversa. Esto permitirá realizar una serie de pruebas de cada una de las aplicaciones, tanto de voz y datos, que son requeridas por la corporación.

Finalmente se determina brevemente el costo aproximado del prototipo.

Capítulo 4: Conclusiones y Recomendaciones

En el último capítulo se discute la forma en la que el prototipo puede ser mejorado, así como también se hacen algunas recomendaciones referentes a su uso y configuración adecuados.

PRESENTACIÓN

La filosofía de código abierto, ampliamente representada por el sistema operativo Linux, está causando una revolución que ya no solamente se limita al tema estricto de la codificación, sino que sienta un precedente y deja muy en claro lo que el desarrollo comunitario es capaz de alcanzar.

Las herramientas que se basan en Linux tienen la ventaja de usar estándares, están ampliamente documentadas, y todas aquellas que implementan soluciones propietarias, lo hacen de forma mucho más flexible, escalable y, dependiendo de la experiencia en su manejo, son incluso más veloces.

El momento en que estas soluciones involucran, en este caso, filtrado de tráfico y telefonía IP, ofrecen a las pequeñas y medianas empresas la posibilidad de utilizar la tecnología que está disponible a un costo mucho menor que el que tendrían que pagar si utilizaran soluciones y hardware propietarios, además sin perjuicio de capacidad, rendimiento ni variedad de aplicaciones.

El desarrollo tecnológico se dirige siempre hacia la reducción de costos, tanto en cuanto a equipos como a servicios. Sin embargo en lo referente a telecomunicaciones, todos estos aspectos han sido manejados por grandes monopolios que han impedido que esta realidad sea posible. Ahora gracias a esta nueva filosofía de desarrollo, el desarrollo en sí mismo, depende de todos nosotros.

En ese sentido se dirige este proyecto, como una alternativa que nos da libertad, y esa misma libertad hace que sea más accesible para todos.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

TELEFONÍA IP

1.1 INTRODUCCIÓN A LA TELEFONÍA IP

No cabe duda de que el fenómeno del Internet ha desencadenado una verdadera revolución a nivel tecnológico, esencialmente en el ámbito de las comunicaciones. La masificación de los servicios que se prestan a través de la red de redes ha provocado la disminución de los costos y, desde luego, el aumento de la capacidad de transmisión y recepción de datos. El acceso a una conexión de Internet de banda ancha se vuelve más común en los hogares y empresas, permitiendo la integración de estas entidades al conglomerado tecnológico interconectado alrededor del mundo; y como consecuencia se desarrolla un proceso paulatino y acelerado de globalización y estandarización de los métodos de comunicación.

Asimismo, la creciente implementación de redes IP, el desarrollo de técnicas avanzadas de digitalización de la voz, control de tráfico en tiempo real, para brindar calidad de servicio en aplicaciones que lo requieran y el estudio de nuevos protocolos y estándares para este efecto han permitido el desarrollo del ambiente propicio para la telefonía IP, como producto directo de la convergencia tecnológica, consolidando aplicaciones (como las de voz y datos) y servicios sobre una infraestructura común.

Por otro lado, surge también, como consecuencia directa de la globalización del conocimiento a través del Internet, el movimiento de software libre que promueve la libertad de ejecutar, modificar, estudiar, copiar y mejorar las aplicaciones de software, como una cultura de desarrollo tecnológico en base a una comunidad cuyos horizontes llegarían incluso a tocar el ámbito de las telecomunicaciones a través del proyecto Asterisk, para el manejo de la transmisión de voz sobre IP.

Asterisk es una aplicación de software libre de central telefónica mediante la cual se pueden conectar varios teléfonos para que se comuniquen entre sí, para realizar llamadas a través de un proveedor de VoIP o incluso mediante la red de

telefonía pública (PSTN). La capacidad de adaptarse a las necesidades de las organizaciones y la flexibilidad del sistema de configuración de Asterisk permite romper con el paradigma impuesto por casi la totalidad de empresas, que ofrecen servicios de telecomunicaciones, cuya calidad se condiciona a la adquisición de licencias costosas, sin tomar en cuenta los equipos, el servicio técnico y de capacitación.

1.2 TELEFONÍA ANÁLOGA

Análogo se refiere a la transmisión de información electrónica que se logra sumando señales de frecuencia o amplitud variable a una onda portadora de frecuencia dada. La tecnología análoga es usada comúnmente por las transmisiones broadcast tradicionales tales como radio y televisión. El término "análogo" se puede atribuir a la similitud entre la fluctuación de la voz humana y la modulación "análoga" o comparable de una onda portadora. La voz humana ocupa un rango de frecuencia de 20 Hz a 20 KHz en donde la mayor cantidad de energía se concentra entre los 300 y 3300 Hz. Cabe resaltar que tanto la voz humana como la onda portadora modulada muestran períodos de baja o ninguna actividad seguidos de períodos de actividad.

1.2.1 OPERACIÓN DEL SISTEMA TELEFÓNICO BÁSICO

Los sistemas telefónicos usan líneas analógicas conmutadas para proveer comunicaciones de voz, convirtiendo ondas de sonido en señales eléctricas. Cuando alguien levanta el auricular para hacer una llamada, un circuito se cierra, energizando un relé, lo que provoca que un dispositivo llamado *buscador de línea (line searcher)* encuentre una línea abierta. Luego, una conexión desde el consumidor a la oficina central de telefonía se establece y se genera un tono de marcado. El buscador de línea entonces prepara el equipo de conmutación de la compañía telefónica para recibir un número telefónico.

La señal que se transmite fluye a través de la bobina de voz en el aparato telefónico de la persona que recibe la llamada. La bobina sujeta al parlante en el receptor vibra en respuesta a la señal para reproducir ondas de sonido y la persona que escucha al teléfono oye la voz reproducida de la otra persona.

El propósito de la Red Telefónica Pública Conmutada (PSTN) es establecer y mantener conexiones de audio entre dos puntos finales. Podemos percibir la limitación inicial de poder únicamente transportar un único tipo de información a través del par de cobre.

1.2.2 SEÑALIZACIÓN ANALÓGICA

El término *señalización* se refiere a las señales específicas transmitidas sobre los circuitos telefónicos que son usadas para enviar información de control de la línea, datos de usuario y conversaciones de voz.

En el bucle local de la PSTN, un circuito abierto sin corriente fluyendo indica una condición de *on-hook* (el auricular está en su nicho). La condición *off-hook* (el teléfono descolgado) es indicada por un circuito cerrado con corriente fluyendo continuamente.

1.2.2.1 Señalización E&M

Este tipo de señalización es el método más utilizado para proveer troncalización analógica. *La señalización E&M provee estados de señalización que indican las condiciones on-hook y off-hook*, disminuyendo la probabilidad de que una línea troncal de dos vías sea tomada simultáneamente en ambos puntos finales.

Utiliza un par extra de cables en las líneas troncales local y remota, uno de ellos designado como *E* mientras que el otro es designado como M. El cable E recibe señales mientras que el M las transmite.

1.2.2.2. Loop Start y Ground Start

Hay dos métodos básicos para detectar la condición de *off-hook* de un abonado y de iniciar una serie de tareas como *notificación de ocupado* o mecanismos de tarifación: las llamadas líneas *loop Start* y *ground Start*.

Las líneas ***Loop Start*** señalizan la condición *off-hook* completando un circuito en el teléfono, esto es, cuando un suscriptor levanta el auricular, fluye corriente a través de los contactos del gancho, energizando el relé de la línea. Luego, un conjunto de contactos de relé se cierran, indicando que el suscriptor desea servicio. Un conmutador de la CO llamado *line finder* (buscador de línea) provee

tono de marcado y conecta al suscriptor con el equipo de conmutación en un circuito disponible.

Las líneas ***Ground Start*** son usadas en el bucle local para conectar las PBX a la CO. La corriente fluye a través de la mitad del relé de la línea como resultado de aterrizar el cable *ring.* Cuando una PBX necesita conectividad, el *line finder* de la CO provee tono de marcado hacia el equipo de conmutación de la CO. Cuando el tono de marcado es detectado, el contacto *ground start* se abre y el resto de la línea es instantáneamente activada.

1.2.2.3 Señalización por Pulso

Cuando usamos marcación por pulsos, un circuito se abre y cierra el mismo número de veces que el número que marcamos, a excepción del número 0, que cuando se marca el conmutador se abre y se cierra 10 veces.

1.2.2.4 Señalización por Tono (*Dual-Tone Multifrequency*)

La señalización por tono utiliza una señal audible de dos tonos para indicar un número único. Dos tonos audibles son colocados en la línea para indicar la tecla presionada.

1.3 TELEFONÍA DIGITAL

Uno de los retos primarios cuando se transmite señales analógicas es que hay una gran variedad de elementos que pueden interferir con esas señales, causando bajo volumen, estática y muchos otros efectos no deseados. En lugar de intentar preservar una forma de onda analógica sobre distancias que pueden llegar a ser miles de millas, por qué no simplemente medir las características del sonido original y enviar esa información al punto final. La onda original no llegará, pero toda la información necesaria para reconstruirla lo hará.

Este es el principio de todo el audio digital (incluyendo telefonía): *muestrear las características de la forma de onda original, almacenar la información medida, y enviar esos datos al punto final*. Entonces, en el extremo receptor, podemos usar la información transmitida para generar una señal de audio completamente nueva

que tenga las mismas características que la original. La reproducción puede llegar a ser tan buena que el oído humano no alcanza a determinar la diferencia.

1.3.1 MODULACIÓN POR IMPULSOS CODIFICADOS (PCM)

Hay muchas formas de codificar audio digitalmente, pero el método más común (y el usado en los sistemas telefónicos) es conocido como *Modulación por Impulsos Codificados.* La modulación PCM nos permite transformar una señal analógica en una secuencia de bits, es decir en una señal digital.

El principio de PCM es que la amplitud de la forma de onda analógica es muestreada a intervalos específicos de modo que pueda ser recreada posteriormente. La cantidad de detalle que es capturada es dependiente de la resolución de bit de cada muestra y de cuan frecuentemente son tomadas las muestras. Una resolución de bit mayor y una tasa de muestreo más alta proveerán mayor exactitud, pero se requerirá más ancho de banda para transmitir esta información más detallada.

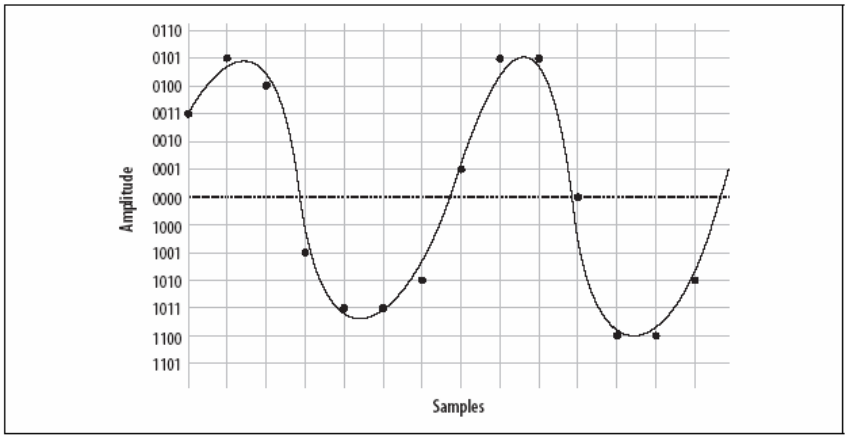


FIGURA 1.1 MUESTREO DE LA ONDA SINUSOIDAL USANDO 4 BITS.

1.3.2 EL TEOREMA DE NYQUIST

De modo que, cuánto grado de muestreo es necesario. El teorema de Nyquist afirma que: “Cuando se muestrea una señal, la frecuencia de muestreo debe ser mayor que el doble del ancho de banda de la señal de entrada de modo que se pueda reconstruir la original perfectamente de la versión muestreada”.

Esto significa que, para codificar de manera exacta una señal analógica, se debe muestrearla a dos veces el ancho de banda total que se desea reproducir. Dado

que la red telefónica no portará frecuencias menores que 300 Hz ni mayores que 4000 Hz, una frecuencia de muestreo de 8000 muestras por segundo será suficiente para reproducir cualquier frecuencia dentro del ancho de banda de un teléfono analógico.

1.4 RED DE CONMUTACIÓN DE PAQUETES

A mediados de los 90, el perfomance de las redes mejoró hasta el punto en el que fue posible enviar un flujo de información en tiempo real a través de la conexión de red. Debido a que el flujo de información es cortado en segmentos que luego son empaquetados, esas conexiones se denominan basadas en paquetes.

El reto es, por supuesto, enviar miles de estos paquetes entre dos puntos finales, asegurando que los paquetes arriben en el mismo orden en el que fueron enviados, en menos de 300 milisegundos, sin ninguna pérdida. Esta es la esencia de la voz sobre IP.

1.5 VOZ SOBRE IP

La tecnología de VoIP ha ido paulatinamente introduciéndose en nuestras actividades de comunicación gracias a la difusión del Internet y a las múltiples posibilidades que se ofrecen al momento de acceder a la red. Inicialmente como un accesorio más de aplicaciones de mensajería como MSN Messenger y luego más orientado a la transmisión de voz como con *Skype*, esta modalidad de telefonía (si es que aún se le puede llamar así) amenaza incluso los intereses de grandes compañías de telefonía tradicional debido al alcance, economía y funcionalidad que ofrece a sus usuarios.

Como su nombre lo indica, la VoIP (*Voice over IP* o Voz sobre IP*)* permite el transporte de voz en forma de paquetes IP a través de cualquier red que funcione en base a IP.

Es de esperarse también que esta tecnología de transporte de voz conjugue la transmisión de voz con la de datos, pues se basa en reducir la información de voz a eso, a datos. Esto se logra encapsulando la voz en paquetes para transportarlos en una red de datos.

En telefonía IP la comunicación se realiza mediante el envío de paquetes que consisten en la información de voz codificada. Esto significa que se dispone de un control mayor sobre la comunicación, lo que permite, por ejemplo, aprovechar de mejor manera los recursos a través de la supresión de los paquetes que contengan silencios.

Asimismo, al tener control sobre los paquetes, es posible generar un número mayor de servicios que hagan de la comunicación de voz una experiencia muy interesante.

Para poder transmitir la voz en forma de paquetes, la señal original debe pasar por un proceso de digitalización en señales PCM a través de un codificador/decodificador de voz para luego ser comprimidas mediante un algoritmo que segmenta estas señales para ser transmitidas en forma de paquetes. En el extremo receptor se realiza un proceso inverso para recuperar la información vocal.

1.5.1 SEÑALIZACIÓN DE VOIP Y PROTOCOLOS DE TRANSPORTE DE VOZ

Los protocolos de transporte que son colectivamente llamados “Internet” no se diseñaron originalmente para la transmisión de flujos en tiempo real sino que se suponía que los puntos finales de la comunicación resolverían la pérdida de paquetes esperando un poco más por su arribo, pidiendo retransmisión o, en ciertos casos, asumiendo como correcta la información pese a la pérdida. En una conversación típica de voz estos mecanismos no serían adecuados, pues nuestras conversaciones no se adaptan apropiadamente a la pérdida de letras o palabras, ni a un retardo apreciable entre el transmisor y el receptor.

El problema con la transmisión de voz basada en paquetes proviene del hecho de que la forma en que hablamos es completamente incompatible con la forma en la que IP transporta información.

El mecanismo para mantener una conexión de VoIP generalmente involucra un conjunto de transacciones de señalización entre los puntos finales (y pasarelas entre ellos), lo que implica dos flujos de medios de comunicación persistente (uno para cada dirección) que transportan una conversación y para manejar este proceso existen una gran cantidad de protocolos y recomendaciones.

1.5.2 PROTOCOLO H.323

H.323 es quizás el estándar más importante que soporta la tecnología de voz empaquetada. Fue diseñado con el objetivo fundamental de *proveer a los usuarios de un servicio de tele-conferencia capaz de transmitir voz, video y datos a través de redes de conmutación de paquetes.*

1.5.2.1 Terminales H.323.- Son los puntos finales que proveen la interfaz del usuario hacia la red, generalmente un teléfono IP.

1.5.2.2 Gateways H.323.- Considerados como puntos terminales y permiten la comunicación de una red H.323 con otro tipo de redes, especialmente la PSTN o sistemas PBX, proveyendo servicios de traducción y control de llamada entre dos tipos de redes diferentes.

1.5.2.3 Gatekeepers H.323.- Desarrollan funciones de control de llamada y administración de políticas para terminales H.323 registrados. Los servicios fundamentales que provee son: traducción de direcciones, control de admisión, control de ancho de banda y administración de zona.

1.5.2.4 Unidades de control Multipunto (MCU).- Administran los recursos de conferencia para tres o más puntos finales los que establecen una conexión con este dispositivo.

1.5.2.5 Pila de protocolos H.323

H.323 es un conjunto de protocolos que trabajan juntos para brindar funcionalidades de llamada extremo a extremo en una red convergente. Sin embargo depende mucho de los servicios de otros protocolos como TCP, IP, UDP y RTP. Los protocolos que conforman el estándar H.323 son RAS (Registro, Admisión y Estado), H.245 y H.225.

1.5.2.5.1 Protocolo de Internet (IP)

IP provee un esquema de direccionamiento jerárquico para H.323, en donde cada punto terminal, gateway, gatekeeper, y MCU debe tener una única dirección IP válida.

1.5.2.5.2 Protocolo de Control de Transmisión (TCP)

TCP es responsable de proveer un mecanismo fiable de control de transmisión sobre el protocolo no fiable IP. En un ambiente H.323, TCP permite establecer la configuración de conexión inicial entre los terminales y gateways/gatekeepers.

1.5.2.5.3 Protocolo UDP (User Datagram Protocol)

UDP es un protocolo no orientado a conexión y sin secuenciamiento, que *sacrifica confiabilidad por velocidad*. Es más rápido que TCP y por esta razón que UDP es utilizado para el transporte de información durante una llamada de VoIP. Si se pierde un paquete de voz por alguna razón, UDP es indiferente a ello, puesto que entregar el paquete perdido fuera de sincronización entorpecerá el proceso de comunicación en lugar de ayudar.

1.5.2.5.4 H.225

Este protocolo provee establecimiento y control de llamada con toda la señalización necesaria para conectar dos puntos terminales H.323. La recomendación ITU Q.931 provee los medios para establecer, mantener y terminar las conexiones de red a través de ISDN.

1.5.2.5.5 H.245

La señalización de control H.245 es usada para negociar las capacidades y uso del canal. Intercambia mensajes de control extremo a extremo, manejando la operación del punto final H.323. La información que transporta en los mensajes de control se relacionan con: intercambio de información de capacidades, apertura y cierre de canales lógicos para transporte de datos, mensajes de control de flujo, comandos en general.

Luego del establecimiento de la llamada, toda la comunicación se cursa sobre canales lógicos y H.245 define los procedimientos para mapear estos canales lógicos, en donde el canal lógico 0 es para control H.245 y múltiples canales lógicos de varios tipos como vídeo, datos, voz, son permitidos por una única llamada.

1.5.2.5.6 Registro, Administración y Estado (RAS)

RAS es un protocolo usado entre puntos finales (terminales y gateways) y gatekeepers, que permite realizar actividades de registro, control de admisión, cambios de ancho de banda y estado, para desligar estos terminales de los gatekeepers. RAS utiliza el puerto 1719 para UDP.

1.5.2.5.7 Protocolo de Transporte en Tiempo Real (RTP – Real-Time Transport Protocol)

RTP provee funciones de transporte de red extremo a extremo adecuado para aplicaciones que requieren transmitir datos en tiempo real como audio, video, datos de simulación, sobre servicios de red multicast o unicast. Este protocolo es utilizado para transportar datos mediante UDP y no direcciona reservación de recursos ni garantiza calidad de servicio para servicios en tiempo real.

El transporte de datos es aumentado por un protocolo de control (RTCP) para permitir el monitoreo de la entrega de la información de una manera escalable para grandes redes multicast y para proveer un control mínimo y funcionalidad de identificación. RTP y RTCP se han diseñado para ser independientes de las capas subyacentes de red y de transporte.

RTCP provee de transporte de control a RTP y realimentación de la calidad de la distribución de datos.

1.5.2.5.8 Códecs

Códec es la abstracción de codificador/decodificador y se utiliza no solamente por parte del protocolo H.323 sino por todos los protocolos de VoIP para definir el grado de los algoritmos de compresión y descompresión que se usarán para el

transporte de voz o vídeo a través de una red convergente. H.323 soporta la mayoría de códecs de voz y video incluyendo:

- **G.7XX** Series ITU de códecs de audio (G.711, G.723, G729).
- **H.26X** Series ITU de códecs de vídeo (H.261, H.263).

1.5.2.6 Etapas de una llamada H.323

Los procedimientos de conexión involucrados cuando se crea una llamada H.323 pueden agruparse en 5 etapas:

a. Descubrimiento y registro
b. Configuración de llamada
c. Flujo de señalización de llamada
d. Flujo de datos y flujo de control de datos
e. Terminación de llamada

1.5.2.7 H.323 y NAT

El método más fácil de manejar esta situación con el tráfico de voz a través de H.323 es abriendo los puertos apropiados a través del dispositivo NAT al cliente interno. Para recibir llamadas se necesitará siempre abrir el puerto TCP 1720 para el cliente, además de que se deberá abrir los puertos UDP para el flujo de información RTP y para los flujos de control RTCP. Los puertos que deberán abrirse dependerán fundamentalmente del cliente que se utilice.

1.5.3 PROTOCOLO DE INICIALIZACIÓN DE SESIONES (*SIP* -SESSION INITIATION PROTOCOL)

1.5.3.1 Descripción

SIP es un protocolo de señalización relativamente simple, utilizado para telefonía y videoconferencia a través de Internet.

SIP, inicialmente definido en el RFC 2543 para luego ser revisado en el RFC 3261, 3263, 3853 y 4320, está basado en SMTP *(Simple Mail Transport Protocol)* y en HTTP *(Hypertext Transfer Protocol),* compartiendo con este último algunas de sus características de diseño, al ser legible para humanos y seguir una estructura en base a *petición/respuesta*.

SIP es un protocolo de capa de aplicación independiente del protocolo de paquetes subyacente (TCP, UDP, ATM, X.25) que está basado en una arquitectura cliente/servidor en la que el cliente inicia las llamadas y el servidor las contesta. Al ajustarse a estándares existentes de Internet basados en texto (como SMTP y HTTP), la detección de errores y depuración de la red se facilitan enormemente ya que el protocolo puede leerse claramente sin necesidad de decodificar la información como en el caso de protocolos no basados en texto, como H.323. Por otro lado, dado que es abierto y basado en estándares, SIP es ampliamente soportado y no dependiente de un único equipo o implementación.

SIP es un protocolo más nuevo que H.323 y como consecuencia no posee su madurez. Sin embargo, debido a su simplicidad, escalabilidad, modularidad y facilidad con la que se integra con otras aplicaciones, SIP es atractivo para su uso en arquitecturas de voz en base a paquetes. Algunas de las características clave que SIP ofrece son:

- Resolución de direcciones, mapeo de nombres y redirección de llamadas.
- Descubrimiento dinámico de capacidades de información del punto terminal, mediante el uso de SDP (Session Description Protocol).
- Descubrimiento dinámico de disponibilidad del punto terminal.
- Origen de sesión y administración entre el servidor y los puntos terminales.

1.5.3.2 Componentes SIP

El sistema SIP contiene dos componentes fundamentales: agentes usuario y servidores de red, en donde un agente de usuario (UA) es un punto terminal SIP que hace y recibe llamadas a través de este protocolo.

- El cliente es nombrado como *cliente de agente usuario (UAC -user agent client)* y es usado para iniciar las solicitudes SIP.
- El servidor recibe el nombre de *servidor de agente usuario* (*UAS -user agent server*) y recibe las solicitudes del UAC y devuelve respuestas para el usuario.

1.5.3.3 Mensajes SIP

SIP trabaja bajo una premisa simple de operación cliente/servidor en donde los clientes o puntos terminales son identificados a través de una dirección única. Estas direcciones vienen en un formato muy similar al de una dirección de correo electrónico: usuario@dominio.com.

Al especificar una dirección SIP se debe tomar en cuenta los siguientes puntos:

- Las direcciones SIP tienen siempre el siguiente formato: usuario@host o usuario@servidor
- El usuario puede ser: nombre, teléfono, o un número común.
- El host puede ser: un dominio, o una dirección IP.

Los usuarios o clientes se registran con servidores SIP para proveer de información de ubicación de contacto.

SIP utiliza dos tipos de mensajes para conexión de llamada y control: *solicitudes* y *respuestas,* que se definen de la siguiente manera:

INVITE Que permite que un usuario inicie una llamada y cuyos campos de cabecera incluyen:

- *Direcciones* del llamante y de la persona que es llamada.
- *Asunto* de la llamada
- *Prioridad* de la llamada
- *Solicitud de enrutamiento* de llamada
- *Preferencias del llamante* para la ubicación del usuario
- Características deseadas de la respuesta

BYE Utilizado para terminar una conexión entre dos usuarios.

REGISTER Transporta información de ubicación a un servidor SIP, permitiendo a un usuario comunicarle al servidor cómo mapear una dirección entrante en una dirección saliente que alcanzará al usuario.

ACK Confirma el intercambio de mensajes fiables.

PRACK ACK provisional

CANCEL Cancela cualquier búsqueda pendiente pero no termina una llamada que ya ha sido aceptada.

OPTIONS Se encarga de solicitar información acerca de las capacidades del punto terminal que es llamado.

Los siguientes tipos de respuesta SIP son usados:

SIP 1xx Respuestas Informativas (como 180, Timbrando)

SIP 2xx Respuestas de éxito (como 200, OK)

SIP 3xx Respuestas de redirección (como 302, temporalmente trasladado)

SIP 4xx Respuestas de falla del cliente (como 404, Usuario no encontrado)

SIP 5xx Respuestas de falla de servidor

SIP 6xx Respuestas de falla global

1.5.3.4 Pila de Protocolos SIP

En un ambiente SIP, las llamadas de voz son transportadas mediante el uso de RTP. SIP depende de estándares adicionales para brindar funciones de señalización y control.

- El conjunto de protocolos G.7xx provee funciones básicas de CODEC.
- El Protocolo de Transporte en Tiempo Real (RTP) permite al receptor detectar pérdida de paquetes, al igual que provee información de temporización que le permite al receptor compensar el desfase en el tiempo de arribo variable de los paquetes (es decir, el jitter).
- El Protocolo de Control de Tiempo real (RTCP) acompaña a RTP y se encarga de obtener información del estado de la red y de calidad de servicio. RTCP provee la función de administrar la entrega de paquetes en tiempo real.
- El Protocolo SCPT *(Stream Control Transmission Protocol)* provee un mecanismo de transporte entre entidades SIP que intercambian una gran cantidad de mensajes. Debido a que SIP es independiente del transporte, el soporte de SCTP es un proceso relativamente sencillo y sus especificaciones están en el RFC 3286.

- El protocolo RSVP (*Resource Reservation Protocol*) está definido en el RFC 2205 y es usado para brindar calidad de servicio. En este protocolo la solicitud es procesada por cada nodo junto con el camino de la sesión y los dispositivos reservan los recursos necesarios para soportar el flujo de información de la sesión.
- El protocolo SDP *(Session Description Protocol)* es un protocolo definido en el RFC 2327 y provee anuncios de sesión e invitaciones de sesión dentro de un ambiente multimedia, permitiendo a los receptores del anuncio de sesión participar en dicha sesión.
- El protocolo SAP (*Session Announcement Protocol*) es otro protocolo para anunciar conferencias y sesiones multicast.

1.5.4 PROTOCOLO MGCP [1](MEDIA GATEWAY CONTROL PROTOCOL)

Este protocolo proviene también del IETF y aunque el despliegue MGCP está más extendido de lo que se podría pensar, pierde terreno rápidamente frente a protocolos como SIP e IAX. Es igualmente un estándar para conferencia multimedia sobre el protocolo IP.

A diferencia de la arquitectura SIP, donde la inteligencia reside en los puntos terminales, el modelo MGCP asume que un servidor posee las facultades para implementar servicios avanzados. Como resultado de esto, los teléfonos MGCP poseen características técnicas mínimas. Este aspecto es beneficioso para portadores, puesto que permite la entrega de servicios avanzados a través de puntos terminales de bajo costo.

MGCP provee capacidades para:

- Determinar la ubicación del punto terminal objetivo.
- Determinar las capacidades de comunicación del punto terminal objetivo a través del Protocolo de Descripción de Sesión (SDP).
- Determinar la disponibilidad del punto terminal objetivo.
- Establecer una sesión entre el punto terminal destino y el origen.

[1] FONG, Paul; KNIPP, Eric, "Configuring Cisco Voice Over IP". Syngress Media Publishing Inc. USA 2002, Segunda Edición.

1.5.5 CONTROLADOR DE GATEWAY DE COMUNICACIONES (*MEGACO* – MEDIA GATEWAY CONTROLLER)

Este controlador dirige las relaciones entre un gateway y un agente de llamada; sin embargo, a diferencia de MGCP, MegaCo soporta un rango más grande de redes, incluyendo ATM, haciendo al protocolo aplicable a un rango mucho más amplio de gateways de comunicaciones.

Luego de que el estándar MGCP fue aprobado, el IETF sometió propuestas y mejoras adicionales, combinando varios estándares en una arquitectura única dando como resultado el desarrollo conjunto de la ITU y el IETF.

1.5.6 IAX[2] (*INTER-ASTERISK EXCHANGE PROTOCOL*)

1.5.6.1 Descripción

IAX es un protocolo P2P (*peer-to-peer*) para comunicación y señalización, por lo que el componente de señalización es más análogo a SIP que a MGCP (que es un protocolo de control de llamada maestro/esclavo).

La información de secuenciamiento y flujo de comunicaciones están incluidos en las tramas IAX. El transporte de comunicaciones no utiliza el protocolo RTP.

Básicamente multiplexa señalización y múltiples flujos de comunicaciones sobre un único puerto asociado, entre dos *hosts* de Internet.

IAX es un protocolo de transporte, muy parecido a SIP, desarrollado por la empresa Digium con el objetivo de comunicar servidores Asterisk. IAX utiliza un único puerto UDP, el 4569, tanto para señalización como para el flujo de transmisión de comunicaciones, permitiendo una comunicación más sencilla detrás de un firewall-NAT. Puede ser utilizado también entre servidores y clientes que manejen del mismo modo el protocolo IAX.

Actualmente IAX se refiere a IAX2, que es la segunda versión del protocolo IAX.

IAX es un protocolo binario y la razón para este diseño es la *eficiencia de ancho de banda*. Este protocolo fue específicamente optimizado para hacer un uso sumamente eficiente de ancho de banda en llamadas de voz individuales.

---

[2] SPENCER, Marc, "Inter-Asterisk EXchange (IAX) Version 2", Diguim, Inc, Enero 2005. (http://www.cornfed.com/iax.pdf)

IAX tiene también la capacidad de troncalizar múltiples sesiones en un flujo de datos, lo que puede ser una ventaja gigantesca cuando de ancho de banda se trate, especialmente cuando se envíe una gran cantidad de flujos simultáneos hacia un punto remoto. La troncalización permite que múltiples flujos de comunicación (datos) puedan ser presentados por una cabecera de datagrama única, para disminuir el *overhead* asociado con canales individuales. Esto ayuda a disminuir la latencia y a reducir la capacidad de procesamiento y el ancho de banda requeridos, permitiendo al protocolo escalar más fácilmente con un gran número de canales activos entre puntos terminales.

Este protocolo posee la habilidad de autenticar en tres vías: texto plano, *hashing* MD5 e intercambio de claves RSA. No existe encriptación del camino de comunicaciones o de las cabeceras entre los puntos terminales, aunque esto puede solucionarse incluyendo soluciones de VPN (Virtual *Private Network*), dispositivos o software para encriptar el flujo en otra capa de tecnología, lo que requiere que los puntos finales pre-establezcan un método para tener estos túneles configurados y operacionales. En los próximos años, IAX será capaz de encriptar las transmisiones entre extremos a través del intercambio de claves RSA, o el intercambio de claves dinámicas en el establecimiento de la llamada, permitiendo el cambio automático de claves. Esto sería muy útil para crear un enlace seguro en instituciones, por ejemplo, bancarias.

El protocolo IAX2 fue, como se mencionó, deliberadamente diseñado para trabajar tras dispositivos que hacen NAT (*Network Address Translation*). El uso de un único puerto UDP tanto para señalización como para transmisión de comunicaciones mantiene el número de vulnerabilidades del *firewall* al mínimo. Estas consideraciones han ayudado a hacer de IAX uno de los protocolos más fácilmente implementados en redes seguras.

1.5.6.2 Configuración de Llamada

A continuación se describe el flujo básico de mensajes usado para establecer una conexión de una llamada de voz. En este ejemplo el host A inicia una llamada enviando un mensaje NEW al host B; el host B inmediatamente envía de regreso un mensaje ACCEPT, indicando al host A que ha recibido la solicitud y que

empieza a procesarla. El host A envía un mensaje ACK al host B, indicándole que recibió el mensaje ACCEPT. Una vez que el teléfono del host B empieza a timbrar, este host envía hacia A un mensaje RINGING, o sea la señal de timbrado.

El host A envía un mensaje ACK de regreso a B indicándole que recibió el mensaje de timbrado. Finalmente, cuando se levanta el auricular del teléfono del host B, éste envía un mensaje ANSWER (de respuesta) y el establecimiento de llamada se completa. En este punto empieza una transmisión de voz en ambos sentidos entre los dos hosts.

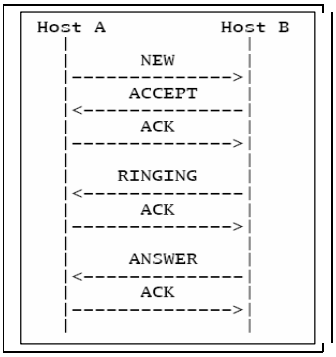


FIGURA 1.2 ESCENARIO DE ESTABLECIMIENTO DE LLAMADA IAX

1.5.6.3 Desconexión de Llamada

El flujo de mensajes, cuando una desconexión de llamada ocurre, podría ser: ell host A inicia la desconexión enviando un mensaje de HANGUP al host B. El host B inmediatamente envía como respuesta un mensaje ACK indicando que recibió la señal de desconexión y que la llamada ha finalizado en el lado del host B.

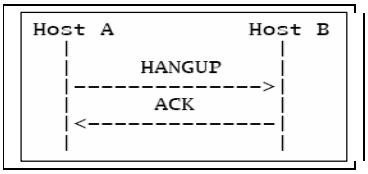


FIGURA 1.3 ESCENARIO DE DESCONEXIÓN DE LLAMADA.

1.5.7 CISCO-SKINNY

Skinny (*SCCP –Skinny Client Control Protocol*) es un protocolo de control propietario de los equipos de VoIP Cisco. Es el protocolo por defecto para los

terminales sobre una PBX *Cisco Call Manager*. Originalmente desarrollado por *Celsius Corporation,* es ahora definido por *Cisco Systems Inc.* como un conjunto de mensajes entre un cliente Skinny y el software con el que funcionan los equipos de central telefónica de Cisco.

1.5.8 CÓDECS[3] (ESTÁNDARES DE CODIFICACIÓN)

Los códecs son generalmente entendidos como modelos matemáticos usados para codificar digitalmente (y comprimir) información de audio analógica. Muchos de estos modelos toman en cuenta la capacidad del cerebro humano para formar una impresión de la información incompleta. Al igual que las ilusiones ópticas, los algoritmos de compresión de voz toman ventaja de nuestra tendencia a interpretar lo que *creemos* que deberíamos oír, en lugar de lo que de hecho oímos. El propósito de varios de los algoritmos de codificación es alcanzar un balance entre eficiencia y calidad.

Originalmente, y aún ahora, el término CODEC se refirió a la expresión COdificador/DECodificador: un dispositivo que hace la conversión entre señales analógicas y digitales; sin embargo, ahora el término parece relacionarse más con COmpresión/DECompresión.

A continuación podemos observar una referencia rápida de los códecs que trabajan para VoIP.

TABLA 1.1 REFERENCIA DE ESTÁNDARES DE CODIFICACIÓN

Códec	**Velocidad de datos (kbps)**	**Requiere Licencia**
G.711	64	No
G.726	16, 24 o 32	No
G.723.1	5.3 o 6.3	Sí
G.729a	8	Sí
GSM	13	No
iLBC	13.3 o 15.2	No
Speex	Variable (de 2.15 a 22.4)	No

[3] http://www.voip-info.org

1.5.8.1 G.711

G.711 es un códec fundamental para la PSTN. De hecho, si nos referimos a PCM con respecto a la red telefónica, es necesario pensar en G.711. Dos métodos de compansión (compresión/expansión) se usan: la ley μ en Norte América y la ley A en el resto del mundo. Cualquiera de estas dos entrega una palabra de 8 bits transmitidos 8000 veces por segundo, lo que indica que deben transmitirse 64000 bits por segundo.

1.5.8.2 G.726

Este códec ha estado en vigencia por algún tiempo (dejó obsoleto a G.721), y es uno de los códecs comprimidos originales. También conocido como Modulación de Pulsos Codificados Diferencial Adaptativa (*ADPCM –Adaptive Diferencial Pulse-Code Modulation*) puede funcionar a diferentes velocidades de transferencia, entre las más comunes están: 16kbps, 24 kbps, y 32kbps.

G.726 ofrece calidad casi idéntica a la de G.711, pero utiliza sólo la mitad del ancho de banda. Esto se debe a que en lugar de enviar el resultado de la medida de una cuantización, envía únicamente suficiente información para describir la diferencia entre la muestra actual y la anterior a ella. Es popular, de hecho, porque no requiere de mucho trabajo computacional del sistema.

1.5.8.3 G.723.1

No debe confundirse con G.723 (que es otra versión obsoleta de ADPCM), este códec está diseñado para una comunicación de baja tasa de transferencia. Tiene dos configuraciones de velocidad de transferencia de información: 5.3 kbps y 6.3 kbps, siendo uno de los códecs requeridos de conformidad con el protocolo H.323 (aunque otros códecs pueden ser empleados con H.323).

Actualmente su uso es bloqueado por patentes y por tanto requiere licenciamiento si va a ser usado en aplicaciones comerciales.

1.5.8.4 G.729a

Considerando el poco ancho de banda que utiliza, éste códec entrega una calidad impresionante, a través del uso de un algoritmo llamado CS-ACELP (*Conjugate-*

Structure Algebraic-Code-Excited Linear Prediction), que es un popular método de compresión de voz. Debido a sus patentes, no se puede utilizar G.729 sin pagar por una licencia; sin embargo, es extremadamente popular y, por tanto, es soportado por muchos teléfonos y sistemas. G729a utiliza un ancho de banda de 8 kbps.

1.5.8.5 GSM

GSM es el código favorito de Asterisk, pues su uso no es bloqueado con ningún requisito de licenciamiento como G.723.1 y G729a y ofrece rendimiento excepcional respecto de la demanda que tiene sobre el CPU. La calidad del sonido es considerada de menor grado que la que ofrece G.729a y opera a 13 kbps.

1.5.8.6 iLBC (*Internet Low Bitrate Codec*)

El Códec de baja velocidad de transferencia de Internet provee una mezcla atractiva de bajo uso de ancho de banda y calidad, siendo especialmente muy bien hecho para mantener una calidad razonable sobre enlaces de red de pérdida.

No es tan popular como los códecs ITU y por tanto podría no ser compatible con los teléfonos y sistemas de VoIP comerciales.

Debido a que iLBC utiliza algoritmos complejos para lograr sus altos niveles de compresión, debe exigir un gran esfuerzo de procesamiento en los sistemas.

iLBC opera a una tasa de transferencia de 13.3 kbps (con tramas de 30 milisegundos) y a 15.2 kbps (con tramas de 20 milisegundos).

1.5.8.7 Speex

Es un códec de tasa de transferencia variable, lo que significa que es capaz de modificar dinámicamente su velocidad de transferencia de información en respuesta al cambio de las condiciones de red. Se ofrece en versiones para banda estrecha y banda ancha, dependiendo de la calidad que deseemos.

Speex es un código totalmente abierto y puede operar en el rango de 2.15 a 22.4 kbps.

1.5.9 CALIDAD DE SERVICIO (*QoS* –QUALITY OF SERVICE)

La calidad de servicio se refiere al reto de entregar un flujo de datos sensible al tiempo, a través de una red que fue diseñada para entregar datos con una estrategia del “mayor esfuerzo” y de forma temporal. Aunque no hay una regla estricta, es generalmente aceptado que si es posible entregar el sonido proveniente de la persona que habla al oído de quien la escucha en 300 milisegundos, es posible un flujo normal de conversación. Cuando el retardo supera los 500 milisegundos, se vuelve difícil que los interlocutores no se interrumpan entre sí. Luego de un segundo, una conversación normal se vuelve extremadamente complicada.

Es comprensible el hecho de que aplicaciones como telefonía IP y videoconferencia requieren de un mayor control de ancho de banda que las aplicaciones tradicionales de Internet como HTTP y FTP, o telnet, que no pueden tolerar pérdida de paquetes pero son menos sensitivas a retardos variables en el flujo de transmisión. La mayoría de aplicaciones en tiempo real muestran un comportamiento exactamente opuesto ya que pueden soportar una razonable cantidad de pérdida de paquetes, pero son usualmente muy críticas ante los retardos altamente variables.

Esto significa que sin un control del ancho de banda, la calidad de estos flujos en tiempo real depende del ancho de banda que en ese instante esté disponible.

En general, la calidad de servicio se refiere a los mecanismos de control que pueden proveer diferentes niveles de priorización a usuarios distintos o flujos de información, o garantizar un determinado nivel de rendimiento a un flujo de datos de acuerdo con las peticiones del programa de aplicación.

Ciertos conceptos son necesarios para garantizar una calidad de servicio específica para aplicaciones de tiempo real en el Internet. La QoS puede también describirse como un conjunto de parámetros que describen la calidad (por ejemplo, ancho de banda, uso de búfer, y uso de CPU) de un flujo de datos específico. La pila de protocolos IP básica provee solo un nivel de calidad de servicio, llamada *mayor esfuerzo*. Los paquetes son transmitidos de un punto a

otro sin ninguna garantía de un ancho de banda especial o un tiempo mínimo de retardo. Con este modelo de tráfico del *mayor esfuerzo*, las peticiones en Internet son manejadas con la estrategia FIFO (*first input first output)* o “el primero que ingresa es el primero en ser procesado”. Lo que significa que todas las peticiones tienen la misma prioridad y se manejan una luego de otra. No hay posibilidad de hacer reservación de ancho de banda para conexiones específicas o de brindar prioridad para solicitudes especiales. Por lo tanto, nuevas estrategias fueron desarrolladas para proveer servicios predecibles al Internet.

Actualmente hay dos mecanismos a través de los que se puede brindar QoS a las redes basadas en IP y por tanto a Internet: Servicios Integrados y Servicios Diferenciados.

1.5.9.1 Servicios Integrados

El modelo de servicios integrados (IS) fue definido por un grupo de trabajo de la IETF e incluye el actual servicio del *mayor esfuerzo* y el nuevo servicio de tiempo real que provee funciones de reserva de ancho de banda sobre las redes IP. Fue desarrollado para optimizar la red y la utilización de recursos por nuevas aplicaciones, como multimedia en tiempo real, la que requiere garantías de QoS.

Los Servicios Integrados permiten mejoras al modelo de red IP para soportar transmisiones en tiempo real y ancho de banda garantizada para flujos específicos.

En este caso se define un flujo como una corriente de datagramas relacionados desde un único remitente hacia un único receptor, dando como resultado una única actividad de usuario y que requiere la misma QoS.

Este modelo se basa en la reserva de recursos para cada conexión individual, es decir que si al establecerse una transmisión de vídeo entre dos terminales, la herramienta de videoconferencia necesita un ancho de banda mínimo de 128 kbps y un retardo mínimo de 100 milisegundos, esa calidad de servicio deberá reservarse, de modo que una transmisión continua de video se pueda asegurar.

Esto quiere decir que para soportar este modelo, un ruteador de Internet debe ser capaz de proveer una QoS apropiada para cada flujo, a través del control del tráfico.

1.5.9.2 Servicios Diferenciados

Los mecanismos de servicios diferenciados no utilizan señalización por flujo y como resultado no consumen un estado por cada flujo en la infraestructura de enrutamiento. Es posible alojar diferentes niveles de servicio a diferentes grupos de usuarios o grupos, lo que significa que todo el tráfico es distribuido en grupos o clases con varios parámetros de QoS. Esto reduce la cantidad de mantenimiento requerido en comparación con los Servicios Integrados.

También llamados *DiffServ*, es un método por el que el tráfico puede ser marcado y mediante esa marca, puede recibir un tratamiento específico.

DiffServ es una arquitectura para redes de computadoras que especifica un mecanismo simple y escalable para clasificar y manejar tráfico de red y para proveer garantías de QoS en las redes IP modernas.

Como se puede apreciar, se ajusta a las necesidades de la transmisión de información en tiempo real como la VoIP que analizamos. Sin embargo, en contraposición con el modelo de Servicios integrados, DiffServ es un mecanismo de “grano grueso”, lo que significa que opera colocando cada paquete en un número limitado de clases de tráfico en lugar de diferenciar el tráfico de red basado en los requerimientos de un flujo individual.

DiffServ se basa en un mecanismo para clasificar y marcar paquetes de acuerdo a la clase a la que pertenezcan, indicando a los ruteadores la implementación de *Comportamientos por Salto (PHBs –Per-Hop Behaviours)*, que definen las propiedades del envío de paquetes asociadas con una clase de tráfico. Diferentes PHBs pueden ser definidos para ofrecer baja pérdida, baja latencia o propiedades de envío con el mayor esfuerzo.

Se puede formar un Dominio de servicios diferenciados agrupando un conjunto de ruteadores que implementan políticas DiffServ comunes definidas administrativamente.

El tráfico de red que ingresa en un dominio DiffServ es sujeto a clasificación y condicionamiento. El tráfico puede ser clasificado en base a diferentes parámetros como la dirección origen, la dirección destino o el tipo de tráfico.

Los clasificadores de tráfico pueden tomar en cuenta cualquier marca DiffServ en los paquetes recibidos o pueden elegir ignorarlos o incluso anularlos.

1.5.9.3 Eco

El eco es un fenómeno causado debido al hecho de que un circuito de lazo local debe transmitir y recibir en el mismo par de cables. Si este circuito no está eléctricamente balanceado, o si un teléfono de baja calidad es conectado al final del circuito, la señales que se reciben pueden reflejarse, formando parte de una transmisión de retorno. Cuando este circuito reflejado regresa, escucharemos las palabras que dijimos pocos momentos antes. El oído humano percibe un eco después de un retardo de aproximadamente 40 milisegundos.

1.6 PBX (*PRIVATE BRANCH EXCHANGE*)

1.6.1. CONCEPTO

Una PBX, también conocida como una central telefónica privada, es un servicio de conmutación telefónica para una entidad privada o una empresa y su objetivo es el de proveer comunicaciones de voz (e incluso datos) a los usuarios dentro de la organización. Estos usuarios suelen compartir un conjunto de líneas externas para realizar y recibir llamadas desde la PSTN.

1.6.2. HISTORIA

Inicialmente, una central telefónica era una oficina en la que convergían todos los cables que conducían a las comunicaciones de voz. En estas centrales existían operadores que recibían las llamadas y las conectaban al teléfono con el cual los abonados deseaban comunicarse. A este sistema se le denominaba “Tablero de Conexiones” o “Tablero de conmutación” (*Exchange Board*), a través del cual se podía interconectar cualquier abonado con otro. Como se puede entender, la conmutación era manual, y a este sistema se le llamó PMBX (*Private Manual Branch Exchange*).

Las centrales telefónicas alrededor del mundo permitían la conexión entre sus abonados pero también con abonados de centrales de otras ciudades o países a

través de centrales de larga distancia cuyas operadoras tenían acceso a sistemas telefónicos fuera del dominio local. El hecho de que la conmutación fuera manual provocaba que el servicio fuese caro y sumamente engorroso.
Posteriormente, con la introducción de nueva tecnología, este sistema manual fue sustituido por mecanismos electromecánicos y electrónicos de conmutación, que tomaron el nombre de PABX (*Private Automatic Branch Exchange*) o simplemente, como ahora se conoce, PBX, en los que la conmutación de los circuitos se hacía de forma automática.

El carácter *privado* de la central se adquirió el momento en el que algunas compañías pensaron en la posibilidad de manejar el tráfico de voz interno de manera autónoma, teniendo el control de la conmutación de los circuitos de manera local, ahorrando, por supuesto, algo de dinero al evitar el uso de la PSTN para ese efecto.
Este sistema telefónico privado permitía a los diferentes departamentos de las empresas comunicarse entre sí, y con el sistema telefónico público. Cada cable correspondiente a los departamentos (extensiones) convergía en un tablero de conexiones (*PBX board*). Esta central era igualmente capaz de comunicarse con la red telefónica externa (PSTN) mediante enlaces troncales que pertenecían a la compañía telefónica local. Gracias al tendido de los cables submarinos, se hizo posible también la comunicación entre continentes.
La tendencia más reciente en el desarrollo de PBX es la PBX VoIP más conocida como IPBX (*Internet PBX*), que utiliza el protocolo IP para transportar sus comunicaciones.
Los usuarios de una PBX pueden fácilmente comunicarse entre sí dentro de su organización, marcando simplemente el número asignado de la extensión. A menudo, para comunicarse con una persona que no pertenezca a la red interna, la central privada debe enrutar la llamada hacia la Red Telefónica Pública Conmutada, lo que puede involucrar el marcado de un código de acceso (0 o 9) junto al número telefónico de la persona a la que deseamos llamar.
Las labores de enrutamiento y conmutación de circuitos son elementos básicos de una PBX; sin embargo, en la actualidad es innumerable la cantidad de servicios

adicionales que puede prestar para las organizaciones y empresas, trayendo beneficios al mundo de la actividad comercial, industrial y financiera.

1.6.3 IPBX

Como se mencionó, a raíz de la revolución generada por las posibilidades de la transmisión de VoIP se empezó a fabricar hardware y software que trabajara con el objetivo de cursar el tráfico telefónico transportado en paquetes, utilizando el protocolo IP para ello. La mayoría de soluciones tienen características parecidas a sus similares PBX e incluso algunas mejoradas como la notificación de correo de voz a través de correo electrónico.

Se desarrollaron algunas soluciones de sistemas en base a software que podrían funcionar utilizando el procesamiento de un ordenador que disponga de las interfaces adecuadas para su interconexión con las extensiones respectivas y con otras redes que pudiesen transportar la información de voz.

1.6.4 FUNCIONES DE UNA PBX

Las funciones principales de una PBX son:

- Establecer conexiones (circuitos) entre los aparatos telefónicos de dos usuarios (o más), mediante el mapeo de un número telefónico con un teléfono físico, asegurando que éste no está ocupado.
- Mantener las conexiones durante el tiempo que los usuarios lo requieran.
- Proveer información de estadísticas de llamada.

A parte de las funciones anteriores, una central PBX puede prestar servicios bastante novedosos, algunos de ellos incluso desconocidos para los usuarios comunes, pero que también están brindando un valor agregado de mucha importancia al servicio de comunicación de voz. Entre ellos podemos contar:

Contestadora Automática.- Es un sistema que permite que quienes llaman sean contestados por una recepcionista “electrónica”, que puede guiarle para alcanzar

la extensión deseada o simplemente darle información o indicaciones respecto de la organización.

Distribuidor automático de llamada.- Es un mecanismo que distribuye las llamadas entrantes hacia in grupo específico de terminales.

Servicios de Directorio Automatizado.- En el que quienes llaman pueden ser enrutados hacia la extensión de un miembro dado de la organización al presionar o decir las letras del nombre de esa persona.

Timbrado de regreso automático.- Un servicio que nos ayuda en el caso de que el número que marcamos está ocupado, marcando automáticamente por nosotros hasta que la línea esté libre y notificándonos de este hecho mediante un timbrado especial.

Estadística de llamada.- Permite llevar un registro de las llamadas y su duración, proveyendo de reportes de la actividad telefónica.

Redirección de llamada.- En el caso de no estar junto a nuestro terminal de uso común, la llamada puede ser reenviada a otro terminal como un teléfono celular.

Parqueo de llamadas.- Es un servicio que permite a una persona poner una llamada en espera en una extensión y poder continuar con la conexión en otra extensión.

Levantamiento de llamadas.- Es un servicio que permite que una llamada pueda ser contestada por un aparato telefónico distinto al que está timbrando, mediante el ingreso de un código o clave.

Transferencia de Llamada.- Es un mecanismo que permite que un usuario reubique una llamada hacia otro terminal.

Llamada en espera.- Un servicio a través del cual es posible recibir una llamada mientras una conexión ya se ha establecido (una llamada existente).

Retorno de la última Llamada.- Servicio que nos permite marcar directamente el último número del que se recibió una llamada.

Conferencia.- Es un servicio a través del cual se puede establecer una conversación telefónica entre más de dos participantes.

Música en Espera.- Es un servicio mediante el que se puede reproducir alguna melodía agradable mientras la persona que llama espera por la comunicación.

Servicio Nocturno.- Es aquel que permite el manejo de las llamadas que se reciben especialmente fuera del horario en el que pueden ser atendidas (horario laboral) y que permite que se redirijan al correo de voz o hacia algún otro servicio específico.

Correo de Voz.- Es un mecanismo que permite el manejo de mensajes telefónicos que se almacenan cuando la persona que contesta una extensión determinada no puede atender la llamada.

IVR (Respuesta de Voz Interactiva).- Es un sistema que permite a quienes llaman consultar en una base de datos u otra fuente de datos usando el terminal telefónico.

1.7 ASTERISK

No hace mucho tiempo todo el entorno tecnológico (especialmente software y telecomunicaciones) se encontraba monopolizado por un pequeño grupo de empresas que creaban las tecnologías y productos, y otras que los utilizaban para prestar servicios. Sin embargo, en la década de 1990, con la explosión producida por el aparecimiento del Internet, los costos de la tecnología cayeron.

Surgieron entonces nuevas empresas, servicios, productos innovadores y hasta revolucionarios, como la corriente de la comunidad desarrolladora de software libre, que desembocó en la creación de una plataforma completa basada en su filosofía.

Los sistemas de comunicaciones telefónicas se mantuvieron propietarios, a pesar de todo este proceso, quizá debido a que su desarrollo parecía no augurar demasiado potencial como otras aplicaciones que brindaban más que todo entretenimiento.

En el año de 1999, luego de fundar la empresa *Linux Support Services* para dar soporte técnico comercial para Linux, Marc Spencer, movido por la necesidad de un sistema telefónico que brindara asistencia técnica durante las 24 horas del día, decidió desarrollar un sistema que le permitiría hacer *todo* (relativo a la señal telefónica) una vez que pudiese recibir una llamada mediante una PC, después de todo –según el decía, “sería sólo software”.

De ese modo, y luego de colocar su sistema en la web, dos años más tarde, la economía había crecido de tal manera, que era más adecuado concentrarse en el sistema de telefonía que había diseñado (y al que había llamado Asterisk), que dar soporte técnico de propósito general para Linux. En esa misma etapa Spencer contactó con Jim Dixon quien se encontraba trabajando en el Proyecto de Telefonía *Zapata* que se complementaba de forma tal que juntos desarrollaron la primera tarjeta PCI de interfaz de telefonía

El mercado de las telecomunicaciones es inmenso, debido a la penetración de los teléfonos en la vida cotidiana y laboral. Además, los consumidores reclaman por un servicio de telecomunicaciones tremendamente *adaptable*, que pueda satisfacer plenamente sus necesidades y las soluciones de telecomunicaciones propietarias son demasiado costosas, lo que hace de Asterisk un sistema revolucionario.

Asterisk es la cumbre de una variedad de transiciones (Propietario – Código Abierto, Conmutación de Circuitos – VoIP, Sólo Voz – Voz, Vídeo, y Datos, etc). Por otro lado, Asterisk es capaz de utilizar desde la tecnología de la era de la telefonía de marcación por pulsos de 1960 hasta la correspondiente a los últimos dispositivos inalámbricos para VoIP.

Quizá lo más importante es que *Asterisk demuestra cómo una comunidad de gente y compañías motivadas pueden trabajar en conjunto para crear un proyecto con un alcance tan significativo que ninguna otra persona o compañía individualmente habría sido capaz de crear.*

De forma general, Asterisk[4] es una plataforma de telefonía convergente de código abierto, que está diseñada para funcionar sobre Linux. Asterisk combina más de 100 años de conocimiento de telefonía en un conjunto robusto de aplicaciones integradas de telecomunicaciones. El poder de Asterisk radica en su naturaleza adaptable, complementada por el soporte de una gran cantidad de estándares.

Aplicaciones como correo de voz, conferencia, encolamiento de llamadas, música en espera, parqueo de llamadas son todas características estándar construidas a

[4] VAN MEGGELEN, Jim; SMITH, Jared; MADSEN, Leif, “Asterisk, The Future of Telephony”. O’Reilly Media Inc. USA, Septiembre 2005, Primera Edición

base de software. Además, Asterisk puede integrarse con otras tecnologías de una manera tal que las PBX propietarias podrían difícilmente soñar.

1.7.1 LA REVOLUCIÓN EN TELEFONÍA

Asterisk ha desencadenado una revolución tecnológica en el campo de las telecomunicaciones, la industria electrónica más grande que ha permanecido intocable por la revolución del “código abierto”.

Aún quedan empresas gigantescas empresas manufactureras de hardware que construyen sistemas extremadamente caros, incompatibles, complicados, caducos sobre hardware hace ya tiempo obsoleto.

El proyecto Asterisk cambia todo esto, pues defiende el concepto del manejo de estándares, lo que brinda libertad para desarrollar innovaciones propias, pues no se imponen límites.

1.7.2 PROYECTO DE TELEFONÍA ZAPATA

Este proyecto fue concebido por Jim Dixon, un ingeniero consultor de telecomunicaciones que fue inspirado por los increíbles avances en la velocidad de los CPU que la industria había garantizado. El pensamiento de Dixon se concentraba en la creación de sistemas telefónicos mucho más económicos que no estuvieran formados más que por los componentes electrónicos básicos requeridos para interconectarse con un circuito telefónico. Con el objetivo de no tener caros componentes en la tarjeta, el procesamiento digital de señales sería manejado en el CPU mediante software. Debido a que esto traería una tremenda carga para el CPU, este visionario estaba seguro de que el bajo costo de los CPUs respecto de su rendimiento, los hacía mucho más atractivos que los costosos DSPs (Procesador Digital de Señales), y mucho más importante, que la relación precio/rendimiento continuaría mejorando a favor del consumidor, a medida que los CPUs continuaran el incremento de su capacidad de procesamiento.

1.7.3 ASTERISK: EL CATALIZADOR DE LA REVOLUCIÓN

No cabe duda de que Asterisk es el catalizador de un proceso que representa una revolución en el ámbito de las telecomunicaciones, especialmente en telefonía.

El sistema telefónico convencional tiene limitaciones de diseño que no han sido corregidas en muchos años y esta falta de flexibilidad no es propia de organizaciones que se supone prestan servicios de calidad a sus suscriptores.

No importa cuan equipado esté un sistema, la realidad es que cualquier PBX en existencia sufre deficiencias, siempre algo saldrá mal y aún cuando se trate de equipos sofisticados, estos en algún momento serán incapaces de adaptarse a la creatividad del consumidor. Habrá usuarios que deseen alguna funcionalidad especial que el equipo de diseño no anticipó, pues su implementación no era justificada y, dado que el mecanismo es cerrado y privativo, los usuarios no podrán adaptarlo por su cuenta.

Aún en esta temprana etapa de éxito, Asterisk está nutrido de un mayor número de colaboradores que cualquier otra PBX, pues la mayoría de empresas no dedican más que unos pocos desarrolladores a sus proyectos, Asterisk posee miles.

1.7.4 REQUERIMIENTOS DE UN SISTEMA DE TELEFONÍA IP EN BASE A ASTERISK

1.7.4.1 Conexión de Banda Ancha

De acuerdo al códec que se utilice para la transferencia de voz a través de nuestra red interna y el Internet, se podrá determinar la capacidad de las conexiones que se requieren para una adecuada conversación vocal. Como se estudió con anterioridad, algunos de los mecanismos de codificación (códecs) pueden permitir la transmisión de la voz ocupando un ancho de banda de cerca de los 100 kbps cuando se incluye en el flujo las cabeceras correspondientes de los paquetes. Esto nos indica que, sin lugar a dudas, será necesaria una conexión de banda ancha si deseamos que las conversaciones se lleven a cabo de una manera medianamente normal, en el caso de llamadas a través de Internet, y desde luego dentro de la red interna. Además, normalmente el tráfico que cursa nuestra red interna a la organización o hacia y desde el Internet, no será

solamente tráfico telefónico, sino que incluirá descarga de información, acceso a servidores FTP, HTTP, y otros. Estos significa que quizá, al menos se necesitará una conexión de 128/128 kbps hacia Internet de modo que se satisfagan las necesidades más básicas de voz y de transferencia de datos en general.

1.7.4.2 Plataforma

El sistema de Asterisk fue originalmente desarrollado para Linux aunque actualmente también funciona en BSD, MacOSX, Solaris y Microsoft Windows. Desde luego, la plataforma mejor soportada es Linux, pues, por ejemplo, para Windows no existe software para el manejo de interfaces con la PSTN.

1.7.4.3 Número de Extensiones

El número de extensiones no es esencialmente un limitante de la estructura de VoIP sobre la que pueden desarrollarse las comunicaciones de voz, sino más bien el número de llamadas que pueden establecerse de manera simultánea. La razón es que por cada llamada se abrirá un canal de voz que consumirá un ancho de banda determinado por el códec y la compresión que se utilice, de modo que mientras más llamadas concurrentes, mayor será el ancho de banda que se deba asignar a toda la conexión. Es importante proyectar estas necesidades de acuerdo a la cantidad de usuarios dentro de la organización y de acuerdo también a las necesidades internas de comunicación, a la distribución de los departamentos y a la intensidad del tráfico de voz que requiere la organización.

1.7.4.4 Servicios

Dentro de los requerimientos que pueden analizarse son los servicios que la central IP va a prestar. Esto por supuesto depende mucho de las necesidades de la organización y aquí las opciones son sumamente variadas. Como se mencionó dentro de las funciones de una PBX, la plataforma de Asterisk nos permite agregar mucha funcionalidad a la PBX de software.

1.7.4.5 Procesamiento

Las necesidades de un sistema como Asterisk son similares a las correspondientes a las de cualquier aplicación embebida en tiempo real. Esto se debe fundamentalmente a su necesidad de tener acceso prioritario al procesador y a los buses de sistema, del mismo modo que sucede con la transferencia a través de la red. Es por tanto imperativo que cualquier función en el sistema que no esté directamente relacionada con las tareas de procesamiento de llamadas de Asterisk esté corriendo bajo una prioridad menor.

Al igual que sucede con los requerimientos de ancho de banda, los requerimientos de rendimiento relacionados con el procesamiento de las señales debe tomarse muy en cuenta en el diseño de un sistema de este tipo, pues cualquier inconveniente al respecto se reflejará en una pérdida de la calidad de audio, eco, retardo, etc.

La tabla siguiente lista algunas guías que se podrán seguir cuando se planee nuestro sistema.

TABLA 1.2 POSIBLES REQUERIMIENTOS DE PROCESAMIENTO Y MEMORIA DEL SISTEMA ASTERISK

Propósito	**Número de canales**	**Recomendado Mínimo**
Pruebas	No más de 5	400-MHz x86, 256 MB RAM
SOHO	5 a 10	1-GHz x86, 512 MB RAM
Sistema pequeño	Hasta 15	3-GHz x86, 1GB RAM
Sistema mediano-grande	Más de 15	CPUs duales, possible arquitectura distribuida

Se puede agregar que la capacidad de almacenamiento de disco duro dependerá de la cantidad de correo de voz que se desea almacenar.

Si lo que se desea se orienta a grandes instalaciones, es común desplegar funcionalidad a través de muchos servidores. Una o más unidades centrales se dedicarán al procesamiento de las llamadas.

Como es común en la mayoría de ambientes en Linux, Asterisk está muy bien proporcionado para crecer con las necesidades de sus usuarios.

La flexibilidad es una razón clave por la que Asterisk resulta conveniente en cuanto a costos para organizaciones que crecen rápidamente; no hay un tamaño máximo o mínimo a considerar cuando se dimensiona la compra inicial. Mientras algo de escalabilidad es posible, con la mayoría de sistemas telefónicos, en realidad ninguno logra igualar la escalabilidad que logra Asterisk a tan bajo costo.

1.7.4.6 Hardware

Los requerimientos de la plataforma de hardware estarán acorde con la confiabilidad y rendimiento que deseemos darle al sistema. Generalmente, cualquier plataforma x86 será suficiente para instalar el software; sin embargo el hardware que seleccionemos generalmente estará de acuerdo a la funcionalidad que se quiera lograr.

En realidad no hay una matriz que nos indique la forma cómo dimensionar el hardware de un sistema que albergará a Asterisk, la forma en la que vaya a ser usado el sistema es un factor fundamental que nos indicará los recursos que deberá poseer.

Entre los factores que permitirán determinar los recursos de hardware necesarios se encuentran:

El máximo número de conexiones concurrentes que soportará el sistema. Como se indicó anteriormente, mientras más conexiones existan, se incrementará asimismo la carga del sistema.

El porcentaje de tráfico que requerirá codificación intensiva. El procesamiento digital de señales hecho en base a software por parte de Asterisk, dependiendo de la codificación que se emplee, puede reducir considerablemente el número de llamadas concurrentes que puede soportar.

El nivel de actividad de conferencia que se requiere. La conferencia requiere que el sistema trans-codifique y combine los flujos de audio casi en tiempo real y ello requerirá que se eleve la carga de procesamiento.

Cancelación de Eco. Se requerirá siempre que interactuemos con la PSTN y este proceso incrementará la carga del CPU.

El efecto que tiene en el rendimiento del sistema cada uno de estos factores es bastante complicado de calcular y aún no existe un método exacto que permita el cálculo del *performance*. Esto es en parte debido a que el efecto de cada componente del sistema depende de muchas variables como: potencia del CPU, chipset de la tarjeta madre, carga total de tráfico, optimizaciones del kernel Linux, tráfico de red, número y tipo de interfaces PSTN, y tráfico PSTN, sin olvidar otros servicios no relacionados con Asterisk que se ejecutan de forma concurrente.

Entre algunos de los efectos de factores importantes están:

Códecs y Transcodificación. Dado que un códec define un conjunto de reglas matemáticas que determinan la forma en que una onda analógica será digitalizada, la diferencia entre varios de ellos radica en gran parte en los niveles de compresión y calidad que ofrecen. De forma general, mientras mayor compresión se requiera, mayor será el trabajo que deba realizar el procesador digital de señales para codificar o decodificar la señal.

Los códecs que no realizan compresión hacen menos intenso el trabajo del procesador, pero requieren mayor ancho de banda de red. Es necesario, por tanto, que cuando se seleccione el códec, se haga un balance entre ancho de banda y uso de procesador.

Unidad de Procesamiento Central (y Unidad de Punto Flotante). Un CPU está compuesto de muchos componentes, uno de los cuales es la Unidad de Punto Flotante (FPU). La velocidad del CPU, junto con la eficiencia de su FPU jugará un rol fundamental en cuanto al número de usuarios que un sistema pueda soportar efectivamente.

Optimizaciones del kernel. Un núcleo de sistema operativo optimizado para su rendimiento de una aplicación en específico es algo que muy pocas distribuciones de Linux ofrecen por defecto.

En Linux, es posible descargar una copia de la última versión del kernel y compilarla en nuestra plataforma. Es posible también adquirir parches que puedan contener mejoras en el rendimiento.

Latencia IRQ (Interrupt ReQuest). La latencia de petición de interrupción es básicamente el retardo entre el momento que una tarjeta periférica (como una tarjeta de interfaz telefónica) pide que el CPU detenga lo que esté haciendo para ejecutar otra tarea. Los periféricos de Asterisk (especialmente las tarjetas Zaptel) son extremadamente intolerantes a la latencia IRQ.

Versión de kernel. Asterisk es oficialmente soportado por la versión 2.6 del kernel de Linux.

Distribución Linux. Las distribuciones de Linux son sumamente variadas. Cualquiera de ellas puede utilizarse para la instalación de Asterisk.

1.7.4.7 Elección del Procesador

No cabe duda de que mientras más capacidad tenga nuestro procesador, más potente será nuestro sistema. Sin embargo no tiene mucho sentido tener la capacidad de procesar gran cantidad de información cuando no podemos transmitirla con la misma eficiencia a través de la red.

Para *sistemas pequeños* de hasta 10 teléfonos, cualquier procesador moderno podría satisfacer las necesidades de Asterisk, aunque se debe tomar en cuenta la escalabilidad que se podría requerir. Para ejecución de pruebas se han utilizado procesadores de 433 a 700 MHz, pero para ambientes de producción debería considerarse un mínimo de 2 GHz.

Los *sistemas medianos* de 15 a 50 teléfonos, se podrán desplegar utilizando uno o dos servidores pero deberá tomarse muy en cuenta el aumento de carga, de manera que se puedan identificar los problemas de rendimiento a tiempo.

En *sistemas grandes* (sobre los 50 usuarios), la carga puede distribuirse a través del uso de múltiples núcleos, de modo que los detalles relacionados con el rendimiento pueden manejarse a través del incremento de máquinas. Sistemas Asterisk muy grandes (de 500 a 1000 usuarios) se han podido crear de esta forma.

1.7.4.8 Elección de la tarjeta madre

Los diferentes buses del sistema deben proveer una latencia mínima. Por ejemplo, si estamos planeando una conexión que utilice interfaces análogas PRI para una conexión PSTN, las tarjetas Zaptel generarán 1000 peticiones de interrupción por segundo. Si existen dispositivos en el bus que interfieren con este proceso, la calidad de la llamada se degradará. Sin duda, los procesadores Intel y AMD aparecen como las marcas más renombradas en esta área. Es necesario determinar si el *chipset* de la tarjeta madre que tenemos o deseamos adquirir tiene algún inconveniente con la latencia IRQ.

Si se utiliza tarjetas Zaptel en nuestro sistema, debemos asegurarnos que el BIOS nos permita máximo control sobre la asignación IRQ. Como regla, las tarjetas madre de calidad ofrecerán mucha más flexibilidad respecto del afinamiento del BIOS.

Se debería considerar el uso de procesadores múltiples, de modo que se provea una habilidad mejorada para manejar múltiples tareas. Para Asterisk, esto será especialmente beneficioso en el área de operaciones de punto flotante.

Es importante adquirir una tarjeta madre que no contenga componentes de audio y/o video integrados. Si deseamos componentes adicionales, será más ventajoso instalarlos nosotros mismos.

La misma recomendación para las tarjetas de red. Siempre será menos posible comprometer el sistema completo, aislando los puntos de falla, de modo que puedan ser inmediatamente reemplazados.

La estabilidad y calidad de un sistema Asterisk dependerá sin duda de los componentes que se seleccionen para su arquitectura. El conocimiento de todos estos componentes permitirá un manejo mucho más conciente de todos los parámetros involucrados.

1.7.4.9 Suministro de Energía

El suministro de energía que se seleccione para el sistema es un factor determinante en la estabilidad de la plataforma entera. Esto no quiere decir que Asterisk resulta ser una aplicación que requiera demasiado consumo de energía;

sin embargo, cualquier cosa relacionada con multimedia o audio profesional es generalmente sensitiva a la calidad de la potencia.

El suministro de energía redundante es en ocasiones necesario en ambientes en los que se requiere de alta disponibilidad. Esto involucra la existencia de dos fuentes de energía, totalmente independientes, capaces de proveer cada una de los requerimientos de energía del sistema.

1.7.5 HARDWARE TELEFÓNICO

Si se va a conectar un sistema Asterisk a cualquier equipo de telecomunicaciones antiguo como la PSTN, se necesitará el hardware correcto. Como en los puntos anteriores, el hardware requerido estará determinado por lo que queremos lograr.

1.7.5.1 Conexión a la PSTN

Asterisk permite la interconexión de redes de telecomunicaciones de conmutación de circuitos con redes de conmutación de paquetes. Debido a la arquitectura abierta de Asterisk (y su código abierto) resulta sencillo encontrar en estos días hardware de interfaz compatible. Cabe destacar que la selección de tarjetas de interfaz de telefonía de código abierto es actualmente limitada, pero a medida que Asterisk crece, eso cambiará rápidamente. Hasta el momento, y como ya se mencionó, una de las formas más populares y menos costosas de conectarse a la PSTN es usar las tarjetas de interfaz involucradas en el trabajo del Proyecto de Telefonía Zapata.

1.7.5.2 Tarjetas de Interfaz Analógicas

A menos que se necesite una gran cantidad de canales, existe la posibilidad de que nuestra interfaz con la PSTN consista de uno o más circuitos analógicos, cada uno de los cuales requerirá un puerto FXO (*Foreign Exchange Office*).

Digium, la compañía que inició el desarrollo de Asterisk, produce la tarjeta de interfaz analógica más popular, conocida como la TDM400P. Esta es una tarjeta base de cuatro puertos que permiten la inserción de hasta cuatro tarjetas hermanas que entregan ya sea puertos FXO o FXS (*Foreign Exchange Station*), La TDM400P puede adquirirse con estos módulos preinstalados, y Digium ha

diseñado números de parte para describir estas configuraciones. La convención de denominación es *TDMxyB*, donde *x* e *y* son números que representan la cantidad de módulos FXO y FXS que posee una tarjeta.

Una tarjeta más antigua producida por Digium fue la X100P. Ya no está disponible por parte de Digium, pero se pueden encontrar todavía clones de ella.

Otra compañía que produce tarjetas analógicas compatibles con Asterisk es Voicetronix, que posee tres tarjetas en su línea: OpenLine4, OpenSwitch6, y OpenSwitch12.

1.7.5.3 Tarjetas de Interfaz Digitales

Si se requieren más de 10 circuitos, o conectividad digital, estamos en el mercado de una tarjeta T1 o E1. Hay que tener en mente que los cargos mensuales de un circuito digital PSTN varían ampliamente. En algunos lugares, tan sólo 5 circuitos podrían justificar un circuito digital; en otros, la tecnología simplemente no será un aspecto que valga la pena el gasto.

Mientras mayor competencia existe entre los proveedores, mayor será la posibilidad de lograr un trato justo.

Digium produce diferentes tarjetas de interfaz de circuitos digitales. Las características en las tarjetas son las mismas; las diferencias primarias se centran en si proveen interfaces T1 o E1, y cuántas de ellas. Aunque es posible, es un consenso dentro de la comunidad de Asterisk que no se instale más de una de estas tarjetas en un sistema único.

Sangoma, es una empresa que ha estado produciendo tarjetas WAN *open source* por varios años y recientemente añadió soporte a Asterisk para sus tarjetas T1/E1.

1.7.5.4 Conexión exclusiva a la Red Telefónica de Conmutación de Paquetes

Si no se requiere una conexión a la PSTN, Asterisk no necesita más hardware que un servidor con una tarjeta de red. Sin embargo, si se brindará el servicio de música en espera o conferencia, y no se dispone de una fuente de temporización física, se necesitará el módulo del kernel Linux *ztdummy*, que es un mecanismo de reloj diseñado para proveer una fuente de temporización en un sistema donde

no existe un generador de temporización de hardware. En las últimas versiones del núcleo de Linux este generador ya no es necesario.

1.7.6 TIPOS DE TELÉFONOS

1.7.6.1 Teléfonos Físicos

Son cualquier dispositivo físico cuyo propósito fundamental es terminar un circuito de comunicaciones de audio entre dos puntos. En su forma más esencial estos dispositivos poseen un handset y un tablero de marcado.

Teléfonos Analógicos.- En un teléfono analógico, la señal transmitida es análoga al sonido producido por la persona que habla.

Teléfonos IP.-

Es un dispositivo muy similar físicamente a un teléfono análogo pero que permite realizar comunicaciones de voz a través de una red IP privada e incluso a través del Internet pues posee una interfaz RJ45 que le permite conectarse a una red asignándole una dirección IP como cualquier PC.

1.7.6.2 SoftPhones (Teléfono de software).-

Un softphone es un programa de software que provee la funcionalidad de un teléfono común en un dispositivo que no es un teléfono como un PC o una PDA. Un softphone debería probablemente tener una especie de pad de marcado, y debería proveer una interfaz que recuerde a los usuarios el diseño del teléfono tradicional. En el sentido más práctico y cotidiano (por ahora) podemos decir que un *softphone* es cualquier dispositivo que funciona como una aplicación en el sistema operativo de una computadora y que presenta la apariencia de un teléfono, proveyendo como función primaria la habilidad de hacer y recibir comunicaciones de audio full-dúplex (formalmente conocidas como “llamadas telefónicas”).

1.7.6.3 Adaptadores Telefónicos

Un adaptador telefónico (usualmente conocidos como ATA o Adaptador de Terminal Analógico) es un dispositivo de usuario final que convierte los circuitos

de comunicaciones de un protocolo a otro. De forma más común, estos dispositivos son utilizados para convertir alguna señal digital (IP o propietaria) a una conexión analógica a la que se pueda conectar un teléfono estándar o una máquina de fax tradicional.

Estos adaptadores podrían describirse como *gateways* dado que esa es su función. Sin embargo, el uso más popular del término *gateway telefónico* permitiría describir más bien a un adaptador telefónico multipuerto, con funciones de enrutamiento más complicadas.

1.7.6.4 Terminales de Comunicaciones

El término se refiere a cualquier dispositivo que permita al usuario final la comunicación (de cualquier tipo) con otro usuario. Esto relativiza enormemente el significado de teléfono pues, cualquier dispositivo, PC, PDA, *laptop*, que permita la realización de una llamada telefónica, podría considerase como un teléfono.

SEGURIDAD FIREWALL

1.8 INTRODUCCIÓN A LA SEGURIDAD EN BASE A FIREWALL

El aumento del número de usuarios de redes privadas, por el incremento de la demanda del acceso a servicios de Internet, como WWW *(World Wide Web),* Telnet, Correo Electrónico y FTP (*File Transfer Protocol*) ha marcado la necesidad de proteger estas redes locales del acceso no autorizado de entes ajenos externos a los que se exponen, por el hecho de conectarse a la red de redes. Es necesario proveer una protección adecuada a cada organización, de manera que los recursos propios de la red privada se mantengan inaccesibles a usuarios no autorizados, evitando así la exportación de información privada.

Por otro lado, es seguro que la implementación de un sistema de seguridad basado en firewall, utilizando una plataforma de sistema operativo Linux, resultará en un coste mucho menor que el que significaría adquirir piezas de hardware propietarias cuyo valor en el mercado alcanza fácilmente las 4 cifras.

1.9 CONCEPTOS FUNDAMENTALES DE SEGURIDAD

1.9.1 POLÍTICAS DE SEGURIDAD

Antes de iniciar con la implementación de cualquier medida de seguridad en un sistema computarizado, es importante preguntarse: ¿qué protegemos?, ¿por qué?, ¿contra qué?, ¿para qué?

Es importante también determinar qué equipos necesitan qué nivel de protección, así como *definir el comportamiento aceptable o inaceptable* de los equipos.

Una *política de seguridad* es, por tanto, un plan o un documento que contiene especificaciones de la forma cómo un equipo debe o no debe ser utilizado, un plan comprensivo de defensa que sea comunicado a la gerencia y a los usuarios finales. Esta política debe incluir un proceso de educación y capacitación a cada usuario del sistema, de la red y de los equipos, de manera que se logre la comprensión de las funestas consecuencias de las violaciones.

Es necesario aclarar que el firewall no constituye la política de seguridad, que es más bien un componente de la misma. Además, debe haber comunicación fluida con el nivel gerencial, pues será éste quien finalmente tome la decisión en cuanto a los equipos que se van a adquirir de acuerdo al nivel de seguridad que se desee implementar. Es deber del administrador de la red, asegurarse de que los gerentes comprendan la importancia de establecer una política de seguridad sólida de acuerdo con los requerimientos de la empresa.

Hay varios aspectos que se deben considerar en una política de seguridad:

- *Seguridad Física*: Especialmente lo que tiene que ver con la protección física de las estaciones de trabajo y los equipos de red, como protección contra robo, incendio, inundación, etc.
- *Seguridad de Red*: Se refiere a cualquier acción que se tome sobre una red, sin que tenga que ver con la estación física.
- *Autenticación*: O establecer si un usuario es realmente quien dice ser.
- *Autorización*: Es el acto de determinar si un usuario tiene permiso para ejecutar ciertas acciones.

La mayoría de las políticas de seguridad se adoptan a través del tiempo, conforme a las necesidades y servicios que se incorporen. Por ejemplo, hace algunos años el correo electrónico no era tan comúnmente utilizado, ahora es indispensable. Es por esto que una política de seguridad debe ser actualizada para reflejar esta tendencia.

1.9.2 SEGURIDAD FÍSICA

La seguridad física es un aspecto primordial, que *involucra todo aquello que prevenga el acceso físico no autorizado* a los equipos y a la red. Parecería un procedimiento obvio, pero es necesario tomar consideraciones específicas pues la protección física de los equipos es importante, sobre todo por los costos que representan y la información que manejan o almacenan. Es necesario asegurar las puertas, determinar una política acerca de quién tiene acceso a qué localidad y es necesario considerar una política de registro. La seguridad física involucra también la protección contra los inconvenientes que puedan derivarse de las irregularidades del ambiente. Se debe, por consiguiente, proteger los equipos del calor, de un corte de energía eléctrica, de la humedad.

1.9.3 SEGURIDAD DE RED

La seguridad en red involucra cualquier proceso que permita asegurar que ningún usuario pueda acceder a nuestros sistemas a través de la red. Esto puede parecer simple, pero es en realidad más complicado que la seguridad física.

1.9.4 AUTENTICACIÓN Y AUTORIZACIÓN

La autenticación es el proceso a través del cual se establece que algo o alguien es auténtico, es decir, se comprueba si es quien o lo que dice ser.

Esta verificación es realizada generalmente a través de una combinación nombre de usuario/*password* que es conocida por el usuario.

Autenticar un objeto significa *confirmar su procedencia*, mientras que autenticar una persona usualmente consiste en verificar su identidad.

Adicionalmente a la autenticación usuario/*password* existen algunas otras técnicas de autenticación que pueden también ser usadas:

- La criptografía de clave pública es a veces utilizada para establecer identidad, y esencialmente para encriptar correo electrónico y RSA.
- Tarjetas inteligentes o *smartcards.* Son usualmente tarjetas con un pequeño microchip en ellas. Estas tarjetas se insertan en un *slot* especial de una computadora y se usan para verificar la identidad de los usuarios. Este método puede ser muy útil en ambientes en los que mucha gente puede usar el mismo terminal, por muy corto tiempo, como en un hospital, universidad o biblioteca.
- Otro método de autenticación es la biometría, que es un mecanismo a través del cual un único aspecto del cuerpo del usuario es escaneado y utilizado para establecer que es realmente él o ella. Este aspecto puede ser una huella digital o el patrón de la retina.

Una vez que el usuario es autenticado, necesita recibir autorización. La autorización es un proceso en el que se establece la autoridad o el derecho de un usuario de ejecutar una operación determinada. Esto significa que la autenticación debe ser un procedimiento previo a la autorización.

1.9.5 HACKERS, CRACKERS Y SCRIPT KIDDIES

Hay tres tipos de personas de las que un administrador de redes debe cuidarse: hackers, crackers y script kiddies.

Un ***hacker*** es alguien que está interesado en tecnología para satisfacer su curiosidad, que desea saber cómo trabajan las cosas, cómo se configuran, y si pueden mejorarse. El aspecto más importante es quizá que se supone que no debería causar daño, pero podría hacerlo accidentalmente.

Un ***cracker*** es alguien cuya intención es conseguir algo, lo que puede significar acceder a nuestros sistemas para utilizar sus recursos (espacio de disco, tiempo de CPU, ancho de banda de la red), o acceder a nuestros datos (números de tarjetas de crédito, contraseñas u otros datos confidenciales). Un cracker puede también desear conseguir publicidad o incluso venganza.

Un ***script kiddie*** es alguien que prácticamente no posee conocimientos de los sistemas y protocolos involucrados, sin embargo usa las herramientas escritas por hackers y/o crackers para vulnerar estos sistemas.

1.9.6 TIPOS DE ATAQUES

Existen varios tipos de ataques que pueden ser usados por los hackers y *crackers*. La mayoría de ataques son, de hecho, alguna combinación de varios tipos de ellos.

1.9.6.1 Escaneo

En un ataque de escaneo un hacker intenta iniciar una sesión con cada servicio conocido que podemos proveer en un determinado host. Un hacker hace esto para establecer qué servicios están siendo ofrecidos, y qué versión de software se está utilizando para proveer ese servicio. Esta información es luego utilizada para compararse con una lista de vulnerabilidades de varios paquetes de software. A través de este ataque se puede llegar a determinar el sistema operativo que está ejecutando la máquina según los puertos que tiene abiertos.

1.9.6.2 Sniffing

En un ataque de *sniffing*, un hacker intenta capturar la información que está en tránsito a través de la red. Esta información puede contener datos confidenciales como contraseñas de determinados usuarios. Ha habido casos en los que los hackers han alterado tablas de enrutamiento en equipos para redireccionar el tráfico a través de un *sniffer* que instalaron en algún lugar.

1.9.6.3 Break-in (Robo)

Un intento de robo es usualmente el resultado de otro tipo de ataque, y usualmente significa que un usuario está tratando de obtener privilegios de usuario o súper-usuario en un servidor.

1.9.6.4 Ataque de Fuerza Bruta

Un ataque de este tipo básicamente significa que se intenta todas las combinaciones posibles para determinar la contraseña de usuario.

1.9.6.5 Ataque *Man-in-the-middle* (hombre en el medio)

Este tipo de ataque es aquel en el que un hacker ubica una computadora que actúa como un legítimo servidor para un cliente. Esto permite que un hacker capture y posiblemente altere la información en tránsito.

1.9.6.6 Virus

Un virus es un programa malicioso que típicamente se adjunta por sí mismo a otro programa. Cuando el otro programa se ejecuta, el programa malicioso se ejecuta también, permitiendo que se adjunte a más programas todavía.

1.9.6.7 Gusanos

Es un programa capaz de replicarse por sí mismo, sin la necesidad de otro programa que lo disperse.

1.9.6.8 Caballo Troyano

Un caballo troyano es un programa que puede hacer algo útil pero a la vez posee un contenido que puede causar daño de algún tipo.

1.9.6.9 Ataque DoS (Denegación de Servicio)

Es un ataque que hace daño interrumpiendo el servicio a usuarios legítimos, usualmente sobrecargando el servidor. Resulta bastante complicado detener este tipo de ataques pues usualmente involucran conexiones de clientes regulares. Un ataque DoS puede ser usado también para ingresar en un sistema puesto que ciertos programas no se comportan adecuadamente bajo gran carga.

Hay una variante de este tipo de ataque denominada DoS distribuido (DDoS), en el que se ejecutan múltiples ataques DoS (decenas o centenas, o incluso miles) a la vez.

1.9.7 FORMAS DE PROTECCIÓN

Existen básicamente tres formas de protegerse de los peligros que involucra una conexión a Internet.

- La *primera* forma es obvia: no conectarse a Internet por ninguna vía. Es una conducta muy segura pero será más perjudicial que tomar el riesgo, pues los usuarios dentro de la compañía pueden considerar el acceso a Internet como algo útil o esencial para su trabajo. Si la compañía no ofrece acceso a Internet, los usuarios podrían llevar sus módems al trabajo y establecer la conexión a Internet por ellos mismos y la red estaría abierta a ataques y ni si quiera lo sabremos.
- La *segunda* forma de protección es forzar a todos los datos a pasar por unos pocos puntos de obstrucción, comúnmente llamados *firewalls*, que pueden limitar y registrar la actividad de Internet.
- La última opción es dar a cualquiera en la compañía acceso a Internet ilimitado, cuando la gente lo necesite y que requiere que protejamos cada host individualmente. Esto es válido para universidades y proveedores de servicio de Internet (ISPs), pues son ellos quienes necesitan acceso ilimitado.

1.9.8 CONCEPTO DE FIREWALL

El concepto de “firewall” es quizá uno de los más mal interpretados términos en el Internet actualmente. Cuando se habla acerca de un firewall, la mayoría de gente piensa en una caja negra que lo protege todo, o piensan en un *Proxy* como si fuera un firewall.

Un corta-fuegos o firewall es *una combinación de componentes que implementan y hacen cumplir la política de seguridad.* Debido a que las políticas de seguridad de cada compañía son diferentes, cada firewall será diferente también.

La política de seguridad define la forma en la que se estructurará el firewall y los componentes de que se conformará.

Un firewall es una pieza de software o hardware que filtra todo el tráfico de red entre una computadora, nuestra red de casa, o la red de la compañía y el Internet.

1.9.9 POSICIÓN DEL FIREWALL

Es usual que el firewall se sitúe entre la red de la compañía y el Internet, permitiendo proteger la red interna de intentos de acceso no autorizado desde la

red externa, que puedan aprovechar vulnerabilidades de los sistemas de la red interna.

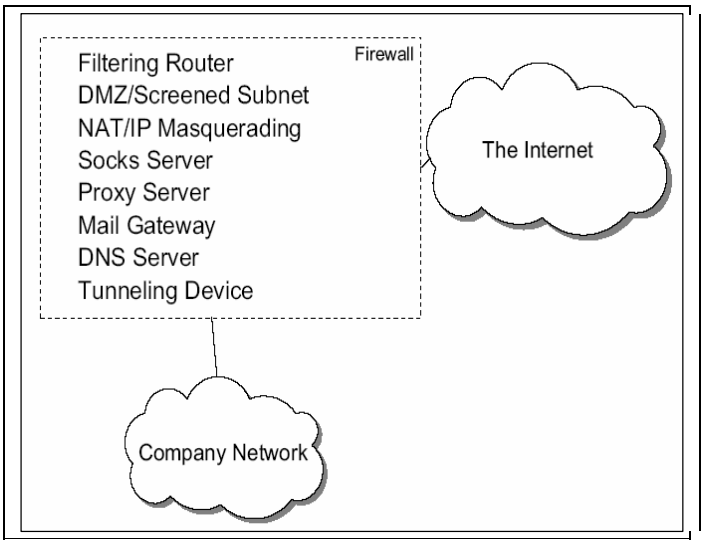


FIGURA 1.4 DIAGRAMA DE LA POSICIÓN TÍPICA DE UN FIREWALL

1.9.10 TIPOS DE FIREWALL

Un firewall puede clasificarse dentro de uno de los tipos siguientes (o como una combinación de ellos).

Firewall de Capa de Red.- Funciona a nivel de la capa de red de la pila de protocolos TCP/IP filtrando los paquetes IP a partir de sus diferentes campos: dirección IP origen, dirección IP destino. Usualmente este tipo de cortafuegos permiten el tráfico a partir de los campos de nivel de transporte (capa 4) tales como el puerto origen o el puerto destino, o incluso a nivel de la capa de enlace de datos (capa 2) como es el caso de la dirección MAC.

Firewall de Capa de Aplicación.- Como su nombre lo indica, este cortafuegos trabaja a nivel de aplicación, es decir en base al tráfico HTTP, permitiendo la intercepción de todos los paquetes que llegan o salen de una aplicación. El filtrado del tráfico se da, por ejemplo, en base a la URL a la que se intenta acceder. Es muy común que se denomine a este tipo de firewall como Proxy, y en general permite que los computadores de una organización entren a Internet de una manera controlada.

1.9.11 DMZ (*DEMILITARIZED ZONE*) Y FILTROS DE PAQUETES

La base de un firewall es usualmente la zona desmilitarizada o DMZ por sus siglas en inglés. La zona desmilitarizada es conectada al Internet y a la intranet a través de *routers* que son capaces de filtrar los paquetes IP que circulan basados en la dirección IP, el número de puerto, protocolo y sentido de la conexión establecida.

Los *rouetrs* que filtran los paquetes son configurados de manera que sea solamente posible establecer una conexión desde la intranet a un dispositivo en la DMZ y desde un dispositivo en la DMZ hacia el Internet. Una conexión directa desde la intranet hacia el Internet no es posible y las conexiones entrantes desde el Internet son también prohibidas.

Existen varias formas de conexión de una DMZ, como se puede observar en la gráfica.

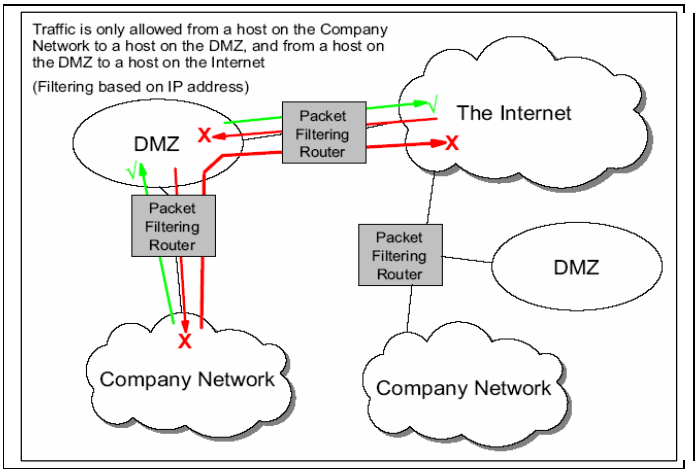


FIGURA 1.5 MÉTODOS DE CONEXIÓN DE DMZ

- Por un lado, en la parte izquierda, se utilizan dos ruteadores, de los que uno de ellos se sitúa entre el Internet y la DMZ, mientras que el otro entre la DMZ y la red interna. Cada ruteador posee, como se puede observar, dos conexiones de red.
- En la parte derecha, se utiliza un solo ruteador para conectar el Internet, la DMZ y la red interna juntos. Este único ruteador posee por tanto tres interfaces de red. La configuración es menos costosa pero ligeramente

más insegura, pues en lugar de vulnerar dos ruteadores, un hacker tiene que encargarse solamente de uno.

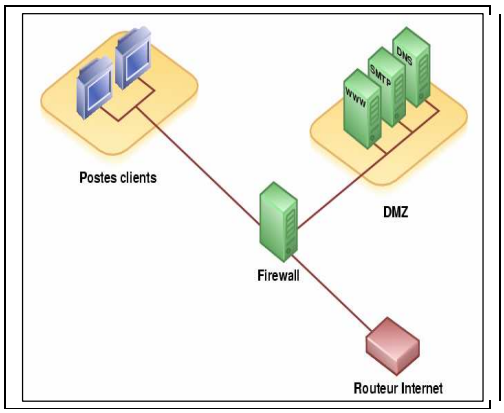


FIGURA 1.6 CONEXIÓN DMZ A TRAVÉS DE UN RUTEADOR. (FUENTE: WIKIPEDIA)

1.9.12 SERVICIOS QUE FUNCIONAN TRAS UN FIREWALL

Quizá los servicios más comunes que deben ser accesibles desde Internet y que por tanto son más vulnerables a ataques son el servicio de correo electrónico (SMTP), el servicio Web (HTTP) y el servicio de resolución de nombres (DNS).

Esto implica que el sistema de firewall deba permitir únicamente el tráfico correspondiente a estos servicios mediante la apertura de los puertos a través de los que ellos se manejan.

El *firewall* debe permitir el envío y recepción de correo electrónico. Esto generalmente significa que debe permitir el reenvío de correo a través de una pasarela, y la recepción de correo a través de protocolos como POP e IMAP mediante la habilitación de los puertos de comunicación correspondientes, en el caso del reenvío, el puerto 25 (SMTP) y en recepción el 110 para POP.

El *firewall* igualmente deberá permitir el paso de solicitudes DNS originadas en las estaciones de la red interna y así también, en el caso de que la empresa posea su propio servidor DNS que atienda peticiones externas, estas peticiones deberán atravesar el firewall.

La compañía puede decidir si ofrecer información al Internet y/o la Intranet, en principio utilizando un servidor Web. Este servidor se coloca usualmente en la DMZ, desde donde puede accederse para los usuarios de Internet y la Intranet.

El servidor Web de la Intranet es entonces colocado dentro de ella, donde sólo los clientes de la intranet pueden conectarse.

De nuevo, los *routers* de filtrado de paquetes necesitan ser configurados para permitir las peticiones entrantes, esta vez para el puerto 80.

1.9.13 VPN (RED PRIVADA VIRTUAL)

Las VPNs son aplicaciones recientes en un firewall usadas para comunicar de forma segura y transparente con otra delegación de la misma compañía, o con otra compañía utilizando el Internet como un medio de comunicaciones.

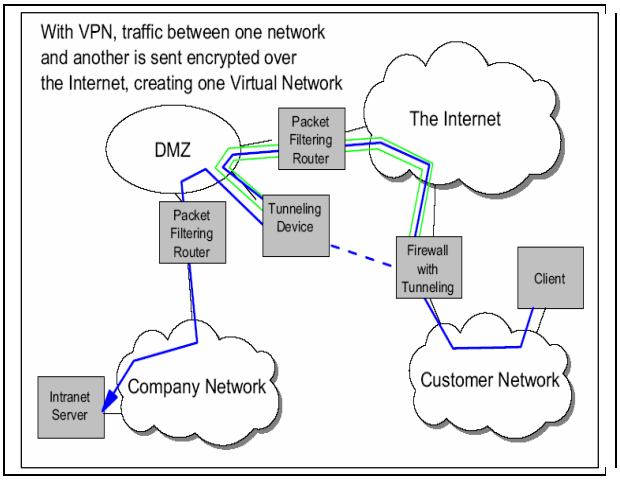


FIGURA 1.7 TRÁFICO PARA UNA VPN A TRAVÉS DE UN FIREWALL

Hay un gran inconveniente sin embargo, y es que es una necesidad la de mantener nuestra información de forma confidencial. Esta confidencialidad está garantizada a través del uso de técnicas de encriptación avanzadas. Cuando un cliente en una red de consumidores desea recuperar alguna información de su servidor de Intranet por ejemplo, éste envía la petición al firewall que incorpora un dispositivo que hace de túnel y que encripta las peticiones y las envía al dispositivo de "*tunelling*" en el otro firewall.

La petición es luego desencriptada y pasada al servidor. La información en la vía de regreso al cliente es encriptada de la misma forma. La encriptación/de-encriptación es completamente transparente para el cliente y el servidor.

1.9.14 PROXY

En lo referente a los sistemas de computación un servidor proxy es un dispositivo o programa que realiza una acción en representación de otro.

Este estándar trabaja de la forma siguiente:

- El cliente establece una conexión con el servidor proxy, usualmente en el puerto 8080.
- El cliente luego envía su petición al servidor proxy, comúnmente en forma de una solicitud HTTP, aún cuando el archivo requerido será descargado usando FTP. El cliente puede también especificar opciones HTTP.
- El servidor proxy entonces reenvía la petición al destino en el Internet y recupera la información correspondiente. Esto se puede realizar usando los protocolos HTTP, HTTPS, FTP, WAIS o protocolos *gopher*, en definitiva trabaja con un limitado grupo de protocolos correspondientes a la capa de aplicación de la pila TCP/IP.
- La información es luego enviada al cliente como resultado de la consulta HTTP.

Un servidor proxy suele ser considerado como un dispositivo de firewall de capa de aplicación en cuyo nivel puede filtrarse el tráfico de manera más granular, basándonos en características específicas de las páginas web (en el caso de HTTP) como la URL, los componentes, permitiendo un registro de eventos y control de acceso granular, mucho más fino. Es posible realizar caché transparente, reduciendo el ancho de banda y acelerando los tiempos de respuesta. Una desventaja con respecto a NAT es que resulta ser más lento si las peticiones no se almacenan en el caché del proxy, además de que trabaja con unos pocos protocolos específicos.

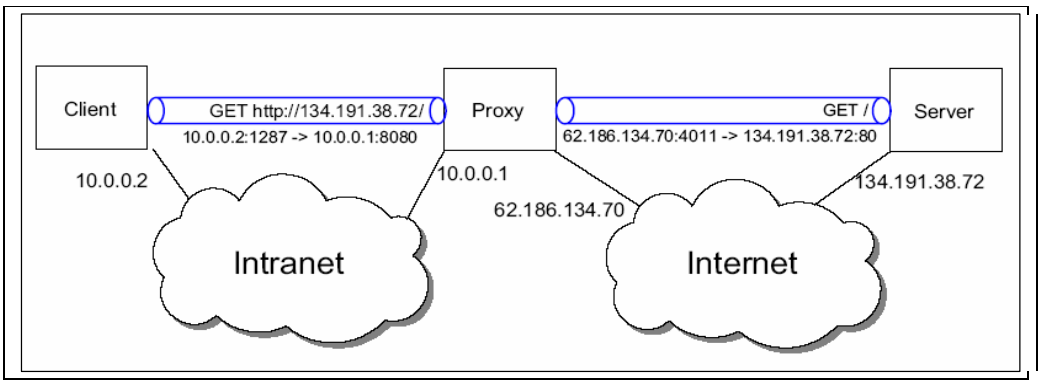


FIGURA 1.8 PROTOCOLO PROXY

1.9.15 VISIÓN GENERAL DEL ESQUEMA COMPLETO

Un firewall, como se mencionó, no es un único dispositivo por sí mismo, sino que contiene un número de componentes, lo que no significa que todos estos componentes puedan concentrarse en un host único.

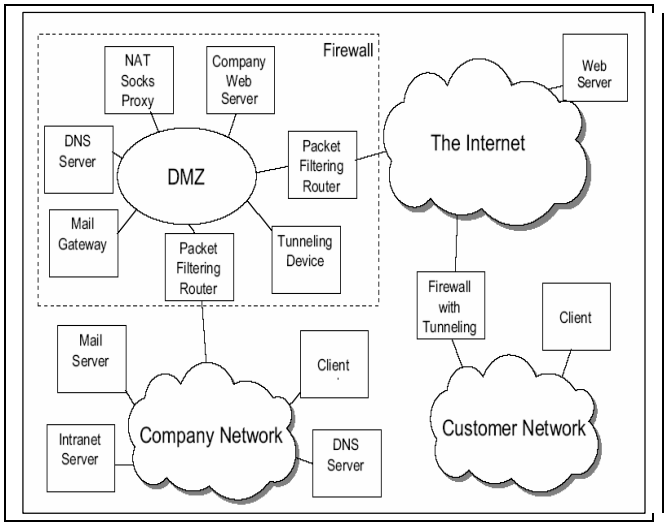


FIGURA 1.9 ESQUEMA DE FUNCIONAMIENTO DEL FIREWALL PARA EL TRÁFICO DE RED.

1.9.16 VULNERABILIDADES

1.9.16.1 Generalidades

Un firewall no es un dispositivo que brinde protección frente a todo y hay algunas cosas frente a las cuales presenta ranuras de seguridad.

Un firewall no puede proteger en contra del mal uso de conexiones permitidas. Por ejemplo, suponiendo que un cortafuegos permite las conexiones HTTP salientes, esta política puede ser utilizada para transferir paquetes IP hacia y

desde un sistema externo. Dado que el firewall generalmente no tiene una forma de verificar los contenidos de la conexión HTTP, esta irregularidad no se detecta. Afortunadamente el mal uso de conexiones permitidas no es posible sin la ayuda desde la red Interna.

Un firewall sólo nos protege de los ataques desde el exterior, es decir desde el Internet. Cualquier usuario interno no podrá ser detenido por el firewall pues tiene acceso a los sistemas de la compañía directamente.

La información que viaja hacia el Internet tampoco puede ser protegida, pues cualquiera puede capturarla e incluso alterarla.

Todas las conexiones que no atraviesen el firewall obviamente no están bajo su control, esto incluye conexiones vía módem desde y hacia sistemas en la red Interna, pero incluye también software y/o información que es traída/llevada dentro y fuera de la compañía en medios físicos (diskettes, CD ROMs, memorias USB).

Obviamente un firewall es incapaz de proteger los sistemas de robos, desastres naturales, y otros agentes externos.

1.9.16.2 Vulnerabilidades con algunos Protocolos de Administración

Muchos protocolos de administración comúnmente usados, como Telnet, SNMP (*Simple Network Management Protocol*) y *syslog* no se diseñaron como protocolos seguros. Estos protocolos transmiten información en texto plano que un atacante puede fácilmente interceptar. Así también, algunos protocolos de administración no poseen integridad a nivel de paquetes, lo que permite que un hacker pueda cambiar fácilmente la información contenida en los paquetes sin que seamos si quiera conscientes de que los datos fueron cambiados.

Telnet.- Toda la información de Telnet se transmite en texto plano, por esta razón, si estamos configurando un dispositivo remoto, cualquier hacker que esté haciendo *sniffing* de nuestra red podrá capturar todas las contraseñas y cuentas que utilizamos para la mencionada configuración.

SNMP.- Es un protocolo muy útil para la administración de red pues brinda la habilidad de monitorear dispositivos remotamente. Podemos usarlo igualmente para configurar dispositivos de red como ruteadores, y estos dispositivos de red

pueden, de forma proactiva, enviarnos información basada en nuestros requerimientos. Sin embargo, SNMP no fue diseñado bajo parámetros de seguridad pues envía información en texto plano, de modo que es vulnerable a comprometer esta información.

Si es necesaria la utilización de SNMP en una red, se debe asegurar que se emplea la versión 3 o superior, que contiene mecanismos de seguridad que impiden a un hacker ver la información que es transmitida por este protocolo.

TFTP.- TFTP (*Trivial File Transfer Protocol*) tiene la misma debilidad que los protocolos antes descritos en el sentido de que transmiten su información en texto plano. Una técnica bastante sencilla para asegurar TFTP es enviar sus paquetes a través de túneles IPSec, que proveen la confidencialidad necesaria para mantener privada nuestra información sensible.

NTP.- NTP o Protocolo de Tiempo de Red (Network Time Protocol) asegura que los dispositivos tengan configurados la hora adecuadamente. Un ataque DoS simple cambia la hora de los equipos de modo que la autenticación de certificados digitales falla porque, por ejemplo, cuando estos certificados se reciben ya están caducados.

Para evitar que un hacker cambie la hora de un dispositivo servidor o manipule los paquetes NTP, es necesario utilizar una versión de este protocolo que soporte mecanismos de autenticación criptográfica, como el caso de la versión 3.

SSH.- El uso de SSH es altamente recomendado cuando se configura dispositivos de red de manera remota. SSH encripta los paquetes que son enviados entre dos hosts; por tanto, esos paquetes no son sujetos de alteración.

SSL.- Muchas de las recientes aplicaciones de administración de redes basadas en GUI (interfaz gráfica) soportan el uso de SSL. Las compras seguras a través de Internet se realizan mediante el uso de este protocolo, pues garantizan la confidencialidad de la información transmitida entre un host y un servidor.

1.10 PILA DE PROTOCOLOS TCP/IP

La pila de protocolos TCP/IP consiste de un conjunto de protocolos, que estructurados uno sobre otro, ofrecen al usuario varios servicios diferentes. En el

diagrama siguiente se muestran las capas que conforman este conjunto de protocolos.

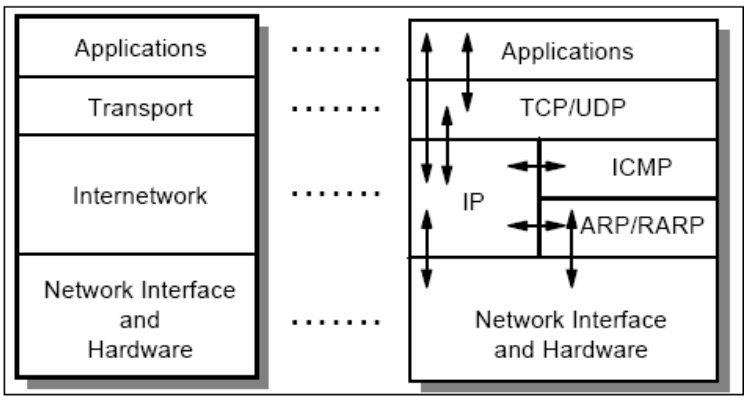


FIGURA 1.10 PILA DE PROTOCOLOS TCP/IP

El protocolo IP (Protocolo de Internet) es la base de la pila de protocolos TCP/IP, y su tarea principal es enviar información desde un host a otro, encapsulando esta información en un “datagrama” y adjuntando una cabecera en donde se guardan algunos detalles acerca de la información como la dirección origen y destino.

Como ya se mencionó, el protocolo IP provee un servicio no fiable, no orientado a conexión, lo que significa que la capa IP hace el mayor esfuerzo para entregar la información al host destino sin ofrecer ninguna garantía.

Los paquetes de datos pueden perderse en la transmisión, pueden duplicarse o su orden puede alterarse. Depende de los protocolos de mayor nivel agregar confiabilidad al detectar los paquetes perdidos y reenviarlos. Las subcapas IP no están “concientes” de la existencia de conexiones y tratan cada paquete IP individualmente y no mantiene información de estado acerca de conexiones, pues esto está a cargo de las capas superiores.

La IANA reserva algunos rangos de direcciones para propósitos especiales. Entre estos rangos podemos contar:

- El rango de clase A 10.0.0.0, los rangos de clase B 172.16.0.0 a 172.31.0.0 y los rangos de direcciones de clase C 192.168.0.0 hasta 192.168.255.0. Estas direcciones no deben ser asignadas a hosts en Internet, y pueden en cambio ser usadas en redes privadas que no están directamente conectadas a Internet.

- El rango de direcciones de clase A 127.0.0.0 que es reservado para dispositivos de *loopback* (interfaces virtuales que representan el host local).

Un hacker usualmente tiene herramientas disponibles para crear paquetes IP arbitrarios y con estas herramientas es capaz de crear cualquiera de los campos de la cabecera IP a su antojo. Por ejemplo:

- El campo longitud total, en combinación con una cantidad correcta de información, puede forzar a que un ruteador realice fragmentación. Esto puede usarse para sobrecargar los búferes, en pilas TCP/IP pequeñas.
- Las cabeceras que gobiernan la fragmentación pueden ser usadas para enviar paquetes que parecen ser fragmentos de un gran paquete IP hacia un host, Debido a que estos paquetes necesitan ser almacenados antes de reensamblarse, el búfer puede fácilmente sobrecargarse.
- La dirección IP origen puede ser borrada (*spoofing*), de modo que un ataque no podrá ser informado en el origen de la transmisión.

1.10.1 PROTOCOLO ICMP (INTERNET CONTROL MESSAGE PROTOCOL)

El protocolo ICMP o Protocolo de Control de Mensajes de Internet es mayoritariamente *utilizado para reportar errores en el protocolo IP*, convirtiéndose en un subprotocolo de diagnóstico y notificación. Es decir, permite enviar mensajes de error indicando que un servicio determinado no está disponible o que un host no puede ser alcanzado, por ejemplo.

ICMP se construye sobre el protocolo IP, lo que significa que utiliza IP como su mecanismo de transporte.

Algunos de los ataques DoS más comunes usan ICMP en varias de sus formas.

Un ataque *smurf,* envía un ping (*echo request*) a una dirección de host o de broadcast, pero especifica una dirección de broadcast en la red local como una dirección IP origen. EL mensaje de respuesta del ping (*echo reply*) será recibido por todas las estaciones en la red, consumiendo el tiempo de CPU y el ancho de banda. Si muchos de estos pings se reciben de esta manera, la red completa puede colapsar e impedir el tráfico de información.

El paquete ICMP de “destino inalcanzable” (*destination unreachable*) usualmente no es enviado por hackers, pero si es ansiosamente esperado por ellos, pues éste paquete le dice al hacker si un servicio dado está corriendo o si está funcionando detrás de un filtro. Esto resulta peligroso ya que esta información puede ser utilizada para afinar la estrategia del hacker, permitiéndole incluso deducir reglas del firewall.

El paquete *Source Quench* puede ser usado para ataques DoS mediante la obstrucción efectiva de todo el ancho de banda hacia un determinado destino.

El paquete ICMP *Redirect* puede ser utilizado para influir en las tablas de enrutamiento de tal manera que se puede lograr que determinado tráfico sea enrutado de forma diferente, utilizando otra ruta, en el caso de que la que haya estado utilizando haya sido una ruta monitoreada por un hacker.

1.10.2 PROTOCOLO UDP (*USER DATAGRAM PROTOCOL*)

El protocolo UDP es un protocolo de la capa de transporte basado en el intercambio de datagramas. Estos datagramas se envían a través de la red sin el establecimiento previo de una conexión, por lo que comúnmente se dice que *no es orientado a conexión*. Otra característica típica es que no posee un mecanismo de confirmación ni de control de flujo, razón por la que es propenso a la presencia de errores en la transmisión de la información.

Este protocolo es una extensión simple del protocolo IP, en el que se agrega números de puerto de 16 bits para identificar el servicio hacia el que la información debe viajar.

Para un hacker, sólo el puerto origen y el puerto destino son interesantes. Con el puerto destino puede seleccionar el servicio al que desee conectarse. El puerto origen es normalmente seleccionado de forma randómica, pero para algunos protocolos un puerto origen menor que 1024 es considerado un puerto “seguro” que ofrece más privilegios que un puerto normal.

UDP no tiene soporte de “*pacing*”. El *pacing* es un proceso en el que quien recibe la información indica cuánto del espacio de búfer está disponible y, por lo tanto, cuánto de información está permitido enviar.

La falta de esta característica significa que es perfectamente legal enviar paquetes UDP tan rápidamente como nuestra red nos permita. Esto es muy usado en ataques DDoS en donde la red a la que pertenece el servidor se llega a sobrecargar.

1.10.3 PROTOCOLO TCP

El protocolo TCP es el más comúnmente usado. Así como UDP, usa IP como su mecanismo de transporte y un sistema de puertos para seleccionar el servicio o aplicación dentro de una misma máquina.

La diferencia con UDP es que TCP ofrece un servicio fiable, orientado a conexión. Esto significa que el protocolo garantiza que los paquetes serán entregados en su destino sin errores y en el orden en que fueron enviados.

Los puertos origen y destino trabajan de la misma forma que los correspondientes a UDP: el puerto destino determina el servicio. El puerto origen puede ser objeto de *spoofing* por debajo de 1023 para simular un puerto conocido o seguro.

El número de secuencia es, de hecho, el objetivo del hacker y para que pueda lograr colarse en las conexiones existentes, un hacker debe predecir los números de secuencia que se usarán. Mediante el uso de números de secuencia randómicos esto se vuelve virtualmente imposible.

Linux y otros sistemas operativos, usan números de secuencia verdaderamente randómicos, pero muchos otros no lo hacen, lo que los convierte en sistemas vulnerables a ataques de *spoofing*.

La bandera SYN es usada en ataques llamados ataques SYN, en cuyo caso un gran número de paquetes SYN son enviados, lo que aparece ante el servidor como si una gran cantidad de conexiones se han abierto, por lo que el servidor se encarga de reservar los recursos para estas conexiones. Sin embargo, el paquete que regresa es ignorado por el hacker, dejando las conexiones “medio abiertas”. Esto inmoviliza los recursos y podría causar que un servidor pobremente diseñado colapse.

Una opción del kernel Linux llamada “*SYN cookies*” previene este ataque en momentos de gran carga, no alojando recursos para una conexión cuando el

primer paquete de establecimiento de conexión arriba. En lugar de eso, la información importante es almacenada en el número de secuencia y es criptográficamente protegida. Los recursos se alojan solamente cuando el tercer paquete retorna con el correcto número de acuse de recibo.

1.11 FILTRADO DE PAQUETES Y NAT (*NETWORK ADDRESS TRANSLATION*)

1.11.1 FILTRADO DE PAQUETES

El filtrado de paquetes es, como puede intuirse, el paso o bloqueo selectivo de paquetes de datos que circulan a través de una interfaz de red.

La capacidad de filtrado de paquetes a través de un mecanismo de enrutamiento, mediante el cual transite la información, es muy importante en el momento de reducir la carga de la red, descartando paquetes cuyo tiempo de vida ha expirado, paquetes erróneos, o simplemente tramas de broadcast. Esto es posible gracias al acceso que un dispositivo puede tener a los paquetes IP, de tal manera que es capaz de modificar la información de sus campos (capa 3 y capa 4) permitiendo que el tráfico se comporte conforme a un objetivo de gestión de la red.

Indudablemente, el filtrado de paquetes puede permitir la implementación de diferentes políticas de seguridad en una red, evitando el acceso no autorizado a la misma y permitiendo el acceso autorizado.

1.11.2 FUNCIONAMIENTO

El filtrado de paquetes funciona de una manera bastante simple, en contraposición a la dificultad que en ocasiones significa la implementación de sus reglas.

Básicamente se analiza la cabecera de cada paquete, y en función de una serie de reglas establecidas con anterioridad, la trama es bloqueada o se permite su paso. Estas reglas pueden contemplar campos como el protocolo utilizado (TCP, UDP, ICMP), las direcciones IP o MAC fuente o destino, el puerto fuente o destino. Esto nos indica que el firewall debe ser capaz de trabajar en los niveles de red (para que pueda discriminar entre las redes IP origen y destino) y de transporte (para hacerlo en función de los puertos usados). En algunas

aplicaciones es posible que estos dispositivos de enrutamiento (firewall) permitan el filtrado de paquetes especificando reglas basadas en la interfaz del dispositivo por donde se reenviará el paquete e incluso la interfaz por la que ha llegado a él.

1.11.3 CRITERIOS DE FILTRADO

Entre los criterios en que se puede basar el filtrado están:

- La interfaz de red por la que el paquete arriba o por la que sale.
- El protocolo especificado en el paquete IP (UDP, TCP, ICMP o cualquier otro que utilice IP).
- La dirección IP origen y/o destino.
- El puerto TCP/UDP origen y/o destino.
- El sentido de establecimiento de conexión.
- La existencia de una conexión TCP del que el paquete afirma ser parte.
- El tipo de paquete ICMP.
- La dirección MAC.
- El ID de usuario del proceso emisor/receptor.

Basado en estas reglas de filtrado, hay dos acciones principales que pueden ejecutarse:

- Permitir el paquete
- Descartar el paquete

1.11.4 ESPECIFICACIÓN DE LAS REGLAS

Las reglas se expresan como una simple tabla de condiciones y acciones que se consulta en orden hasta encontrar una regla que permita tomar una decisión sobre el bloqueo (o descarte) o el reenvío de la trama. Este proceso puede incluir implementaciones que permiten indicar si el bloqueo de un paquete se notificará a la máquina origen mediante un mensaje ICMP.

Es importante *tomar en cuenta el orden* de análisis de las tablas para poder implementar la política de seguridad de una forma adecuada, pues mientras más complejas sean las reglas y su orden de análisis, será más complicado para el administrador comprenderlas.

Citando un ejemplo de tabla de reglas de filtrado podemos tener:

Origen	Destino	Tipo	Puerto	Acción
155.43.0.0	*	*	*	Deny
*	192.53.22.0	*	*	Deny
155.42.0.0	*	*	*	Allow
*	192.22.34.0	*	*	Deny

El sistema de reglas basado en la tabla anterior nos indica que si llega un paquete proveniente de la dirección 155.43.0.0, éste será descartado sin importar cuál es su destino, su tipo o el puerto que utiliza. Asimismo, cualquier paquete que se dirija a 192.53.22.0 será descartado.

Sin embargo, y para ilustrar la complejidad este sistema de reglas podría implicar, imaginemos que llega un paquete de un sistema de la red 155.42.0.0 hacia 192.22.34.0; si analizamos la tercera regla desde la parte superior, ésta nos indica que dejemos pasar el tráfico proveniente de 155.42.0.0 y la comprobación de reglas termina en esa línea. Si, en cambio, se verifican las reglas desde la parte inferior, el mismo paquete, al tener como destino la red 192.22.34.0, será descartado.

Este ejemplo nos da una idea del cuidado que se debe poner al diseñar la estructura de reglas, de una forma tal que no se contrapongan.

Un aspecto importante a tomar en cuenta es la decisión a tomarse cuando ninguna de las reglas se ha comprobado positivamente. En esos casos es necesario definir una regla por defecto al final de la lista, que generalmente se encarga de impedir el tráfico de los paquetes cuya información no encajó con ninguna de las reglas anteriores.

1.11.5 NETWORK ADDRESS TRANSLATION (NAT)

El crecimiento de Internet ha superado todas las expectativas iniciales, pues el número de usuarios alcanza fácilmente los varios cientos de millones, superando incluso la población de los Estados Unidos. Es tal la magnitud de crecimiento que éste número se va duplicando año a año.

NAT es un estándar cuyo mecanismo permite, entre otras cosas, enfrentar el inconveniente de la escasez de direcciones IP mediante el manejo de direcciones privadas.

NAT es también conocido como enmascaramiento IP y provee el mapeo entre direcciones IP internas (o privadas) y direcciones IP externas asignadas de forma oficial (públicas). Esto resulta sumamente útil debido al hecho de que es posible mantener una estructura de direccionamiento privado al interior de la empresa y acceder al Internet a través de una o varias direcciones IP públicas. Esto es necesario por el hecho de que no es posible (o no suele permitirse) que paquetes con direcciones IP (origen o destino) privadas puedan enrutarse a través de Internet. Debido a la escasez de direcciones IP públicas, causada por el crecimiento acelerado de usuarios, es muy común utilizar una o muy pocas direcciones IP públicas para acceder a Internet desde la red interna de la empresa.

NAT está definido en el RFC 3022 y su discusión está en el RFC 2663.

1.11.5.1 Funcionamiento

NAT ha representado una magnífica solución temporal al problema de agotamiento de las direcciones, al permitir que varios usuarios se conecten a la red, compartiendo una única dirección IP.

El protocolo TCP/IP es capaz de generar varias conexiones simultáneas con un dispositivo remoto. Como se mencionó, en la cabecera de un paquete IP existen campos en los que se indica la dirección origen y destino, con sus respectivos puertos. Esta combinación de números define una conexión única.

Básicamente, un gateway (pasarela) NAT cambia la dirección origen en cada paquete de salida y, dependiendo del método, también el puerto origen para que sea único.

Estas traducciones de dirección se almacenan en una tabla, de manera que sea posible recordar qué dirección y puerto le corresponde a cada dispositivo cliente y de ese modo saber a dónde deben regresar los paquetes de respuesta.

Si un paquete que intenta ingresar a la red interna no existe en la tabla de traducciones, éste es descartado. Este comportamiento permite definir en la tabla que en un determinado puerto y dirección se pueda acceder a un determinado dispositivo, por ejemplo un servidor web, lo que se denomina NAT inverso o DNAT (*Destination NAT*).

Se puede comprender, por tanto, que a parte de permitir la conexión de varios equipos a Internet a través de una sola dirección pública, el hecho de "enmascarar" los paquetes IP, mediante el cambio de información de sus campos, le dota a nuestra red interna de un nivel de seguridad adicional, por el simple hecho de cambiar la cabecera IP de la forma que se mencionó.

Como se puede apreciar, NAT está completamente separado del filtrado de paquetes IP, pero se implementa en Linux a través de la misma herramienta. NAT involucra el cambio de direcciones IP y si es necesario los números de puerto que están especificados en el paquete.

1.11.5.2 Formas de NAT

Generalmente hablando, existen 2 formas de NAT.

- *Source NAT*
- *Destination NAT*

Con *Source NAT* (SNAT), la dirección IP origen y, si es necesario, el puerto origen son cambiados. Esto generalmente es denominado como Enmascaramiento IP y es comúnmente utilizado en un firewall para dar a los clientes internos (con una dirección IP pública) acceso a servidores fuera del firewall. Para el servidor involucrado, la conexión parecerá provenir del firewall mismo, ya que la dirección IP origen del paquete es cambiada a la dirección IP del firewall mientras el paquete atraviesa el firewall.

Con *Destination NAT* (DNAT), la dirección IP destino y, si es necesario, el puerto destino son cambiados.

Hay básicamente dos aplicaciones para esto.

- *Port Forwarding* (envío de puerto), en el que un paquete IP que va dirigido a un puerto específico en el firewall mismo es reenviado a una dirección IP

en la red interna. Esto resulta útil si, por ejemplo, deseamos colocar nuestro servidor Web dentro del dispositivo de firewall sino en nuestra red interna. Para el mundo externo, aparecerá como si nuestro servidor Web está funcionando en el firewall.

- Con *Proxificación Transparente,* un paquete interno que se origina en la red interna y tiene como destino algún host en Internet es enrutado a un puerto local en el firewall. En este puerto nosotros corremos típicamente un servidor proxy. Esto permite a los clientes en la red interna una conexión transparente hacia Internet, es decir sin advertir que está utilizando un proxy.

Algo bastante relacionado con NAT es el *packet mangling*, que es un proceso en el que otros campos del paquete TCP/IP, como los de prioridad o TTL, son cambiados.

1.12 COMPONENTES Y ARQUITECTURA DE UN FIREWALL

1.12.1 COMPONENTES

A partir del estudio realizado con anterioridad se puede determinar 3 componentes fundamentales en el diseño de un sistema de firewall:

Filtrado de Paquetes.- O un proceso de implementación de reglas de filtrado, a través de las que se implementan políticas de seguridad en una red, cuyo objetivo es evitar el acceso no autorizado sin descartar el acceso autorizado.

Proxy de Aplicación.- O aplicaciones de software que permiten reenviar o bloquear conexiones a servicios como FTP o HTTP. El dispositivo servidor sobre el que corre esta aplicación es conocido como pasarela o gateway.

*Monitoreo de Actividad.-*El monitoreo o registro de los eventos que detecta el firewall es de suma importancia para la seguridad de la red, especialmente porque es un medio de determinar información sobre intentos de ataque o la existencia de tramas sospechosas.

1.12.2 ARQUITECTURAS

La arquitectura de un firewall supone la ubicación de los diferentes dispositivos encargados de filtrar el tráfico, enrutarlo, rechazarlo de manera que provea de un nivel de seguridad a la red de la organización.

1.12.2.1 Arquitectura *Dual-Homed Host*

Una arquitectura *dual-homed host* es construida alrededor de una computadora que posee al menos dos interfaces de red. Este host podría actuar como un *router* entre las redes a las que estas interfaces están conectadas; es capaz de enrutar paquetes IP de una red a otra.

Sin embargo, para implementar una arquitectura *dual-homed host* para un *firewall*, esta función de enrutamiento debe ser deshabilitada. Por tanto, los paquetes IP desde una red (por ejemplo el Internet) no son directamente enrutados a otra red (por ejemplo la red interna protegida). Los sistemas al interior del firewall pueden comunicarse con el host *dual-homed* y los sistemas fuera del firewall (en el Internet) pueden también comunicarse con el host *dual-homed*; sin embargo, estos sistemas no pueden comunicarse directamente entre sí. El tráfico IP entre ellos está completamente bloqueado.

La arquitectura de red para un host-firewall *dual-homed* es bastante simple, el host se conecta a través de una interfaz diferente con la red interna y con el Internet.

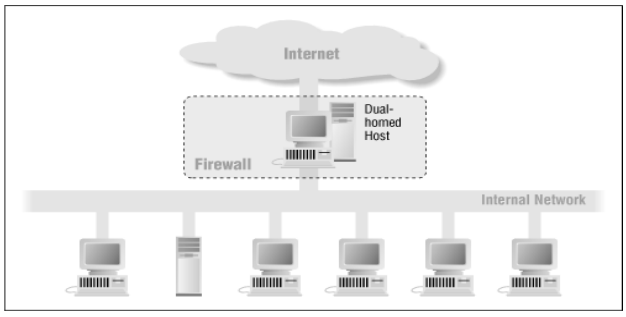


FIGURA 1.11 ARQUITECTURA DUAL-HOMED HOST

Un sistema de host *dual-homed*, por tanto, podrá únicamente proveer servicios mediante la proxificación de los mismos o permitiendo que los usuarios se logueen directamente en el host dual.

La proxificación es mucho menos problemática, pero podría no estar disponible para todos los servicios en los que estamos interesados.

1.12.2.2 Arquitectura *Screened Host*

Aun cuando una arquitectura de host *dual-homed* provee servicios desde un host que está conectado a múltiples redes (pero tiene enrutamiento deshabilitado), una arquitectura de *host screened* provee servicios desde un host que está conectado solamente a la red interna, utilizando un ruteador separado.

En esta arquitectura, la seguridad primaria es provista por el filtrado de paquetes. (Por ejemplo, el filtrado de paquetes permite que se realice una conexión directa, en lugar de utilizar los servidores de poxificación).

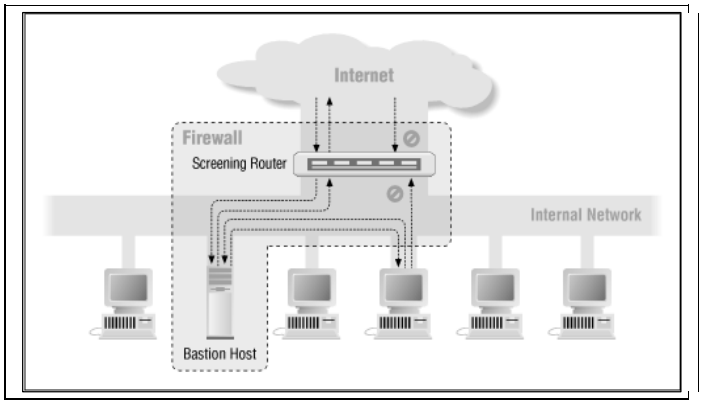


FIGURA 1.12 ARQUITECTURA SCREENED HOST

Como se puede observar, un host bastión (un host que ofrece un servicio y debe estar accesible para el Internet desde la red interna) se ubica en una red interna y el filtrado de paquetes en el ruteador que hace de pantalla se conecta de tal manera que el host bastión es el único sistema en la red interna en el que los hosts en el Internet pueden abrir conexiones (por ejemplo para entregar correo entrante). Aún entonces, solamente ciertos tipos de conexiones son permitidas. Cualquier sistema externo que intente acceder a servicios o sistemas internos tendrán que conectarse a este host. El host bastión necesitará, como consecuencia, mantener un alto nivel de seguridad.

El filtrado de paquetes le permite también al host bastión abrir conexiones permitidas al mundo externo.

La configuración del filtrado de paquetes en el ruteador de pantalla, puede ejecutar una de las siguientes acciones:

- Permitir a otros hosts internos abrir conexiones a hosts en el Internet para ciertos servicios (permitiendo esos servicios mediante el filtrado de paquetes.
- No permitir ninguna conexión desde los hosts internos (forzando a estos hosts a usar servicios de proxificación a través del host bastión).

Se puede mezclar y combinar estos enfoques para diferentes servicios; algunos pueden permitirse directamente mediante filtrado de paquetes, mientras que otros podrán permitirse indirectamente a través de un proxy. Todo esto depende de la política en particular que se esté intentando implementar.

Debido a que esta arquitectura permite a los paquetes moverse desde el Internet hacia las redes internas, podría parecer más riesgoso que la arquitectura *dual-homed*, que está diseñada de tal manera que ningún paquete pueda alcanzar la red interna. En la práctica, sin embargo, la arquitectura *dual-homed* es susceptible a fallas que permiten a los paquetes cruzar desde la red externa a la interna, pues debido a que este tipo de falla es completamente inesperada, no existen protecciones contra este tipo de fallas.

La mayor desventaja de este tipo de arquitecturas es que si un hacker logra acceder al sistema de host bastión, no hay nada que podamos hacer en cuanto a seguridad de la red entre el host bastión y el resto de los hosts internos.

El *router* presenta también un único punto de falla; si éste se ve comprometido, la red entera está en las manos del hacker.

1.12.2.3 Arquitectura *Screened Subnet*

Esta arquitectura agrega una capa extra de seguridad a la arquitectura *screened host* agregando una red de perímetro que aísla mucho más la red interna del Internet.

Aislando el host bastión en una red perimetral, es posible reducir el impacto de un acceso no autorizado dentro de él. El hacker ya no da en el blanco instantáneamente, gana algo de acceso, pero no a todo.

Con el tipo más simple de arquitectura *screened subnet*, hay dos ruteadores que hacen de pantalla (o muralla –*screening routers*), cada uno conectado a la red perimetral. Uno de esos ruteadores se ubica entre la red perimetral y la red externa, mientras que el otro se ubica entre la red perimetral y la red interna (usualmente el Internet). Para alcanzar la red interna con este tipo de arquitectura, un hacker tendría que atravesar los dos ruteadores. Aun si el atacante irrumpe de alguna forma en el host bastión, todavía tendrá que vulnerar el ruteador interno, lo que significa que ya no existe un único punto vulnerable que comprometerá la red interna.

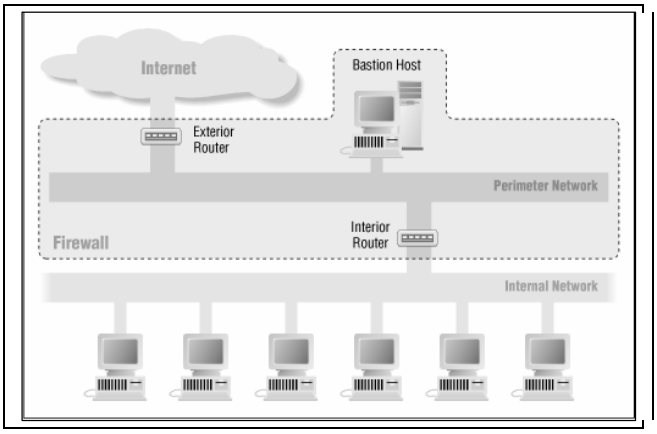


FIGURA 1.13 ARQUITECTURA SCREENED HOST, UTILIZANDO DOS RUTEADORES.

A continuación se analizan algunos de los componentes de esta arquitectura:

Red Perimetral

La red perimetral constituye otra capa de seguridad, una red adicional entre la red externa y nuestra red interna protegida.

Host Bastión

Con la arquitectura *screened subnet*, nosotros conectamos uno o varios hosts bastión a la red perimetral; este host es el punto principal de contacto para las conexiones entrantes desde el Internet, por ejemplo:

- Para que las sesiones de correo electrónico entrante (SMTP) permitan la entrega de correo.
- Para conexiones entrantes FTP hacia el servidor FTP anónimo.

- Para consultas DNS (Servicio de Nombres de Dominio)

Algunos de los servicios salientes (desde clientes externos hacia servidores en el Internet) se manejan en alguna de las formas siguientes:

- Filtrado de paquetes en los *routers*.
- Configuración de proxy para acceso indirecto de la intranet al Internet.

Router Interior

El ruteador interior protege la red interna del Internet y de la red perimetral.
El ruteador interior ejecuta la mayoría del filtrado de paquetes de un firewall. Permite servicios salientes seleccionados desde la red interna hacia el Internet.

Router Exterior

Llamado también ruteador de acceso, protege la red perimetral y la interna del Internet. En la práctica, los ruteadores exteriores tienden a permitir casi todo lo que salga de la red perimetral y generalmente hacen muy poco filtrado de paquetes. Las reglas de filtrado de paquetes para proteger las máquinas internas necesitarían ser esencialmente las mismas tanto en el ruteador interno como en el externo; si hay algún error en las reglas que permita el acceso a un hacker, el error probablemente se presentará en los dos ruteadores.
Con frecuencia, el ruteador exterior es colocado por un agente externo (por ejemplo nuestro ISP), y nuestro acceso a él suele ser limitado. Un agente externo que mantiene un ruteador probablemente colocará unas pocas reglas de filtrado general de paquetes, pero no estará dispuesto a mantener un conjunto de reglas variable y complicado.

1.13 FIREWALL EN LINUX (IPTABLES[5])

1.13.1 INTRODUCCIÓN

IPtables en Linux es el sistema de firewall al que está vinculado el núcleo de Linux y que se ha extendido enormemente a partir de la versión 2.4 de su kernel.

[5] http://www.netfilter.org, "Firewalling, NAT and Packet Mangling for Linux"

De la misma forma que el sistema anterior, *ipchains,* iptables no es un servicio que pueda caerse por un error de programación y que podamos iniciarlo o detenerlo. En general, nunca representará tanto peligro como las aplicaciones que escuchan en determinado puerto TCP. Iptables está integrado en el kernel, por lo que es parte del sistema operativo, y se pone en marcha mediante la aplicación de reglas utilizando un conjunto de comandos, a través de los que se añaden, borran o crean estas reglas.

Por todo esto, un firewall de iptables no es otra cosa que un script de shell en el que se van ejecutando las reglas de firewall.

Aun cuando *iptables* es técnicamente sólo la herramienta que controla el filtrado de paquetes y los componentes de NAT dentro del kernel, el término *iptables* se utiliza muchas veces para referirse a toda una infraestructura, incluyendo *netfilter* y *connection tracking* (seguimiento de conexiones) y NAT, así como a la herramienta propiamente dicha.

1.13.2 BREVE HISTORIA

Junto con *netfilter* (un conjunto de módulos dentro del núcleo Linux que interceptan y manipulan paquetes de red), iptables, inició en 1998 como renovación del proyecto precedente *ipchains.*

Antes de la aparición de *iptables*, los programas más utilizados para la estructuración de firewalls en linux eran *ipchains* en el núcleo versión 2.2 e *ipfwadm* en el núcleo 2.0, que se basaba en *ipfw* de BSD.

Estos dos programas alteraban la codificación de red para poder manipular los paquetes, pues no existía un *framework* general para el manejo de paquetes hasta que apareció *netfilter*. *Iptables* mantiene la idea básica de *ipfwadm*, es decir un conjunto de listas de reglas mediante las que se especifica qué hacer con cada paquete y en función de qué parámetros. *ipchains* en cambio, implementa el concepto de *cadenas de reglas* mientras que iptables extendió este concepto a la idea de tablas; es decir, se consultaba una tabla para decidir si había que NATear un paquete, mientras que se consultaba otra para decidir cómo filtrar un paquete.

Es importante destacar que mientras *ipchains* e *ipfwadm* combinan filtrado de paquetes y NAT (esencialmente tres tipos de NAT: enmascaramiento de IP,

redireccionamiento de puertos y redirección), *netfilter* permite separar las operaciones sobre los paquetes en tres partes: *packet filtering* (filtrado de paquetes), *connection tracking* (seguimiento de conexiones) y NAT (traducción de direcciones de red). Cada una de estas partes se conecta a los módulos de netfilter en diferentes puntos para acceder a los paquetes. Los subsistemas de seguimiento de conexiones y NAT son más generales y poderosos que los que realizaban *ipchains* e *ipfwadm*.

Esta división permite a iptables, a su vez, usar la información que la capa de seguimiento de conexiones ha determinado acerca del paquete: esta información estaba antes asociada a NAT. Esto hace de iptables un sistema superior a *ipchains*, pues tiene la capacidad de monitorear el estado de una conexión y redireccionar, modificar o detener los paquetes de datos en base al estado de la conexión y no únicamente en base al origen, destino o contenido del paquete.

En este punto se distinguen dos términos que se originan de la capacidad de iptables de mantener información respecto del estado de la conexión y estos son *firewall stateful* que es un firewall estructurado mediante la utilización de iptables para monitorear el estado de la conexión y *firewall stateless* que es una estructura de firewall que no maneja información sobre el estado de las conexiones. Esta situación pone en ventaja a iptables, pues permite una toma de decisiones más acertadas por el hecho de que iptables está al tanto del contexto completo del que surge un paquete.

1.13.3 OPERACIÓN

La estructura de netfilter/iptables permite al administrador del sistema definir reglas respecto de qué hacer con los paquetes de red. Estas reglas se agrupan en cadenas y las cadenas se agrupan en *tablas*: cada tabla está asociada con un tipo diferente de procesamiento de paquetes.

Cada regla especifica qué tipo de paquetes son los que cumplen esta regla y un *destino* que indica lo que se hará con el paquete, si es que éste cumple la regla. Cualquier paquete de red que llega a una computadora recorre al menos una cadena y cada regla de esa cadena intenta corresponderse con el paquete. Si la regla se cumple, el recorrido a través de la cadena se detiene y el destino de la

regla indica lo que se debe hacer con el paquete. Si el paquete llega al fin de una cadena determinada sin corresponderse con ninguna regla de la cadena, la *política de destino* de la cadena dicta qué hacer con el paquete.

Mediante *iptables*, las reglas se agrupan en cadenas. Una cadena es un conjunto de reglas dirigidas a paquetes IP, que indican lo que se hará con ellos. Una regla puede desechar el paquete de la cadena, con lo que otras cadenas no se considerarán. Existen tres cadenas básicas: INPUT, OUTPUT y FORWARD o ENTRADA; SALIDA y REENVÍO.

En el anexo se puede observar la descripción de los componentes de las reglas de tráfico, los parámetros de comando y la forma de implementar cada una de las reglas a través de `iptables`.

1.14 DETECCIÓN Y PREVENCIÓN

Dentro de un sistema que tiene la posibilidad de ofrecer varios servicios como una estación y dentro de una plataforma en la capacidad de ser atacada a través de las aplicaciones que genera y como en cualquier otro sistema de firewall, se deben establecer algunas medidas que permitan la detección y/o prevención de un ataque que pueda comprometer nuestros sistemas y provocar pérdida de información o daño de los equipos.

1.14.1 SISTEMA DE ARCHIVOS

Existen básicamente tres formas de detectar ataques en nuestro sistema:

- Cambios en el sistema de archivos
- Paquetes extraños en la red
- Entradas extrañas en el archivo de *log* (bitácora)

1.14.1.1 Punto de Partida

El punto de partida de un sistema es una copia del mismo cuando se está seguro de que no ha sufrido alteraciones. Esta copia es usualmente un archivo o número

de archivos que son almacenados en un medio seguro, como un disco compacto, una cinta de sólo lectura, o un *disquette* de las mismas características.
Este punto de partida puede utilizarse para identificar los cambios en nuestro sistema. Existen varias razones para querer conocer esto:

- Podríamos querer conocer qué archivos han sido cambiados, añadidos o borrados luego de una tarea de administración del sistema. No solamente para entender qué es lo que se ha hecho sino también para actualizar el punto de partida con esta información.
- Podríamos desear conocer qué archivos borró, cambió o añadió un hacker o los permisos que alteró durante un ataque.
- Los sistemas computacionales tienden a cambiar en el tiempo y los archivos de log incrementan su tamaño y se sobrescriben, los usuarios crean y borran archivos, el correo se acumula, etc. Es muy interesante conocer qué cosas están yendo normalmente y cuáles no.

Existen varias formas para crear un punto de partida:

- La primera forma es realizar un respaldo completo del sistema que incluya todos sus archivos. Esto puede utilizarse para recuperar archivos que se borran, o para restablecer el sistema completo en caso de cambios masivos realizados por un hacker. Sin embargo, no es muy práctico para determinar cambios pequeños.
- Se puede crear un punto de partida que incluya los aspectos que para nosotros sean esenciales.
- Podemos usar varias herramientas que están disponibles para el monitoreo del sistema de archivos. Estas herramientas crean un punto de partida del sistema de archivos solamente e incluyen herramientas para revisar de forma automática la situación actual en comparación con el punto de partida.

Un punto de partida puede ser generado por nosotros mismos y debe consistir de todos los archivos importantes, configuraciones e información de estado de

nuestro sistema. Usando este punto de partida deberemos ser capaces de determinar rápidamente los cambios que ha hecho un hacker.

Algunos de los aspectos que se deben tomar en cuenta son:

- Se necesita todos los lineamientos de configuración, la mayoría de los cuales se encuentran dentro de directorios específicos como /etc y /boot.
- Es necesario capturar la salida de varios comandos que detallan el estado de nuestro sistema.
- Se necesita alguna información respecto de los ejecutables que existen en nuestro sistema, de modo que seamos capaces de detectar si han cambiado. La aplicación **md5sum** hace esto utilizando el algoritmo MD5 para crear un *checksum* criptográfico de un archivo.

Todos estos archivos y la salida de los comandos podrían, por ejemplo, ser almacenados en un archivo comprimido dentro de un disco floppy.

1.14.1.2 Chequeo de integridad del sistema de Archivos

Existen herramientas de chequeo de integridad del sistema de archivos que lo sondean y almacenan todas las características importantes de cada archivo: usuario, grupo, permisos, tamaño, etc. La mayoría de herramientas agregan uno o más *checksums* criptográficos a la lista también. La lista de características de cada archivo es almacenada en una base de datos (a veces encriptada) y el sistema de archivos actual es chequeado regularmente con esta base de datos.

Algunas de las herramientas que existen son:

- *Tripwire*.- Es quizá la herramienta más antigua y clásica, desarrollada en el mundo académico y que posteriormente se hizo comercial. Hay sin embargo, una versión libre para Linux.
- *AIDE*.- Es una versión libre que se creó para sustituir a Tripwire aun cuando no llega a superar a la versión comercial de este último.
- *L5.*- Es básicamente lo mismo que AIDE.
- Existen otro tipo de herramientas y una larga lista de ellas puede hallarse en http://www.securityportal.com.

1.14.2 SISTEMAS DE DETECCIÓN DE INTRUSIÓN EN LA RED

La Detección de Intrusos de Red significa básicamente monitorear todos los paquetes en la red como un *sniffer*, y detectar los intentos de ataque. La mayoría de monitores de red solamente detectan escaneo de puerto pues son los más sencillos de detectar (una dirección IP que se conecta a un gran número de puertos en un corto tiempo es un signo seguro de escaneo de puertos).

La mayoría de monitores de paquetes de red trabajan de forma autónoma y nos advierten de intentos de ataques, vía correo electrónico, por ejemplo.

Entre algunas de las herramientas se pueden mencionar: *Psionic Portsentry, Scanlogd* y el conocido *Snort.*

1.14.3 ENFRENTANDO LOS ATAQUES

Una vez que se ha determinado que estamos siendo atacados, las acciones que deben tomarse dependerán de muchos factores. Es necesario seguir algunos pasos para recopilar información respecto del ataque:

- Primeramente, es necesario obtener información, mediante el seguimiento de los eventos que ocurren en la red mediante herramientas propias del sistema operativo Linux como *tcpdump* o *ethereal*, preferiblemente desde un host que no esté siendo atacado, de modo que el hacker no sospeche que está siendo rastreado. Esta operación es útil para determinar lo que está sucediendo durante el ataque pero puede jugar también un papel importante para posibles acciones legales.
- Luego, antes de iniciar cualquier labor de investigación en el firewall mismo, deberemos asegurarnos que cada acción que tomemos mediante la ejecución de comandos sea registrada junto con su respectiva salida, de manera que podamos analizar posteriormente lo que está ocurriendo y evaluar nuestras propias acciones.
- El siguiente paso es determinar la fuente del ataque. Intentos simples de ataque o ataques DoS pueden utilizar direcciones IP falsificadas pero los ataques más ingeniosos no suelen utilizar este tipo de mecanismos. De esta forma podemos identificar la fuente del ataque al que nos enfrentamos. Mediante una consulta DNS reversa en esta dirección IP y

podremos tener una idea de cuál es la fuente del ataque. Quizás no será muy conveniente contactar con el administrador del sistema vía correo electrónico pues la red está siendo atacada. Será mejor contactarlo por teléfono.

- Será necesario también determinar el objetivo del ataque, es decir identificar si se trata de un simple escaneo de puertos general o de un ataque dedicado hacia un servicio en particular. Esto se logra detectar fácilmente mediante un sondeo de la red.

En este punto deberíamos tener una buena idea acerca de la clase de ataque que estamos enfrentando y si es necesario tomar acciones para detener el ataque, mediante algunas de las formas siguientes:

- Bloquear la dirección IP origen del ataque, lo que es sencillo si el ataque proviene de un número limitado de hosts, pero será casi imposible si sufrimos un ataque DDoS. Este bloqueo se podrá realizar fácilmente a través de iptables con una regla específica.
- La otra alternativa es deshabilitar el servicio que es objetivo del ataque. Esto no puede hacerse tan a la ligera, y hay que tener mucho cuidado con los servicios que dejan de funcionar.
- Cuando todo esté tranquilo nuevamente, es necesario primero chequear cualquier daño del sistema y si algún daño existe, asegurarnos de repararlo (o al menos restaurar el respaldo del sistema más reciente).
- Será necesario determinar la razón por la que el ataque fue exitoso, y corregir la vulnerabilidad, desconectado, incluso, el servicio hasta que se pueda encontrar un parche. Solamente cuando se haya corregido la falla podremos restaurar el servicio de forma normal.
- El último paso es analizar el ataque y nuestra respuesta a él. Debemos guardar todos los archivos que sean relevantes, incluyendo la salida de los comandos que permiten el monitoreo de la red para iniciar una cadena de evidencia, si es necesario.

CAPÍTULO 2

DISEÑO Y CONFIGURACIÓN DEL PROTOTIPO VOZ Y DATOS SOBRE LINUX

CENTRAL IP PARA LA CORPORACIÓN MACHÁNGARASOFT

2.1 REQUERIMIENTOS DE TELEFONÍA

2.1.1 REQUERIMIENTOS DE CONEXIÓN

La corporación MachángaraSoft, como ya se mencionó, está conformada por 9 empresas que comparten el mismo espacio físico. Cada empresa tiene en promedio 3 personas y esta cifra va creciendo conforme las empresas se van desarrollando. La corporación dispone de 2 líneas telefónicas que son contestadas por una asistente común. Una vez que ingresa una llamada telefónica desde la PSTN, en cualquiera de los teléfonos, la asistente se encarga de contestar y redirigir el aparato telefónico hacia el miembro de la empresa que corresponda.

Esto es claramente inadecuado pues se debería aprovechar la infraestructura de red existente para que estas llamadas sean enrutadas *automáticamente* de manera controlada, dependiendo de los requerimientos de quien llama.

Asimismo, la imagen corporativa de las empresas mejoraría al disponer de un sistema interactivo de voz, reflejando la organización y la utilización de soluciones tecnológicas *opensource* como *Asterisk,* que pueden acelerar el desarrollo de las empresas gracias a su costo sumamente reducido.

Inicialmente será necesario asignar únicamente una extensión por empresa y una designada a la asistente, que será la operadora, en caso de que se generen necesidades de comunicación específicas que no puedan satisfacerse a través del sistema automático de voz. Del mismo modo los clientes internos necesitan comunicación con la PSTN y el servicio deberá permitir el enrutamiento desde la LAN hacia la infraestructura de Andinatel y a través de ella a llamadas regionales, internacionales y a dispositivos de telefonía celular.

El tráfico telefónico no es necesariamente masivo y por tanto, el servidor deberá estar en capacidad de atender, en el peor de los casos, 3 llamadas hacia o desde la PSTN , incluyendo 4 llamadas IP simultáneas tomando en cuenta la población del parque, esto sin ningún problema de perfomance del procesador ya que se determinó que el *hardware* ideal incluiría un procesador de 3.0 Ghz, lo cual es más que suficiente, tomando en cuenta ciertas experiencias en foros de discusión calificados[6] y las pruebas que se ejecutarán con el prototipo.

2.1.2 SERVICIOS ADICIONALES

De manera general, existe una gran cantidad de aplicaciones y servicios que se pueden brindar a través de Asterisk y telefonía IP. Los servicios adicionales (no la estricta conexión de voz) que se requiere del sistema son: *IVR (direccionamiento de extensiones, información de la corporación, servicio nocturno)*, *Correo de Voz, Parqueo de Llamadas, Fax al correo electrónico, Directorio, Transferencia de llamada, Llamada en espera y Conferencia.*

2.1.3 REQUERIMIENTOS DE CONTROL

Gracias a la implementación del sistema de Central IP en base a Asterisk se abren nuevas y poderosas posibilidades de control del tráfico de voz tanto saliente, interno y entrante. Esto es especialmente útil para restringir llamadas de alto costo como llamadas internacionales y a dispositivos celulares. Esto es estrictamente un requerimiento pues los costos de telefonía se han incrementado excesivamente los últimos meses.

En este sentido es también necesario que se establezcan límites en el tiempo de llamada de manera que las líneas no sean utilizadas excesivamente para asuntos personales. Un límite pertinente sería de 5 minutos por llamada saliente y entrante. Adicionalmente, aunque sería redundante, se podría agregar un mecanismo de *log* o monitoreo de llamadas y su consumo.

[6] http://www.voip-info.org/wiki-Asterisk+dimensioning

2.2 DIMENSIONAMIENTO DEL SERVICIO DE CENTRAL IP

2.2.1 CARACTERÍSTICAS DE HARDWARE

2.2.1.1 *Servidor*

Debido a que los requerimientos iniciales son bastante reducidos, por el grupo relativamente pequeño de extensiones que se crearán, se podría utilizar un equipo modesto, sin embargo, como ya se indicó, se deberá tomar en cuenta el crecimiento en el número de empresas que se reflejaría en el aumento de clientes de telefonía.

Por lo tanto, las características del servidor serían las siguientes:

- Procesador Intel (opcionalmente AMD) Pentium 4 de 3.0 Ghz
- Memoria RAM de 1 GB expandible a 4 GB
- Disco Duro de 120 GB
- Tarjeta de Red 10/100 Mbps

Debido a que el sistema partirá desde cero, es posible determinar una serie de estándares como la utilización de códecs específicos y sin costo como SIP e IAX de modo que no sea necesario el proceso de transcodificación en el servidor, disminuyendo notablemente la carga sobre el mismo. Esto hará del sistema mucho más escalable y su instalación representará menores costos.

Como el servidor de telefonía IP debe estar conectado a la PSTN de Andinatel, se necesitará una tarjeta que incluya 3 módulos FXO que se inserte en la ranura PCI de la estación y que permita interactuar al servidor con la infraestructura telefónica analógica. Cada línea de teléfono se conectará a un módulo FXO y la red de la corporación se conectará a través de la tarjeta de red correspondiente hacia un *switch*. Para la demostración del prototipo se utilizará una tarjeta clónica de un solo puerto FXO, pero para el sistema en producción se requerirá de al menos una tarjeta *Digium* modelo *TDM400P* con los tres módulos mencionados y con capacidad de colocar hasta 4 en total (para futuras líneas analógicas). En el anexo C se especificarán algunas características técnicas de estas interfaces.

Haciendo un breve análisis del tráfico telefónico en un día a mitad de semana, y en la hora pico (de 11h00 a 12h00), se obtuvo los siguientes datos:

TABLA 2.1 DATOS DE FLUJO TELEFÓNICO MIÉRCOLES 9 DE MAYO DE 2007.

Nº	RECIBIDA/HECHA	INICIO	FIN
1	HECHA	10:57:00	10:59:00
2	HECHA	11:01:00	11:10:00
3	HECHA	11:12:00	11:13:00
4	HECHA	11:20:00	11:22:00
5	HECHA	11:27:00	11:30:00
6	HECHA	11:32:00	11:33:00
7	RECIBIDA	11:43:00	11:45:00
8	RECIBIDA	12:01:00	12:05:00

El tiempo medio de llamada es de 3 minutos. Se observa que el número de llamadas es de 8 en un período de 60 minutos (la hora pico).
El cálculo de la intensidad de tráfico sería:

$$A = \frac{1}{T}(n * tm) = \frac{1}{3600[s]}(8 * 180[s]) = 0.4[erlangs]$$

La fórmula de *Erlang* y las tablas que de ella se generan permiten calcular el número de troncales (circuitos u órganos) que brindarán servicio a una determinada intensidad de tráfico con un mínimo de pérdida establecido.
Si se asume como aceptable un Grado de Servicio de 0.01 (1 de cada 100 llamadas se pierde), tomando en cuenta el tráfico calculado, **se requerirán al menos 3 circuitos** o líneas analógicas a la PSTN.
Tomando en cuenta el crecimiento que se proyecta, lo más recomendable sería adquirir una tarjeta TDM2400P con 3 módulos FXO, de manera que, a medida que la intensidad de tráfico se eleve, se vayan agregando módulos y troncales analógicas.

2.2.1.2 Teléfonos IP

La solución ideal para acceder al servidor de telefonía IP es la adquisición de dispositivos dedicados que puedan ser configurados con todos los parámetros de direccionamiento necesarios para la comunicación dentro de la LAN, a través de Internet y mediante la PSTN, es decir, un teléfono IP. En el anexo C se puede observar las especificaciones y capacidades de algunos modelos. Debido a que se está adaptando un sistema completamente nuevo, la compra de los equipos se orientará específicamente a aquellos que manejen protocolos de comunicaciones y códecs no propietarios, como SIP e iLBC respectivamente.

2.2.1.3 Adaptadores ATA

Una solución para la conexión de los usuarios al servidor consiste también en la reutilización de sus teléfonos analógicos "normales" mediante un Adaptador Telefónico Analógico o *gateway*, que permite convertir las señales analógicas generadas por un teléfono convencional en paquetes IP que puedan transportarse a través de la red *Ethernet.* En este caso, la configuración de los parámetros de direccionamiento se realizará en estos dispositivos mediante una conexión vía web. En el mercado actual no es significativo el ahorro que se esperaría al utilizar adaptadores respecto de teléfonos IP, por lo que es claro que conviene adquirir un teléfono IP frente a un adaptador, aunque algunos de estos adaptadores podrían permitir la conexión de 2 teléfonos analógicos en cuyo caso el costo se reduciría mucho.

En el caso de que la decisión se inclinase por obtener un adaptador, una buena opción será el dispositivo de marca **Linksys** modelo PAP2-NA, por su costo y especialmente la variedad de especificaciones (ver anexo C) que posee.

2.2.2 CARACTERISTICAS DE SOFTWARE

2.2.2.1 Sistema Operativo

El paso inicial luego de tener listo todo el hardware necesario para el servidor es la instalación de la plataforma operativa. *Asterisk* funciona únicamente bajo plataformas Linux y la distribución mejor soportada es sin duda *CentOS,* pues la mayoría de distribuciones específicas para el funcionamiento de Asterisk y sus

herramientas, como *Trixbox*[7] *o* Asterisk@Home[8] y dispositivos (*appliances*) dedicados para telefonía IP tienen como base a este clon binario de *RedHat*.

2.2.2.2 Softphones

Otra solución bastante particular que puede aprovecharse en empresas en las que cada miembro dispone de una estación de trabajo se relaciona con aplicaciones de *software* para vídeo y voz muy similares a otras como *Skype* o *MSN Messenger*. Estas aplicaciones se denominan *softphones* o teléfonos de software y se pueden configurar muy fácilmente con la dirección IP de la máquina *host*. Así, los clientes de telefonía podrán comunicarse utilizando los dispositivos de entrada y salida de la tarjeta de sonido del equipo, aunque la calidad del sonido se ve muy degradada.

Se debe tomar en cuenta el sistema operativo de los clientes pues como es de esperarse, varias de las aplicaciones para *Windows* son propietarias y tienen un costo. Para Linux en cambio hay una gran variedad de aplicaciones que pueden descargarse de Internet. Las opciones incluyen soporte tanto para protocolos SIP e IAX como para códecs igualmente sin costo como GSM, Speex e iLBC.

Entre los *softphones* que se tomarán en cuenta en el sistema están:

TABLA 2.2 SOFTPHONES (PROTOCOLO, Y SISTEMA OPERATIVO)

Software	Sitio	Protocolo	S. O.
Xlite	http://www.xten.com	SIP	Linunx
Kiax	http://kiax.sf.net	IAX2	Linux
Netmeeting	http://microsoft.com	H.323	Windows
Gnomemeeting	http://gnomemeeting.org	SIP/H.323	Linux

2.2.2.2.1 Para Windows

- **Idefix**[9].- Es una aplicación de cliente de telefonía IP que utiliza el protocolo IAX de Asterisk. Posee una versión sin costo y una avanzada con características adicionales.

[7] http://www.trixbox.org
[8] http://www.asterisknow.org
[9] http://www.asteriskguru.com/idefisk/

- **KIAX**[10].- Es otra aplicación de telefonía que utiliza protocolo IAX de Asterisk para comunicarse con el servidor. La versión de Windows es muy sencilla.
- **XLite**[11].- Es un teléfono propietario de *software* para VoIP desarrollado por *Counterpath*. Existen versiones de evaluación, pero la última versión comercial posee soporte adicional para vídeo y mensajería instantánea a través del protocolo SIP.

Existen en realidad una gran variedad de aplicaciones que podrían utilizarse, lamentablemente la mayoría de las que están disponibles para sistemas *Windows* tienen un costo. Ventajosamente, con las herramientas de las versiones de evaluación se pueden satisfacer la mayoría de necesidades de conexión de telefonía IP.

2.2.2.2.2 Para Linux

La variedad de aplicaciones cliente para telefonía IP sobre plataformas Linux es muy extensa. Incluso hay versiones libres de aplicaciones propietarias y que soportan gran cantidad de códecs, asimismo de distribución y uso libre. Entre las que se utilizarán están:

- **Ekiga Softphone**[12].- La versión anterior de este se denominaba *GnomeMeeting*, y es una aplicación *opensource* de VoIP para telefonía IP y vídeo conferencia que funciona tanto para Linux como Windows. Además tiene soporte para protocolos SIP y H.323 (su versión liberada), así como de varios códecs de audio y vídeo de alta calidad.
- Existe una versión de **KIAX** para Linux que es mucho más funcional que su correspondiente para Windows así como otros varios clientes SIP[13].

[10] http://sourceforge.net/projects/kiax

[11] http://www.counterpath.com/

[12] http://www.ekiga.org/

[13] http://www.linphone.org/,http://www.wirlab.net/kphone/

La implementación del sistema deberá incluir la configuración de 10 clientes SIP (o IAX de preferencia) entre teléfonos IP y *softphones* correspondientes a cada una de las extensiones dentro de la corporación. No tiene sentido que se instalen dispositivos que funcionen con protocolos y códecs propietarios pues involucran un costo mayor.

2.2.2.3 Paquetes de Instalación

Para instalar y configurar la central PBX para VoIP se requiere de la instalación de varios paquetes, entre los que están (se detalla el nombre completo con la versión):

Paquetes propios del Sistema Operativo

gcc	**bison**
libxml2-devel	**ncurses-devel**
libtiff-devel	**audiofile-devel**
mysql-server	**subversion**
php-gd	**libogg-devel**
php-mysql	**zlib-devel**
kernel-devel	**lame**

Todos estos paquetes, a excepción de **lame,** se encuentran en la distribución de *CentOS,* cuya instalación se recomendó. Este último puede instalarse fácilmente a través de Internet mediante la herramienta **yum.**

Paquetes específicos para la instalación de la Central IP:

zaptel-1.2.11.tar.gz Instala los *drivers* necesarios para que Asterisk pueda interactuar con el hardware analógico y digital de telefonía

libpri-1.2.4.tar.gz Es una librería opcional, para el funcionamiento de interfaces ISDN PRI, pero es recomendable instalarla.

asterisk-1.2.13.tar.gz Es el paquete que brindará todas las funcionalidades de IP PBX del sistema.

`asterisk-sounds-1.2.1.tar.gz` Instala los sonidos por defecto de la PBX, en este caso, en idioma inglés.

2.2.2.4 Estructura de directorios del sistema IP PBX de Asterisk

Asterisk usa varios archivos en un sistema Linux para manejar muchos de los aspectos del sistema, tales como grabaciones de correo de voz, plan de marcación, registros de eventos, módulos de hardware y binarios del sistema.

En la tabla 2.3 se puede apreciar algunos de los directorios necesarios que se crean durante el proceso de instalación.

TABLA 2.3 ESTRUCTURA DE DIRECTORIOS DE ASTERISK

Directorio/Contenido	Descripción
`/etc/asterisk/`	Esta carpeta contiene básicamente todos los archivos de configuración que manejan el funcionamiento de la IP PBX.
`/usr/lib/asterisk/modules/`	Este directorio contiene todos los módulos que maneja Asterisk para sus aplicaciones, códecs, formatos y canales.
`/var/lib/asterisk/`	Contiene archivos con información de la base de datos Asterisk y entre otros directorios, el correspondiente a los sonidos del sistema.
`/var/spool/asterisk/`	Contiene directorios con información de las comunicaciones realizadas a través del servidor, incluyendo los correos de voz de cada extensión.
`/var/run/`	Contiene información sobre el ID de proceso de todos los procesos del sistema.
`/var/log/asterisk/`	Almacena la información de *log* o registro de eventos del servidor Asterisk. El tipo de información que se registra puede controlar.
`/var/log/asterisk/cdr-csv`	En este directorio se almacenan registros de detalle de todas las llamadas que se realizan a través del servidor.

2.3 CONFIGURACIÓN DE LA CENTRAL IP

En esta sección se detalla el proceso de configuración de la central Asterisk, orientándose básicamente a las interfaces analógicas, las extensiones de los usuarios y el plan de marcación del sistema. Este último es sin duda el corazón de la central IP pues maneja el comportamiento del tráfico telefónico en base a los

requerimientos de los usuarios, de las personas que llaman a la organización y de la corporación misma, especialmente en lo referente al control de las llamadas y al desarrollo de una experiencia de voz más confortable y organizada para los usuarios. La lógica del plan de marcación se detalla en las páginas siguientes.

Fundamentalmente, los archivos de configuración que se manejarán y editarán para modificar y controlar las funciones de la central de telefonía IP se detallan en la tabla 2.4.

TABLA 2.4 ARCHIVOS DE CONFIGURACIÓN DE INTERFACES.

Archivo de Configuración	Descripción
`zaptel.conf`	Configuración de bajo nivel de las interfaces analógicas de *hardware*.
`zapata.conf`	Configuración de la interfaz entre Asterisk con el *hardware*.
`extensions.conf`	Configuración del Plan de Marcación
`sip.conf`	Extensiones con protocolo SIP.
`iax.conf`	Extensiones con protocolo IAX2.

2.3.1 CONFIGURACIÓN DEL SISTEMA OPERATIVO

La configuración del sistema operativo Linux base para el funcionamiento de la aplicación *Asterisk* involucra simplemente la instalación de los paquetes mencionados anteriormente que contienen algunas de las librerías esenciales para el desarrollo de algunas de las funcionalidades del sistema de IP PBX.

El detalle en cuanto al proceso de verificación e instalación de todos estos paquetes puede encontrarse en el anexo B. Un detalle especial en cuanto a un error existente en 2 archivos de compilación debe tomarse en cuenta y se aclara también en el mencionado anexo.

Aparte de estas consideraciones no hay nada más que tomar en cuenta con relación a la preparación del sistema operativo para la instalación del aplicativo Asterisk. Sin embargo, puede tomarse en cuenta algunos puntos relacionados con la seguridad y accesibilidad del servidor como en el caso de cualquier otro servidor crítico para las comunicaciones de una empresa.

Un aspecto importante se relaciona con la instalación de los sonidos correspondientes a determinado idioma para los mensajes que se reproducirán en la central IP PBX. Inicialmente se encuentran disponibles los sonidos por defecto

en inglés pero existen proyectos que han generado completas estructuras de sonidos en español con diferentes acentos[14]. Lo más importante al instalar los sonidos es mantener la estructura para los idiomas dentro del archivo donde se almacenan.

2.3.2 CONFIGURACION DE INTERFACES

En el sistema que se propone se establece la necesidad de configuración de interfaces tanto analógicas (2) para la interacción con la red de telefonía pública, e interfaces IP en base a protocolos de VoIP como SIP e IAX. Ventajosamente, Asterisk brinda soporte para estas interfaces y protocolos.

2.3.2.1 Interfaces Zaptel

Se conectará cada línea telefónica para su interacción con el servidor *Asterisk* a través de una tarjeta PCI TDM400P de *Digium* que posee 2 módulos FXO para ese efecto. En el caso del prototipo se utilizará una tarjeta AX100P clónica con un solo módulo FXO para la demostración. Las características técnicas y de funcionamiento se describen en el anexo C.

Para determinar que la tarjeta se encuentra presente para el SO, se puede ejecutar el comando **lspci** obteniendo, en el caso de utilizar la tarjeta X100P, la siguiente salida:

```
# lspci | grep X100P
00:08.0 Communication controller: Motorola Wildcard X100P
```

En definitiva, el proceso de configuración de los módulos, luego de que el *hardware* haya sido reconocido por el sistema operativo, es bastante sencillo y similar, ya sean FXO o FXS.

Esta configuración se la realiza en el archivo **`/etc/zaptel.conf`** y cada vez que éste sea editado será necesario recargarlo para que los cambios tengan efecto en el manejo de los dispositivos, mediante el comando **`/sbin/ztcfg –vv.`**

La configuración básica que permite definir un puerto FXO con señalización FXS y protocolo de señalización *kewlstart* sería el siguiente:

[14] Un *stand* muy popular de sonidos para Asterisk en español puede descargarse de http://www.ip-flow.com.ar/elianna_pack.html.

```
fxsks=1
defaultzone=us
loadzone=ec
```

Estos cambios serán suficientes para proceder a cargar los *drivers* para la tarjeta de modo que puedan ser usados por el kernel de Linux. Entre los módulos que deberían cargarse están:

wctdm
wcfxo
wctdm24xxp
wcte11xp
wct1xxp
wct4xxp
zaptel

La carga de estos módulos puede verificarse con el comando **lsmod.**

Para probar que el *hardware* y los puertos han sido instalados y configurados correctamente, se usa la aplicación *ztcfg* así:

```
# /sbin/ztcfg –vvv
```

En el caso del prototipo con la tarjeta X100P y un solo módulo FXO, la salida esperada sería la siguiente:

```
Zaptel Configuration
======================
Channel map:
Channel 01: FXS Kewlstart (Default) (Slaves: 01)
1 channels configured.
```

Una vez que los módulos se cargaron adecuadamente se debe constatar que estén en capacidad de interactuar con (este caso) la red de telefonía analógica. Para ello se conecta la línea telefónica al módulo que se ha configurado y se ejecuta el comando *zttool.*

Esta herramienta mostrará las alarmas del dispositivo que se ha configurado. Si esta en **OK**, significa que ese módulo está listo para transmitir y recibir información.

Ahora es necesario configurar a Asterisk de modo que pueda interactuar con el *hardware* instalado. Esto se logra editando el archivo **/etc/asterisk/zapata.conf**. Este archivo permite configurar varias características y funcionalidades asociadas

con los canales de *hardware,* como Identificador de llamadas, llamada en espera, cancelación de eco y otras opciones más, tal y como se aprecia en estos parámetros de configuración:

```
[channels]
language=es       ;lenguaje a utilizarse en los sonidos
usecallerid=yes   ;Activar identificador de llamadas
hidecallerid=no
callwaiting=no    ;Deshabilita tono de llamada en espera
threewaycalling=yes ;Habilita conferencia de llamada
transfer=yes      ;Habilita transferencia de llamada
echocancel=yes    ;Cancelacion de eco
txgain=4.0        ;Ganancia en tx
rxgain=4.0        ;Ganancia en rx
busydetect=yes    ;Habilita detección de ocupado
context=entrante-pstn    ; Las llamadas entrantes por el canal 1 van al
contexto [entrante-pstn] en el archivo extensions.conf.
signalling=fxs_ks       ; Se usa señalizacion FXS en un canal FXO
channel => 1            ; El canal 1 esta conectado a la PSTN
```

Cabe resaltar que todas las opciones que se indican corresponden al canal definido inmediatamente después de ellas, de modo que si se requiere agregar alguna, se debe hacerlo antes de la definición del canal al que corresponde.

2.3.2.2 Interfaces SIP

La configuración de extensiones correspondientes a dispositivos que funcionarán con este protocolo se realiza en el archivo **`/etc/asterisk/sip.conf`**. Su estructura es bastante sencilla y engloba las opciones de registro, autenticación y configuración básica de cada dispositivo SIP que tendrá acceso a comunicación a través del servidor Asterisk.

Este archivo contiene la definición de un contexto general dentro del que se establecen algunas opciones que no varían para cada uno de los dispositivos o usuarios que se conectarán al servidor. En el caso de que la opción general no corresponda a algún dispositivo específico, como por ejemplo un códec, se podrá

especificar dentro de las opciones propias de ese dispositivo y tendrá preponderancia sobre la configuración general.

Muchas de las opciones que se definen en el contexto general son las mismas que se definen en cada contexto específico. Entre ellas tenemos, el puerto UDP al que se asociarán los clientes (típicamente el 5060), si enviarán peticiones de resolución DNS, el contexto dentro del archivo **`/etc/asterisk/extensions.conf`** en el que se manejará su comunicación, los códecs que se aceptarán, el soporte de vídeo y algunas opciones para trabajar con NAT.

Un ejemplo dentro del archivo en relación al dispositivo SIP de la operadora sería el siguiente:

```
[operadora]
type=friend             ;puede hacer y recibir llamadas
username=operadora      ;nombre de usuario para el registro
secret=operadora        ;password del usuario
host=dynamic ;obliga a que el usuario tenga que registrarse con el server
qualify=yes              ;permite mantener abiertas las sesiones NAT
nat=yes
canreinvite=no
context=operadora       ;contexto al que pertenece en el dialplan
disallow=all
allow=gsm               ;permitir codec GSM
allow=speex             ;permitir codec Speex
```

El archivo de configuración (`sip.conf`) completo se puede encontrar en el anexo A.

2.3.2.3 Interfaces IAX

La configuración de dispositivos IAX se incluye en el archivo **`/etc/asterisk/iax.conf`**. La utilización de dispositivos que manejen el protocolo IAX2 de *Asterisk* es una seria alternativa a la anterior debido a que éste último protocolo es mucho más fácil de manejar cuando existe un proceso de NAT que tienen que atravesar las comunicaciones de los clientes. La información y la señalización viajan a través del mismo puerto y por ello, las consideraciones a nivel de *firewall-NAT* son menos complicadas.

La estructura de configuración es muy parecida a la correspondiente a SIP.

Aún cuando no se utilizará este protocolo para la configuración del prototipo, es una firme opción en caso de que el flujo de comunicación SIP se convierta en un problema para el manejo del cortafuegos, así como para la implementación de troncales entre 2 servidores *Asterisk*.

2.3.3 CONFIGURACIÓN DEL PLAN DE MARCACIÓN

El plan de marcación es el núcleo del sistema de telefonía IP pues maneja el procesamiento de todas las llamadas internas, hacia la PSTN, hacia Internet y desde luego todas las llamadas entrantes a través del servidor *Asterisk.*

Esto implica que el *dialplan* prevea en su configuración todos los posibles eventos que un usuario de telefonía puede generar, dándole especial importancia al control de acceso a determinadas redes cuyo uso podría generar un gasto excesivo a la corporación. Todo esto se logra especialmente mediante el uso de algunas de las muchas aplicaciones embebidas en el sistema de IP PBX de Asterisk que hacen que el proceso de configuración pueda adaptarse a las necesidades específicas del parque y dejar un margen amplio de flexibilidad del que no goza ningún otro producto propietario.

La configuración de la estructura del plan de marcación y la ejecución de sus herramientas se realiza editando el archivo **`/etc/asterisk/extensions.conf.`** El archivo completo generado para las necesidades de la corporación puede encontrarse en el anexo A.

2.3.3.1 Conceptos

La construcción del plan de marcación para *Asterisk* involucra varios conceptos que gobiernan la lógica del sistema mismo de PBX.

Entre esos conceptos se puede mencionar:

2.3.3.1.1 Contextos

Los contextos son grupos de instrucciones (extensiones) que llevan un nombre entre corchetes (**`[contexto]).`** Cada uno de estos grupos está aislado el uno del otro a menos que explícitamente se los relacione entre sí mediante alguna herramienta. Este mecanismo permitirá organizar jerárquicamente los privilegios de determinadas entidades de la corporación, controlando el acceso a conexiones

y aplicaciones. Es necesario, sin embargo, tener mucho cuidado con el manejo de estos contextos pues así como permiten el control de acceso a determinadas características, una mala configuración podría provocar que se permita el uso fraudulento del sistema y que por ejemplo, usuarios externos a la corporación puedan realizar llamadas hacia la PSTN o celular a través del servidor de telefonía.

2.3.3.1.2 Extensiones

Las extensiones son instrucciones dentro de cada contexto, instrucciones que se ejecutarán activadas por una llamada entrante o por un patrón de dígitos marcado por el usuario y que ejecutarán una aplicación relacionada con el procesamiento de esa llamada.

Estas extensiones en Asterisk están formadas por tres partes como se puede ver en la tabla 2.5.

TABLA 2.5 PARTES DE UNA EXTENSIÓN EN EL DIALPLAN

Parte	Descripción
Nombre o número	Identifica a la extensión
Prioridad	Determina la secuencia de ejecución de las instrucciones
Aplicaciones	El comando que ejecuta alguna acción sobre la llamada

Un ejemplo de la extensión típica que permite contestar una llamada sería el siguiente:

```
exten => 301,1,Answer()
exten => nombre,prioridad,aplicacion()
```

Una extensión especial llamada *start* o **s** es de especial utilidad cuando se reciben comunicaciones de dispositivos de los que no se puede establecer su extensión de origen. Esta extensión es empleada para el manejo de las llamadas entrantes que vienen desde la PSTN.

2.3.3.1.3 Prioridades

Las prioridades determinan la secuencia de ejecución de cada una de las extensiones o instrucciones al interior de un contexto. Es como un identificador que indica cada una de las etapas que atraviesa una llamada cuando alcanza una

extensión. Dentro de esa secuencia estaría por ejemplo: contestar una llamada, reproducir algunos mensajes de indicación, esperar por el marcado de un conjunto de teclas, etc.

Aún cuando hay algunas excepciones, la numeración debe empezar en 1 y paulatinamente ir creciendo. Sin embargo, en ocasiones en las que las etapas de una extensión llegan a crecer mucho, la inclusión de una nueva instrucción implicaría el cambio de varios de los valores de prioridad. En estos casos es útil la utilización de la prioridad **n** que representa el valor de prioridad anterior sumado uno, haciendo al plan de marcación mucho más susceptible a cambios.

Un ejemplo de una situación típica podría ser la siguiente:

```
exten => 301,1,Answer( )
exten => 301,2,Playback(mens_bienvenida)
exten => 301,3,Hangup( )
```

En este ejemplo se puede apreciar la secuencia de comunicación cuando una llamada ingresa a la extensión 301 desde que se contesta, reproduce a quien llama un mensaje de bienvenida hasta que finalmente cuelga el canal. Los valores de prioridad desde el 2 pueden ser reemplazados por la letra **n** que representará exactamente la misma estructura y funcionamiento del sistema.

Asimismo, a esta prioridad **n** se le puede agregar una etiqueta que identifique una instrucción determinada pues sin un número de prioridad explícito, no podría ser alcanzada por una instrucción **Goto**[15] por ejemplo.

Tomando en cuenta las últimas consideraciones, la secuencia de instrucciones de la extensión 301 podría quedar como sigue:

```
exten => 301,1,Answer( )
exten => 301,n(bienvenida),Playback(mens_bienvenida)
exten => 301,n,Hangup( )
```

2.3.3.1.4 Aplicaciones

[15] La instrucción **Goto** es una aplicación del *dialplan* a la que se hará referencia en las páginas siguientes.

Las aplicaciones ejecutan comandos específicos que manejan de alguna forma una llamada. Estos comandos pueden reproducir un sonido, contestar una llamada, colgarla y muchas otras formas de manejar la comunicación telefónica.

Muchas de estas aplicaciones funcionan simplemente al ser llamadas mientras que otras requieren del paso de algunos parámetros, que se especifican entre paréntesis y separados por comas.

Dentro anexo B se explican varias de las aplicaciones que se utilizaron para la configuración del sistema de telefonía IP para la corporación MachángaraSoft aún cuando el conjunto de todas las aplicaciones instaladas en el sistema pueden listar mediante el comando **show applications**, dentro de la consola de *Asterisk* a la que se accede vía el comando **asterisk –r.** Asimismo el detalle y parámetros de cada función se obtiene con **show application <*nombre de la aplicación*>.**

Una descripción muy minuciosa de estas aplicaciones puede encontrarse fácilmente en Internet[16].

2.3.3.1.5 Variables

Dentro del plan de marcación, como en cualquier esquema de programación, la utilización de variables permite brindar mayor claridad a la lógica de configuración. Generalmente, a lo largo del *dialplan*, se utilizan variables para identificar el canal y protocolo VoIP a través del que un usuario se comunica con el servidor Asterisk, de modo que el argumento de la aplicación que permite el marcado hacia una extensión sea más fácil de comprender.

Por ejemplo, el marcado de la extensión 301 involucra la comunicación con el área de soporte, la instrucción sería:

```
exten => 301,1,Dial(SIP/soporte,10)
```

Lo anterior se daría en el caso de que el área de soporte utilice un dispositivo SIP. Es posible también definir inicialmente una variable SOPORTE que contenga la cadena de marcado, por ejemplo **`SPORTE=SIP/soporte`** y utilizarla cada vez que se haga referencia al marcado hacia la extensión correspondiente a ese departamento.

[16] http://www.dis.org.nz/asterisk/

```
exten => 301,1,Dial(${SOPORTE},10)
```

Hay que tomar en cuenta la forma cómo se llama a las variables definidas en el plan de marcación (entre llaves y precedidas de un signo **$**).

2.3.3.1.6 Macros

Los macros dentro del plan de marcación son el equivalente a las subrutinas en estructuras de programación. Al igual que las subrutinas, los macros permiten generalizar una secuencia de instrucciones de manera que pueda ser útil para varios casos diferentes, sin que tengan que repetirse todas las instrucciones una y otra vez si no que únicamente se llame a la ejecución del macro, pasándole los diferentes parámetros, según sea el caso.

Estos macros se definen entre corchetes y su nombre va precedido de la cadena **`macro-`** ([ ]), como por ejemplo, **`[macro-voicemail]`** y los argumentos que se le pasan se identifican como **`ARG1`**, **`ARG2`**, etc. Además se definen automáticamente otras variables que pueden utilizarse una vez que se llama a un macro, como por ejemplo, **`MACRO_EXTEN`** que devuelve el valor de la extensión desde la que fue ejecutado el macro.

El procedimiento más común dentro del plan de marcación es la de marcado a una extensión y si está ocupada o nadie atiende se envíe a un buzón de voz. Este proceso luciría así:

```
exten => 301,1,Dial(SIP/soporte,10,tT)
exten => 301,2,VoiceMail(u301@default)
exten => 301,102,VoiceMail(b301@default)
```

Si se dispone de varios cientos de extensiones, el trabajo de configuración anterior para cada una de ellas será arduo y mucho más cuando se desee cambiar algo en la funcionalidad de la secuencia. Ahora, si generalizamos la secuencia usando variables y un macro:

```
[macro-voicemail]
exten => s,1,Dial(${ARG1},10,tT)
exten => s,2,VoiceMail(u${MACRO_EXTEN}@default)
exten => s,102,VoiceMail(b${MACRO_EXTEN}@default)
```

Sería posible utilizar la secuencia anterior cada vez que se desee controlar el marcado hacia una extensión, como sigue:

```
exten => 302,1,Macro(voicemail,${SOPORTE})
```

Y eso sería suficiente para cada extensión adicional.

La estructura del plan de marcación se sustenta sobre estos conceptos, sin embargo, los rodean un sin número de herramientas, características y aplicaciones esencialmente, que le dan la funcionalidad y flexibilidad de que se jacta este sistema.

2.3.3.2 Lógica del Plan de marcación para la corporación

La lógica del plan de marcación detalla de forma general el comportamiento del sistema IP PBX para la corporación bajo cualquier escenario posible de comunicación.

Para el desarrollo del prototipo se plantea la utilización del protocolo de VoIP SIP en todas las extensiones de la corporación, por la facilidad que involucra su característica de manejo de estándares.

Se plantea también la utilización de una tarjeta genérica X100P[17] para la interconexión del servidor con la PSTN.

Si una llamada ingresa a la corporación a través de la PSTN, un control determinará si son horas de oficina y procesará la llamada, reproduciendo un mensaje de bienvenida junto con un menú de voz que indique, a la persona que llama, cómo alcanzar al departamento o empresa con el que desea comunicarse.

Si la llamada se ejecuta fuera de horas de oficina, un mensaje advertirá de esa situación y la llamada se colgará.

Los mensajes de voz de direccionamiento no indicarán las extensiones de áreas como gerencia, de modo que estas comunicaciones sean filtradas por la operadora.

Si una de estas extensiones se encuentra ocupada o no hay nadie disponible en ella, aún habrá la posibilidad de dejar un mensaje en el buzón correspondiente de

[17] En el anexo C puede consultarse la hoja de datos de esta tarjeta

voz para que el dueño de la extensión la recoja o la reciba como archivo adjunto mediante correo electrónico.

Dentro de las opciones de comunicación que el IVR presentará será la de marcar una extensión para obtener tono de fax de modo que el servidor *Asterisk* pueda recibir un fax.

El sistema esperará un tiempo de 15 segundos por el marcado de una opción dentro de un menú, luego de lo cual la llamada será terminada y mientras espera se reproducirá una melodía.

El primer menú permitirá direccionar las llamadas entrantes vía la PSTN hacia las empresas del parque, a la administración del mismo, hacia una operadora o a la extensión correspondiente al tono de fax. Un submenú permitirá redireccionar las llamadas hacia cada una de las empresas miembros y dentro de ellas a un área en específico (como soporte, ventas o educación).

Dentro del primer menú, se indicará también una opción mediante la que se pueda acceder al directorio de la Corporación a través del que podrá comunicarse con un miembro, digitando los 3 primeros dígitos de su apellido.

De igual forma, en el caso de que sea necesario, se habilitarán salas de conferencia a las que podrán acceder dispositivos internos y externos a la corporación al marcar una extensión (600 o 601) dentro del servidor *Asterisk.* Dependiendo de los requerimientos se podrá configurar un número límite de participantes dentro de cada sala de conferencia.

Es posible que quienes llamen marquen secuencias de dígitos que no sean correctas o que simplemente tomen demasiado tiempo en ingresar una. En el primer caso, un mensaje de voz alertará del error y se reproducirán nuevamente las instrucciones. En el segundo caso, el sistema se despedirá y colgará la llamada.

Todo esto permitirá controlar posibles eventos que se manifiesten de acuerdo a la forma en que podría interactuar el usuario con el sistema.

En la corporación, los miembros podrán comunicarse entre sí marcando directamente el número de extensión o incluso el nombre en caso de que manejen terminales con esa capacidad. Asimismo tendrán acceso al servicio de

directorio a través de la extensión 501 y al servicio de Buzón de Voz (*VoiceMail*) usando la extensión 500.

La comunicación desde la corporación hacia la PSTN se controlará a través de patrones de marcado que definirán el nivel de acceso a llamadas locales, regionales, internacionales, comerciales y a la red celular. Estos patrones y la creación de contextos de forma jerárquica y ordenadamente asignados permitirán determinar los permisos de acceso dependiendo de las necesidades estrictas de los departamentos.

Por ejemplo, el área de soporte generalmente recibirá llamadas de los clientes para la solución de algún inconveniente, por lo que no deberían tener acceso a llamadas a celular o internacionales. Los gerentes y la operadora, en cambio, deberán tener acceso a cualquier salida de comunicación, en el primer caso por la jerarquía del cargo y en el caso de la operadora por que es quien, por ejemplo, tendrá que transferir una llamada internacional a un miembro del área de soporte en algún caso especial autorizado por el supervisor.

En todos los casos el acceso hacia la PSTN se obtendrá anteponiendo el dígito 9 a cualquier número telefónico que se marque.

2.3.3.3 Elementos del Plan de marcación MachángaraSoft

En el anexo A se detalla el archivo de configuración y cada uno de los elementos mencionados.

2.3.3.3.1 Contextos

En la configuración del plan de marcación del sistema de telefonía IP para el MachángaraSoft se crearán 15 contextos en los que se organizará el tráfico de comunicaciones.

En los dos primeros contextos se definirán parámetros de funcionamiento del archivo de configuración (`[general]`) y variables globales (`[globals]`) a las que se hará referencia desde cualquier parte del *dial plan*.

El resto de contextos se describen a continuación:

```
[default]
```

Este contexto contendrá los patrones de marcado necesarios para que un dispositivo o usuario, al pertenecer a este contexto, pueda marcar y comunicarse con los números de emergencia como el 911 o el 101.

[entrante-pstn]

Este contexto manejará el servicio de IVR que se dará a todas las llamadas que ingresen al servidor a través de la tarjeta conectada a la PSTN, así como también re-dirigirá todas estas llamadas a las extensiones y servicios (directorio, buzón de voz, fax) correspondientes al interior de la corporación.

Además este contexto contiene un pequeño control que permite determinar si la llamada se ha ejecutado fuera de horas de oficina para reproducir un mensaje de advertencia.

[interno]

Este contexto maneja todas (y únicamente) las comunicaciones que se pueden establecer internamente entre los miembros de la corporación. Goza también de servicios como Directorio, Buzón de Voz y Sala de conferencia.

[local]

Este contexto permite la realización de llamadas locales desde la red interna hacia la PSTN. Las llamadas locales pueden iniciar con 2 o 3 seguido de un patrón de 6 dígitos más en nuestro país.

El contexto local incluye al contexto interno, pues una entidad que tenga acceso a la realización de llamadas locales (dentro de la provincia), con mucha más razón podrá comunicarse internamente con otras entidades. Se incluye también la posibilidad de realizar llamadas gratuitas a través del prefijo 1800.

[regional]

El contexto regional, como es obvio, permite la realización de llamadas a otras provincias. El patrón que se verifica es el marcado de los prefijos 02, 03, …07 correspondiente a cada provincia y el prefijo de llamada local 2 o 3. Se incluye también el contexto local.

[internacional]

Este contexto permite el acceso a llamadas internacionales. Si se tiene discado directo internacional el patrón de control de acceso sería 00. El resto de dígitos dependerá del código de país y de región y por tanto será variable
El contexto regional y por tanto el interno y local están incluidos en éste.

[celular]

El patrón de control de acceso a la red celular a través de la PSTN es por supuesto el 09 más 7 dígitos adicionales. Este contexto se asignará únicamente a las áreas de gerencia y a la operadora, pues este tipo de llamadas son las que más costo representan y, por tanto, las que deben estar más restringidas.

[comerciales]

Las llamadas comerciales involucran el marcado de prefijos 1900 y 1700 que, al tener costo, deben ser manejados de manera cuidadosa. Se asignará esta posibilidad a los gerentes y a la operadora solamente.

A continuación se describen contextos más específicos que se crearon para asignar por áreas los “privilegios” de comunicaciones haciendo que esta asignación sea mucho más manejable. Este contexto está conformado únicamente por las instrucciones de inclusión de los contextos anteriores.

[gerencia]

Este contexto representa al área de gerencia que generalmente tendrá mayores facilidades de acceso y por esa razón se incluyen los contextos *default*, interno, local, regional, internacional, celular y comerciales.

[operadora]

La operadora también posee acceso a todos los contextos y por ello se incluye en este contexto todos los contextos generales antes mencionados.

[soporte]

Al área de soporte de cada empresa se le permitirá únicamente la realización de llamadas locales y de emergencia, por tanto este contexto incluirá a su vez los contextos *default* y local.

[ventas]

El área de ventas de las empresas posiblemente necesitará acceso a llamadas telefónicas a nivel nacional. Se incluirá aquí el contexto *default* y regional.

[pagos]

El área de pagos generalmente recibirá llamadas de los proveedores y no necesitará de un acceso telefónico completo. Se incluirá el contexto *default* y local.

[clientes]

Este contexto particular permitirá a los clientes registrarse en el servidor de la corporación a través de Internet y comunicarse con las empresas sin tener que utilizar la PSTN o la red celular. Este debería incluir el contexto **interno** de modo que cualquier cliente pueda utilizar el servidor únicamente para comunicarse al interior de la corporación

2.3.3.3.2 Patrones de Marcado

Para el manejo de llamadas, hacia la PSTN especialmente, resulta de mucha utilidad el uso de patrones de marcado. Es obvio que no es factible crear una extensión para cada número posible que pueda marcar un usuario que se conecta hacia la PSTN u otra red de comunicaciones pública. Entonces es necesario manejar la comunicación mediante prefijos de marcación, los patrones de marcado que indican que alguien marca hacia una red (telefonía celular o fija), hacia una ubicación (local, regional, internacional) o requiriendo algún servicio en específico (servicio de emergencia, llamadas comerciales 1800 o 1700).

Dentro del plan de marcación (*dialplan*) un patrón representa una extensión cuya identificación puede variar y se representa con un sub-guión (_) seguido del patrón que se desea controlar.

En este sentido existen algunos caracteres y combinaciones que pueden ser de utilidad:

TABLA 2.6 CARACTERES ESPECIALES PARA LA IDENTIFICACIÓN DE PATRONES EN EL DIALPLAN

CARÁCTER/COMBINACIÓN	DESCRIPCIÓN
X	REPRESENTA A CUALQUIER DÍGITO DEL 0 AL 9
Z	REPRESENTA CUALQUIER DÍGITO DEL 1 AL 9
N	REPRESENTA CUALQUIER DÍGITO DEL 2 AL 9
[12346-9]	CUALQUIER DÍGITO DENTRO DE LOS CORCHETES: 1, 2, 3, 4 Y DEL 6 AL 9
.	MÁSCARA QUE REPRESENTA CUALQUIER NÚMERO DE DÍGITOS.

La extensión que representa el marcado hacia números locales (dentro Pichincha) se identificaría de la siguiente forma_`[23]NXXXXX`.

Este patrón representa a cualquier número de 7 dígitos que inicien con 2 o 3, es decir, justamente el patrón que permite marcar hacia cualquier número telefónico dentro de la provincia de Pichincha.

2.3.3.3.3 Extensiones

Dentro del dial plan la asignación de las extensiones estará de acuerdo a la tabla 2.7.

Cada una de estas extensiones dentro del *dialplan* estará registrada en este caso en el archivo **`/etc/asterisk/sip.conf`**, pues se ha planteado la utilización de este protocolo de VoIP.

Se crearán las extensiones SIP **`[operadora]`** y **`[admincorp]`** para el registro respectivo de los dispositivos SIP de la operadora y del administrador de la corporación. En cuanto a las empresas, dependiendo de los servicios que presten se crearán las siguientes extensiones (por ejemplo en el caso de la empresa *Refundation Consulting Group*): **`[pagos_ref]`**, **`[soporte_ref], [ventas_ref]`**, **`[gerente1_ref]`**, **`[gerente2_ref]`**. Todas ellas correspondientes a cada uno de los departamentos de las empresas.

TABLA 2.7 DISTRIBUCIÓN INTERNA DE EXTENSIONES EN LA CORPORACIÓN.

Nº Ext.	Descripción
1	Administración de la corporación
2	Indicaciones de direccionamiento hacia las empresas
201	IVR de Refundation Consulting Group
202	IVR de NDeveloper
203	IVR de SoporteLibre
204	IVR de MagmaSoft
205	IVR de DreamQuest
26	IVR de LogicStudio
207	IVR de SantaFe
208	IVR de NeoQuality
209	IVR de DEcuador
300	Recepción de Fax
301	Ventas Refundation Consulting Group
302	Soporte Refundation Consulting Group
303	Educación Refundation Consulting Group
304	Pagos Refundation Consulting Group
305	Gerente 1 Refundation Consulting Group
306	Gerente 2 Refundation Consulting Group
311-316	Departamentos NDeveloper
321-326	Departamentos SoporteLibre
331-336	Departamentos MagmaSoft
341-346	Departamentos DreamQuest
351-356	Departamentos LogicStudio
361-366	Departamentos SantaFe
371-376	Departamentos Neoquality
381-386	Departamentos DEcuador
500	Manejo de Correo de voz
501	Directorio
600	Sala de conferencia 1
601	Sala de conferencia 0
610	Extensión para clientes
700	Extensión de parqueo

2.3.3.3.4 Aplicaciones

Ya se ha mencionado de manera general todas las aplicaciones de que estará conformado el plan de marcación de la corporación *MachángaraSoft*, ahora se describe el funcionamiento de cada una dentro del archivo de configuración:

`Answer([retardo])` Esta aplicación permite responder un canal si existe una señal de timbrado. Si se especifica un tiempo de retardo (que estará en milisegundos), el servidor esperará ese tiempo antes de contestar la llamada.

`Dial(tecnología/recurso&[tecnología/recurso2]`*`,[timeout],`*`[opciones])`
Conecta una llamada que recibe el servidor con un canal especificado como **`tecnología/recurso`**. Esta aplicación se utilizará con 3 de sus parámetros de funcionamiento. El parámetro **`tecnología/recurso`** se refiere al protocolo de VoIP (*SIP, IAX2, Zap*) que se utiliza para comunicarse con un canal o extensión registrada en el servidor. Utilizando el operador **`&`** es posible intentar la conexión con varios canales a la vez.
El parámetro **`[timeout]`** determina el tiempo máximo que timbrará una extensión hasta que el estado del canal sea considerado como *unavailable* o no disponible.
Las posibles opciones de esta aplicación pueden consultarse mediante el comando **`show application Dial`**, dentro de la consola de Asterisk.

`Hangup()` Cierra un canal que está siendo utilizado.

`Wait(segundos)` Esta aplicación espera el número de segundos especificado y continua luego de ello con la siguiente prioridad.

`Playback(archivo de sonido&[archivo de sonido 2])` Reproduce uno o varios archivos de sonido que se pasan como parámetros. No permite interacción con el usuario.

`Background(archivo de sonido&[archivo de sonido 2])` Reproduce uno o varios archivos de sonido que se pasan como parámetros, mientras se espera por el marcado de un patrón extensión. El momento que el patrón se marca, se interrumpe la reproducción del archivo de sonido.

Set(nombre1=valor1[,nombre2=valor2]) Permite establecer el valor de una variable de canal. En el *dialplan* propuesto se utiliza esta aplicación para determinar el valor de algunos *timeouts.*

Goto([[contexto,]extension,]prioridad) Es una función que permite saltar a una prioridad, extensión y/o contexto en particular. En el *dialplan* se utiliza esta aplicación cuando un usuario digita un patrón incorrecto (saltando a la extensión i) para reproducir nuevamente las instrucciones del IVR.

GotoIf(condicion?[etiquetasiverdad]:[etiquetasifalso]) Es un **Goto** condicional. Si la condición se cumple, la aplicación salta a la primera etiqueta, si es falsa, salta a la segunda. Se utiliza para limitar el número de participantes en una sala de conferencia. Si es mayor que un número determinado, no se permite la conexión a la sala.

GotoIfTime(<horas>|<diassemana>|<diasmes>|<meses>?[[contexto|]extensión|]prioridad) Permitirá el salto a una ubicación específica dentro del plan de marcación si el tiempo especificado como parámetro corresponde con el tiempo actual. El tiempo se refiere a la fecha, hora, día de la semana especificados.

WaitExten(segundos,opciones) Esta aplicación espera que el usuario ingrese una extensión por un número de segundos mientras se reproduce una melodía si esto e especifica en las opciones.

MeetMe([númerodesala],[opciones]) Permite ingresar al usuario que marca la extensión en la que se especifica esta aplicación dentro de una conferencia.

En este plan de marcación se especifican opciones para que cada vez que un usuario ingrese o salga de la sala, todos los usuarios sean advertidos de esos eventos.

`MeetMeCount(númerodesala,[variable])` Permite reproducir el número de usuarios existentes en una sala de conferencia o en el caso de que se especifica una variable, el valor correspondiente al número de participantes se retornará en la variable. El conteo de participantes permite en este caso limitar su número.

`Congestion()` Esta aplicación indica una condición de congestión al canal que llama en caso de que la llamada no se pueda establecer.

2.3.4 CONFIGURACIÓN DE SONIDOS

Si se desea configurar un idioma específico para la reproducción de mensajes de Asterisk se debe especificar como **`es`** el valor de **`language`** (`language=es`) en el archivo **`/etc/asterisk/zapata.conf`** y en los archivos **`sip.conf`** e **`iax.conf`**, si se utilizan esos protocolos de VoIP entre las extensiones.

Una vez que se ha configurado en estos archivos el idioma, se debe descargar el *set* de sonidos para español. Estos *sets* existen para varios acentos: colombiano, argentino, español[18].

Estos son archivos comprimidos en formato *gsm* que deben copiarse dentro del directorio **`/var/lib/asterisk/sounds`** en una carpeta con el nombre **`es`**.

Algunos de los sonidos específicos que se deben crear con el nombre de la corporación y de las empresas deben grabarse con cualquier aplicación para ese efecto. Se utiliza una aplicación llamada *Audacity* para grabar estos mensajes que se generan en formato *wav*. Para convertir cada archivo a formato *gsm* (que es el que básicamente maneja *Asterisk*) se puede utilizar el comando[19]:

```
# sox mensaje.wav -r 8000 -c1 mensaje.gsm resample -ql
```

Si se guarda el archivo generado dentro del directorio de sonidos se podrá llamarlo desde el *dialplan* tan solo por su nombre, sin la extensión *gsm.*

2.3.5 CONFIGURACIÓN DE LOS CLIENTES DE TELEFONÍA

[18] http://www.asterisk-es.org/modules/mydownloads/visit.php?cid=1&lid=11, http://www.voip-info.org/tiki-index.php?page=Asterisk+sound+files+international

[19] http://www.voip-info.org/tiki-index.php?page=Convert+WAV+audio+files+for+use+in+Asterisk

La solución menos costosa y más práctica resulta ser la adquisición de adaptadores con 2 puertos FXS que permiten la conexión de 2 teléfonos analógicos a la red IP y además la compleja funcionalidad de los teléfonos IP no resulta necesaria.

La configuración ya sea de *softphones*, adaptadores ATA o teléfonos IP se resume en darle al dispositivo una dirección IP, el protocolo de comunicación de VoIP, la dirección IP del servidor *Asterisk*, el nombre de usuario y la contraseña del cliente.

2.3.5.1 Softphones

Se puede detallar aquí un ejemplo de configuración de un cliente *softphone* para *Windows* (XLite) y otro para Linux (*Ekiga*).

La versión de *XLite* que se configuró es la 3.0. La descarga e instalación son muy sencillas[20]. Una vez con este *softphone* instalado, la configuración tomará muy pocos segundos.

Los parámetros IP de *XLite* se tomarán directamente de los configurados en la máquina en donde se instaló, por esa razón lo que resta es configurar la dirección IP del servidor Asterisk y el nombre de usuario y contraseña de quien se conecta a él. Esto se logra dando *click* derecho sobre el teléfono y escogiendo la opción *SIP Account Settings*. Luego damos *click* en *Add* para agregar una cuenta y la editamos.

[20] http://www.counterpath.com/index.php?menu=download_xlite&platform=win

FIGURA 2.1 XLITE SOFTPHONE

Los parámetros de importancia relevante son:

Username: Donde debe ir el nombre del usuario o la extensión SIP que se creó en el servidor *Asterisk*.

Password: Que contiene la contraseña con que se configuró la cuenta anterior en el servidor.

Authorization user name: El mismo nombre de usuario.

Domain: Que debe contener la dirección IP o el nombre de *host* del servidor *Asterisk* al que se conectará el *XLite*.

Una vez configurados estos parámetros, se acepta los cambios y en la pantalla del *softphone* deberá indicar que el usuario está registrado.

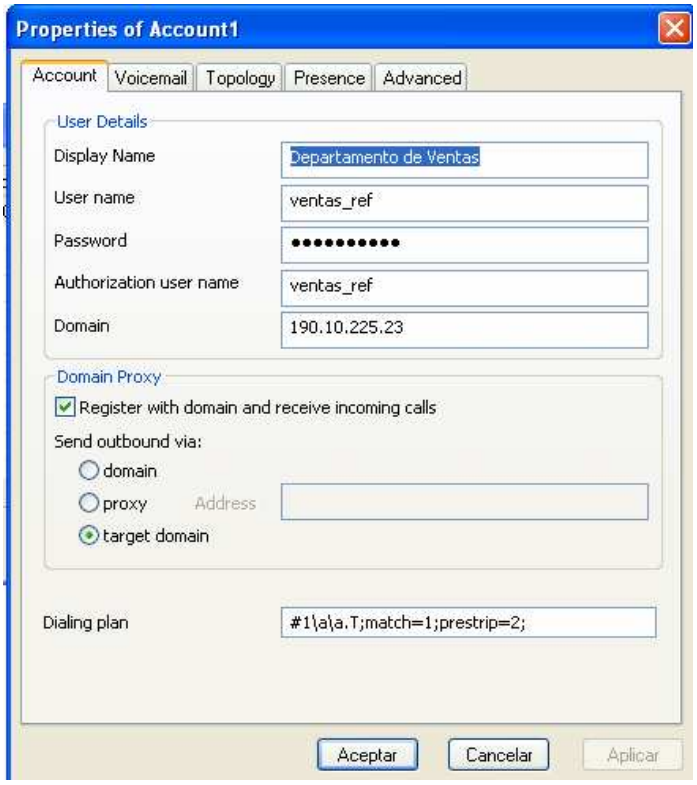


FIGURA 2.2 CONFIGURACIÓN DE XLITE PARA CONEXIÓN CON ASTERISK

En Linux es ya muy común la utilización del software telefónico llamado *Ekiga* cuya configuración es igual de sencilla que la anterior.

FIGURA 2.3 INTERFAZ DE MARCADO TELEFÓNICO EKIGA

Para configurar los parámetros de conexión y autenticación con el servidor Asterisk se deberá seguir dos pasos fundamentales.

- En el menú *Edit*, en la Opción *Preferences*, se configuran los parámetros de red del cliente *softphone*, asignando la dirección IP a través de la que la aplicación se conectará al servidor de telefonía.

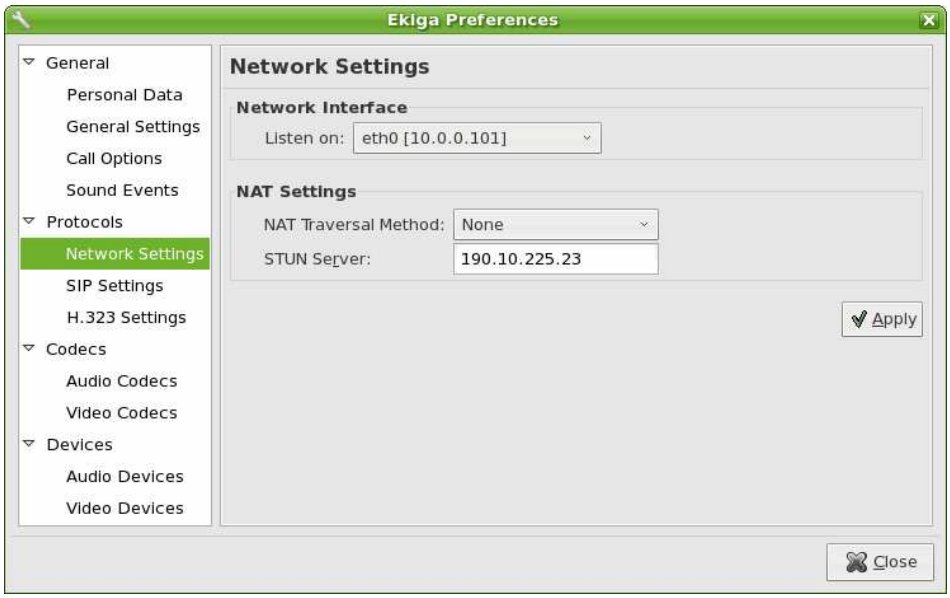


FIGURA 2.4 CONFIGURACIÓN DE DIRECCIÓN IP DEL CLIENTE EKIGA

- Al igual que en el caso de Xlite y por supuesto de cualquier tipo de cliente ya sea software o hardware, deben configurarse los parámetros de autenticación con el servidor Asterisk y también la dirección IP en la que este último escucha las peticiones de conexión.

En este caso el servidor está en Internet y tiene una dirección 190.10.225.23. El cliente que se registra pertenece al departamento de pagos (`pagos_ref/pagos_ref`).

FIGURA 2.5 CONFIGURACIÓN DE PARÁMETROS DE AUTENTICACIÓN EN CLIENTE EKIGA

Eso será todo, luego de probar los dispositivos de sonido, el cliente estará listo para registrarse con el servidor y acceder a las posibilidades de comunicación que ofrece.

2.3.5.2 Adaptadores

Para la configuración de los ATAs se tomará como referencia el modelo *Lynksys* PAP2-EU con dos puertos FXS. Las especificaciones se detallan en el anexo C.

Este ATA tiene un puerto Ethernet y dos puertos FXS al que se podrá conectar 2 teléfonos respectivamente.

La configuración de este adaptador es por demás sencilla[21]. Se debe conectar la alimentación correspondiente al dispositivo, mediante un cable de red directo se conecta el puerto Ethernet con el *switch* de la empresa y se conecta el teléfono analógico a uno de los puertos FXS. Ya debe haber tono en el teléfono y si se dispone de un servicio de DHCP, éste le asignará una dirección IP. Para averiguar esta dirección se levanta el auricular del teléfono, se marca **** y luego de escuchar el mensaje de bienvenida se marca 110#. Luego de este proceso, un mensaje de voz en inglés revelará la dirección IP del dispositivo.

Desde cualquier ordenador dentro de la red se abre un *browser* y se accede vía web al ATA al digitar en el espacio correspondiente a la URL la dirección IP que se obtuvo.

[21] http://www.linphone.org/, http://www.wirlab.net/kphone/

Será necesario que se lleve a cabo un proceso de autenticación (nombre de usuario/contraseña) para que sea posible acceder a los paneles de configuración. Existe una gran cantidad de opciones de configuración pero solo unas pocas interesan para este caso.

- El paso inicial es darle al dispositivo una dirección IP fija. A esa opción se accede a través de la pestaña **System** (figura 2.6) donde será posible configurar todos los parámetros de red necesarios para que el adaptador pueda comunicarse con el servidor de telefonía *Asterisk.* En este caso sería: dirección IP, máscara de red, Dirección *Gateway* y servidores DNS, entre otros.

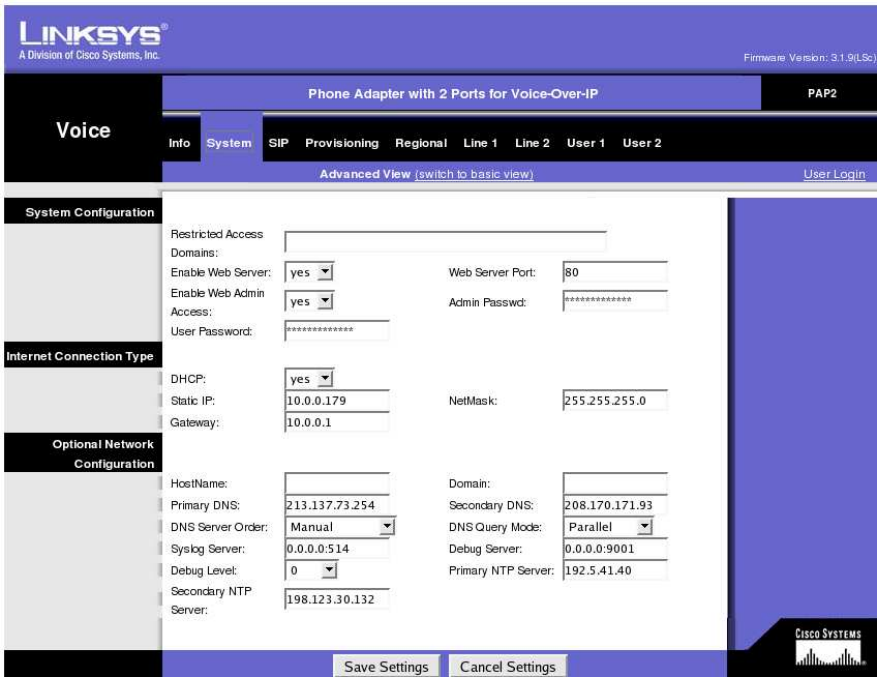


FIGURA 2.6 CONFIGURACIÓN DE PARÁMETROS DE RED DEL ATA.

- El siguiente paso es configurar en el dispositivo los parámetros de acceso al servidor *Asterisk*, como es la dirección IP, el puerto al que deberá conectarse y, desde luego, el nombre de usuario y contraseña de la línea conectada. Este proceso se realiza en la pestaña **Line1** (o Line2 si se configuran las 2 líneas) del panel. Como se puede observar en la figura 2.7.

Se debe tener especial cuidado con la configuración del *códec* a través del que se comprimirá la información, pues debe estar de acuerdo con los códecs con los que *Asterisk* puede y permite trabajar, en este caso en el archivo correspondiente a las extensiones SIP; **`sip.conf`**.

En la pantalla de la pestaña *Info* (figura 2.8) se puede observar el estado de las líneas conectadas al ATA, es decir si se registraron (estado de registro) y si están abiertas o cerradas.

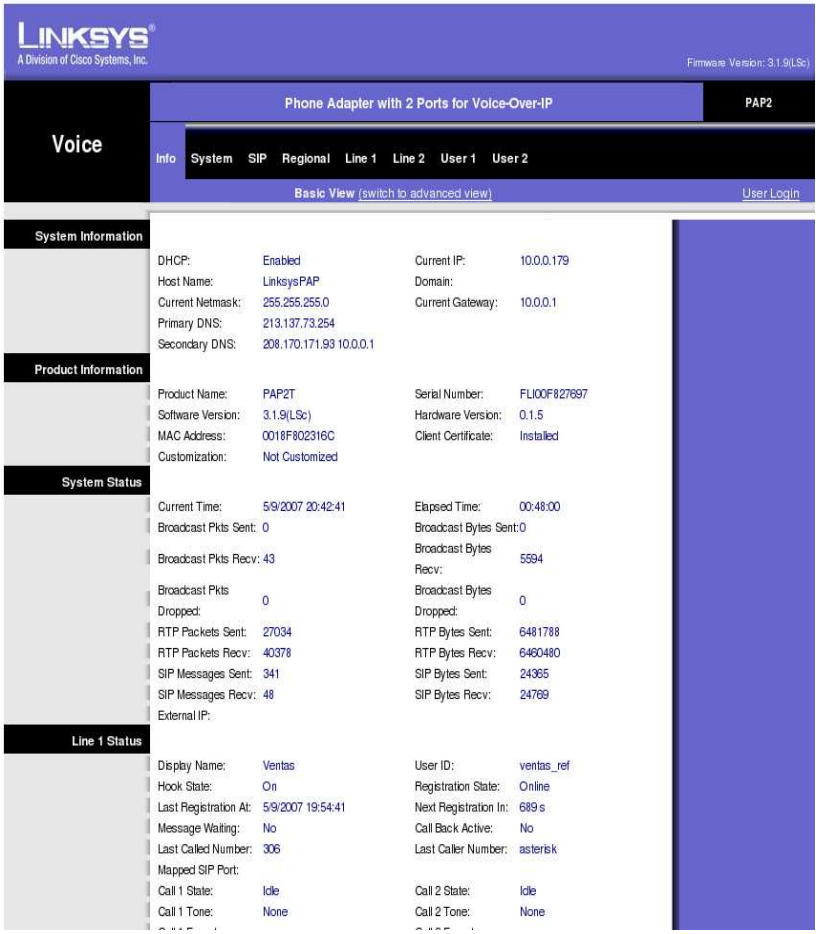


FIGURA 2.7 CONFIGURACIÓN DE PARÁMETROS DE CONEXIÓN AL SERVIDOR ASTERISK.

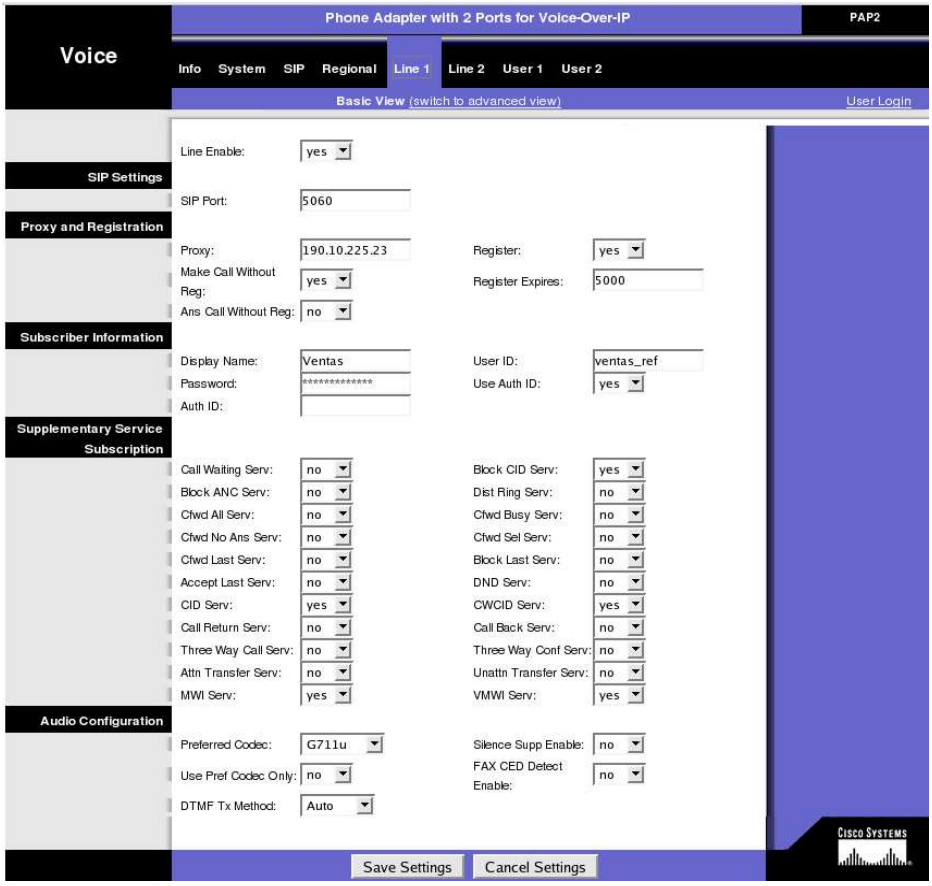


FIGURA 2.8 PANTALLA DE INFORMACIÓN DE CONFIGURACIÓN DEL ATA LINKSYS PAP2-EU

2.4 CONSIDERACIONES RESPECTO A DISPOSITIVOS FIREWALL/NAT

Si los dispositivos clientes (*softphones*, teléfonos IP, ATAs) necesitan registrarse y mantener comunicación con el servidor *Asterisk*, y se encuentran detrás de dispositivos que hacen NAT, estos se configurarán en el servidor, en el archivo **`sip.conf`** (en el caso de dispositivos SIP) con la opción **`nat=yes`** para que la comunicación SIP y RTP pueda fluir adecuadamente.

Por otro lado, el *firewall* se configurará con reglas que permitan el flujo de información a través de los puertos de comunicaciones que sean utilizados por los protocolos de VoIP determinados. En el caso de SIP, el puerto a través del que se transmite señalización es el 5060 (puerto en el que el servidor *Asterisk* escucha las peticiones SIP) mientras que todos los datos se transmiten de forma aleatoria a través de los puertos 10000-20000, aunque este rango puede reducirse editando el archivo **`/etc/asterisk/rtp.conf.`**

En el caso de IAX2 (protocolo de VoIP de Asterisk) el manejo de la comunicación a través de un *firewall/NAT box* es más sencillo pues la señalización y datos viajan a través de un único puerto.

Es comprensible que en el caso de SIP se necesita permitir abrir demasiados puertos lo que sin lugar a dudas significa un riesgo de seguridad. Lamentablemente existen en el mercado muy pocos dispositivos de VoIP que funcionen con IAX2, limitando su configuración.

FIREWALL PARA LA CORPORACIÓN MACHÁNGARASOFT

2.5 ANÁLISIS ACTUAL DE LA RED

Con el fin de determinar las necesidades específicas de seguridad de la red de la corporación MachángaraSoft, es sin duda importante determinar el estado actual de la misma. Este análisis permitirá establecer los lineamientos hacia los que se orientará el diseño del sistema de seguridad.

Como premisa se puede mencionar que la corporación se encuentra formada por alrededor de 15 empresas jóvenes, 9 de las cuales trabajan en un solo ambiente, en el tercer piso del Edificio Santorini (entre las calles Mariano Aguilera y La Pradera) y hacia las cuales se enfocará el diseño y configuración.

2.5.1 INFRAESTRUCTURA DE RED

La pequeña red de datos de la Corporación permite la conexión a Internet de todos sus miembros representados por las 9 empresas que allí trabajan. La información respecto de la infraestructura se obtuvo a partir de una inspección manual realizada con la autorización de los miembros del parque.

Se puede verificar la existencia de alrededor de 21 estaciones de trabajo, con un promedio de 3 estaciones por empresa.

2.5.2 ESTADO ACTUAL

El papel de la estructuración de la red de la corporación es fundamentalmente permitir el acceso a Internet a los miembros de cada empresa a través de un "*router*" y del cable-módem de la empresa *TVCable*.

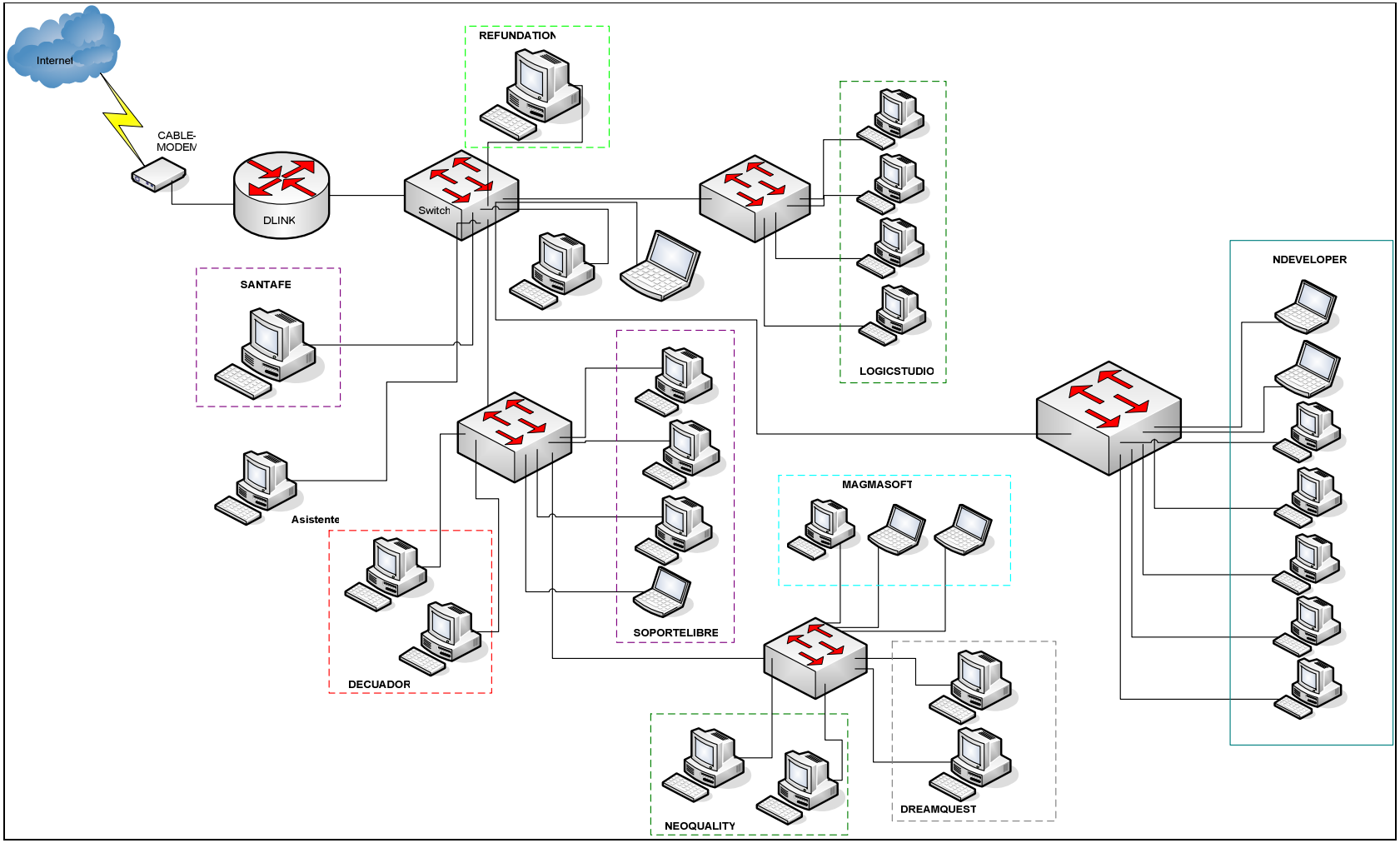


FIGURA 2.9 ESTRUCTURA DE LA RED DE LA CORPORACIÓN MACHÁNGARASOFT

Casi en su totalidad está compuesta por equipos Pentium 4, con procesador superior a 2 GHz y 512 MB de memoria RAM.

El cableado a través del que se interconectan los equipos utiliza tecnología Ethernet IEEE 802.3 con categoría 5 y conectores RJ45. El método de control de acceso al medio es CDMA/CD, bastante común en las redes LAN Ethernet.

2.5.3 RECURSOS INFORMÁTICOS Y CONEXIÓN A INTERNET

Estaciones de trabajo.- Entre las máquinas correspondientes a las 9 empresas y a la asistente suman alrededor de 21 PCs, todas ellas con una conexión a Internet a través de un *router* D-LINK. La mayoría de estaciones utiliza alguna distribución del sistema operativo Linux, lo que favorece notablemente la seguridad de la red. Las características son esencialmente las mismas, es decir, Procesador Intel Pentium 4 mayor a 2 GHz y 512 MB o más en RAM.

Al momento no existen servidores de correo, DNS o de proxificación.

2.5.3.1 Ancho De Banda

El ancho de banda que comparten todas las empresas para su comunicación con Internet es de 400 Kbps para *dowload* de información y 150 Kbps para *upload.*

2.5.3.2 Topología y Distribución

En general, la conexión de cada una de las empresas y sus miembros con el ruteador se realiza en forma de una estrella extendida, compuesta por 5 switches pequeños conectados en cascada. Entre las empresas, la más grande está conformada por 8 miembros (que corresponden a 8 estaciones de trabajo) y la más pequeña está formada por 3 miembros.

2.5.3.3 Administración de la Red

Se puede apreciar que la red es aún pequeña, aunque el acelerado crecimiento y desarrollo de las empresas que la conforman hace pensar que en los próximos meses, estas cifras se duplicarán sin problema. Por ese motivo es primordial que exista un mecanismo de administración de los recursos informáticos y de la conexión a Internet de forma que puedan utilizarse adecuadamente por todos los

miembros del parque. Esto se hace palpable ya que la cantidad de usuarios de la red va creciendo de manera tal que se hace necesario el incremento de la seguridad en el acceso a otras redes (especialmente Internet) y, desde luego, el incremento del ancho de banda en función del crecimiento de la población de la corporación. Asimismo, es esencial el monitoreo periódico de la red interna, de manera que se pueda detectar violaciones de las políticas de seguridad de la red, y de la utilización de los recursos, especialmente los correspondientes al acceso a Internet.

En general, no existe un mecanismo formal de administración o monitoreo de la red a parte del que permite el pequeño ruteador que realiza el enmascaramiento IP de la red interna, la asignación dinámica de direcciones IP y la redirección de peticiones DNS hacia el Internet.

En este sentido es necesario el mantenimiento de los componentes de la red, incluyendo las estaciones de trabajo y los medios de conexión. Asimismo se necesitan políticas de monitoreo, control de acceso y herramientas que permitan obtener un diagnóstico continuo del funcionamiento de la red.

2.5.3.4 Aplicaciones en la Red

No existen aplicaciones internas propias de la red, más que las que ofrece el dispositivo de enrutamiento, es decir la aplicación del protocolo NAT que permite la salida a Internet, y la asignación de direcciones IP (DHCP) a las estaciones internas.

2.5.3.5 Gestión de Recursos

La gestión de recursos es complicada puesto que los activos (esencialmente las estaciones de trabajo) son propiedad de cada una de las empresas que, aún cuando forman parte del parque, poseen recursos propios e independientes. En este sentido es necesario establecer políticas que permitan llevar un registro del software que se instala, así como de los usuarios que acceden a la red y de los equipos que se conectan a ella.

2.5.3.6 Acceso a Internet

El acceso a Internet se logra a través de un módem perteneciente a la compañía TVCable, que posee dos puertos de acceso, cada uno de los cuales brinda una dirección IP pública asignada dinámicamente, mediante una conexión de 400 Kbps de bajada y 150 Kbps de subida, con canal compartido.

El resultado del monitoreo de consumo de ancho de banda muestra que la velocidad de descarga (en color verde) de información es más o menos de 34 KB/s, lo que representa 272 kbps. Además el tráfico de subida (en color azul) es prácticamente nulo.

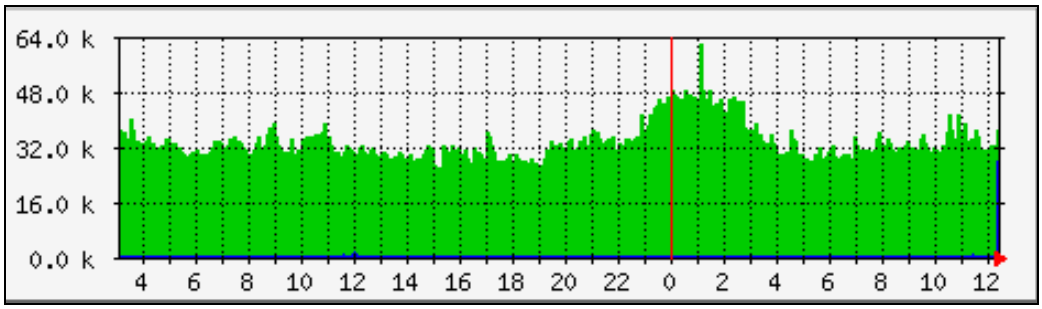


FIGURA 2.10 MEDICION DE TRAFICO EN UN DÍA NORMAL

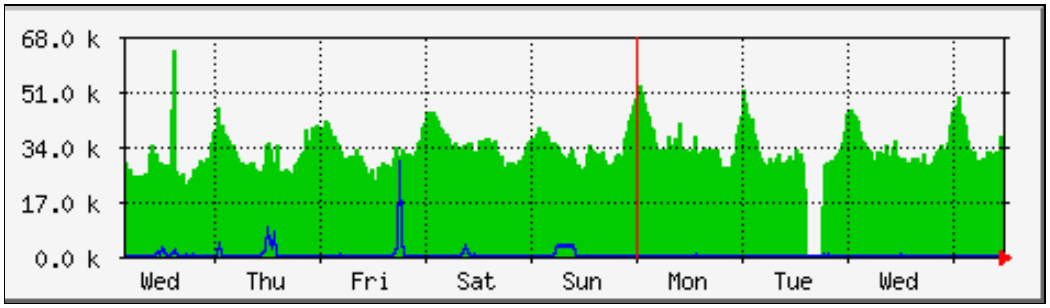


FIGURA 2.11 MEDICION DE TRAFICO DURANTE UNA SEMANA

Un *router* de marca DLINK es el encargado de permitir la conexión de toda la red interna hacia la WAN en Internet y es, desde luego, el responsable del enrutamiento de todas las conexiones y del enmascaramiento IP (NAT).

2.5.4 VULNERABILIDADES

Luego del breve análisis anterior se puede concluir que existen algunos puntos muy débiles en la red.

En la parte física, no hay un estándar de etiquetado que permita identificar al puerto de cada switch con la estación que está conectada, por lo que la simple desconexión de un cable se convertirá en un inconveniente al identificar las fallas.

Asimismo, el cableado no cumple con las normas de construcción, lo que podría provocar una disminución importante de la capacidad de transmisión de datos.

Los dispositivos de interconexión tampoco se encuentran sujetos, por lo que corren el riesgo de sufrir daños serios al golpearse.

En general no existen políticas que privilegien la seguridad física de los equipos como una ventilación adecuada, el uso de detectores de humo, aseguramiento de los equipos contra descargas de voltaje y cortes de energía, etc.

En cuanto a la seguridad del software, no existen tampoco prácticas de seguridad definidas pues la mayor parte de las empresas trabaja probando y evaluando distintas aplicaciones. Pese a esto, en las empresas que manejan un sistema operativo como Windows, por ejemplo, no hay un mecanismo de actualización de antivirus, debido al costo elevado de las licencias. Por este mismo motivo no se manejan políticas de actualización o parchado, que involucran el pago de dinero por el software requerido.

2.6 REQUERIMIENTOS DE CONEXIÓN DE LA CORPORACIÓN

Fundamentalmente, para el diseño de un firewall de red, se pueden aplicar dos políticas:

- *Denegar todo el tráfico que no está explícitamente permitido* o
- *Permitir todo el tráfico, excepto el que está explícitamente denegado*.

Es comprensible que la primera es una política *restrictiva* y, por tanto, mucho más segura que la segunda, más *permisiva.* La política restrictiva involucrará mucho más trabajo para el administrador pues se deberá investigar las necesidades de conexión específicas de los usuarios de la red interna, de modo que pueda habilitar todos los servicios correspondientes y solamente aquellos.

Para asegurar la eficiencia en el diseño del firewall se escoge la aplicación de una *política restrictiva* que evite cualquier tipo de tráfico que no esté explícitamente permitido por el administrador a través de la implementación de las respectivas reglas de tráfico.

Para que el firewall cumpla su cometido en este caso, la labor, más que proteger contra vulnerabilidades específicas, es abrir básicamente los caminos (puertos) de comunicaciones necesarios para que los usuarios tengan una experiencia satisfactoria de conexión al Internet, es decir que todos los servicios a los que están acostumbrados (y que estén permitidos usar) estén disponibles. Por ello es necesario analizar brevemente algunos de los requerimientos (o servicios) de conexión de este caso en particular.

2.6.1 REPORTE DE ERRORES VÍA ICMP

La utilización de un protocolo como ICMP resulta necesaria para poder determinar la existencia o actividad de los *hosts* en Internet o incluso del equipo de *firewall*. Los usuarios, por tanto, requerirán que se permita (por parte del *firewall*) el reenvío de paquetes ICMP (*Internet Control Message Protocol*), así como también el administrador considerará de utilidad que la interfaz de conexión con Internet junto con la interna puedan recibir (INPUT) estos paquetes de control para determinar el estado de actividad del equipo de seguridad.

En resumen, el dispositivo de *firewall* deberá permitir el reenvío (FORWARD) de paquetes ICMP desde la red interna hacia el Internet e igualmente deberá aceptar todos los paquetes ICMP provenientes de la red interna dirigidos a la interfaz interna y los paquetes provenientes del Internet dirigidos a la interfaz externa.

2.6.2 NAVEGACIÓN HTTP

El objetivo principal de conexión a Internet es, por supuesto, la navegación en búsqueda de información. Debido al hecho de estar formada por empresas eminentemente dedicadas al desarrollo tecnológico, de infraestructura y software, el acceso a páginas web con información de ese estilo es muy importante, si no vital. Por ello el firewall debe permitir este tipo de tráfico, reservándose, desde luego, la potestad de restringir ciertas páginas que no tengan relación con la investigación tecnológica y que puedan poner en peligro los sistemas internos o el desempeño de los miembros de las empresas. Se hace referencia en este caso a páginas de pornografía o de entretenimiento inútil para los intereses de las empresas.

Aún cuando la filosofía de acceso al Internet es bastante abierta, conforme las empresas van creciendo, se hace necesaria la implementación de políticas al respecto que permitan controlar el uso de los recursos de red, pues al ser compartidos, no se debe permitir abusar de ellos o darles mal uso.

La utilización de la herramienta *iptables* en Linux permite el manejo de políticas a nivel de capa de enlace, de red y de transporte, por lo que nuestro firewall deberá valerse de otra herramienta que permite el filtrado a nivel de capa de aplicación para el protocolo HTTP, tal es el caso de los *proxy* de aplicación como el *Squid*, que viene en todas las distribuciones de Linux por defecto y permite la aplicación de reglas de control de acceso en función de la dirección URL, palabras típicas, la extensión de los archivos que se cargan (como *applets* de Java), ligero control en el ancho de banda mediante la utilización de un caché y algunas otras funcionalidades bastante útiles.

Para la navegación web se hacen las peticiones al puerto 80 en la capa de transporte, razón por la que se deberá aceptar cualquier conexión desde la red interna cuyo puerto de destino sea el mencionado, así como los paquetes relacionados con la conexión pertinente. Además, se utilizará el servicio de proxificación de *Squid* en el mismo sistema de firewall que escuchará las peticiones en el puerto 8080, de manera que la configuración del dispositivo de seguridad deberá permitir el paso de las conexiones entrantes en ese puerto. Para aprovechar las capacidades de enmascaramiento IP se configurará el *Proxy-firewall* de tal forma que sea transparente, es decir que los usuarios de la red interna no tengan que configurar sus *browsers* (navegadores) para apuntar a la dirección del proxy, una labor que puede resultar tediosa para ellos si se tiene que realizar constantemente.

El firewall deberá permitir explícitamente el tráfico de reenvío (FORWARD) desde la *intranet* cuyos paquetes tengan como destino el puerto 80, que serán redirigidos para que salgan a Internet a través del servicio *proxy*, razón por la que el firewall deberá permitir también el tráfico entrante (INPUT) desde la red interna cuyo puerto destino sea el 8080, correspondiente al servicio de proxificación.

Un pequeño detalle que se debe tomar en cuenta es que la navegación a través de HTTPS (HTTP que utiliza conexiones SSL) se establecen en un puerto

diferente al 80 correspondiente a HTTP, el 443, razón por la que será necesario aceptar el paso de los paquetes cuyo puerto destino sea en efecto el 443, pues de otro modo los usuarios no podrán establecer una conexión con páginas como www.gmail.com o www.hotmail.com que utilizan HTTPS.

2.6.3 RESOLUCIÓN DE NOMBRES EN INTERNET (DNS)

Este servicio es sin duda imprescindible para la interacción de un usuario con el Internet. La resolución de nombres permite que un nombre de host pueda ser mapeado a su correspondiente dirección IP. En ese sentido, es necesario configurar el firewall en el borde de la red interna de manera que re-direccione las peticiones DNS desde el interior hacia los servidores *master* en el Internet. Para este efecto nuestro firewall debe aceptar las peticiones en forma de paquetes cuyo puerto de destino es el 53 y que vayan hacia el servidor DNS. Debido a que DNS puede funcionar tanto con TCP o UDP, deberán aplicarse reglas que permitan la transmisión de peticiones DNS tanto UDP como TCP.

Dentro de este servicio se puede plantear no solamente la redirección de las peticiones DNS hacia el exterior sino también resolución de nombres para las estaciones de la red interna. Se debe, sin embargo, tener cuidado de que no se libere información al respecto a través del DNS, pues la red interna sería blanco fácil de *crackers* dado que dispondrían de datos de la estructura de direccionamiento interno de la red. Esta situación involucraría un inconveniente de seguridad en el caso de que nuestro firewall fuese también el servidor DNS para la *intranet* y para el exterior. En este caso en particular se debería configurar dos servicios DNS independientes cada uno funcionando en una sola interfaz, de manera que peticiones de resolución externas (provenientes de Internet) respecto de nuestra red interna no sean atendidas.

En realidad, lo recomendable es disponer al menos de tres servidores DNS, como se muestra en la figura siguiente. El primero debería ubicarse en la red que funge como Zona Desmilitarizada (DMZ) y debería encargarse de resolver las peticiones DNS provenientes de Internet relacionadas con nuestros *hosts* bastión (servidor de correo, servidor *web, etc.*), así como redireccionar las peticiones DNS de nuestra red interna respecto de hosts ubicados en Internet. El segundo servidor

DNS sería el encargado de resolver las peticiones provenientes de la red interna acerca de las estaciones que la conforman, poniendo especial énfasis en el hecho de que solamente deberían resolver las peticiones provenientes de la *intranet*. Un tercer servidor DNS serviría de respaldo en caso de que el anterior fallara.

Así pues, los *routers (firewall)* que delimiten la DMZ deberán configurarse de acuerdo a los requerimientos de tráfico de cada uno de los servidores DNS y tomando en cuenta los riesgos que implica la posibilidad latente de que agentes externos puedan resolver nuestra infraestructura interna de red si el *firewall* está mal estructurado.

Es muy común en nuestro entorno que empresas pequeñas concentren todos sus servicios de acceso a Internet (servidor de correo electrónico, servidor web, servidor DNS, firewall, etc.) de manera muy centralizada (en un solo *host*), por el costo que implica la adquisición de los equipos necesarios para distribuir estos servicios.

En el anexo D se puede apreciar el procedimiento para la configuración de un servidor DNS en la plataforma operativa de Linux.

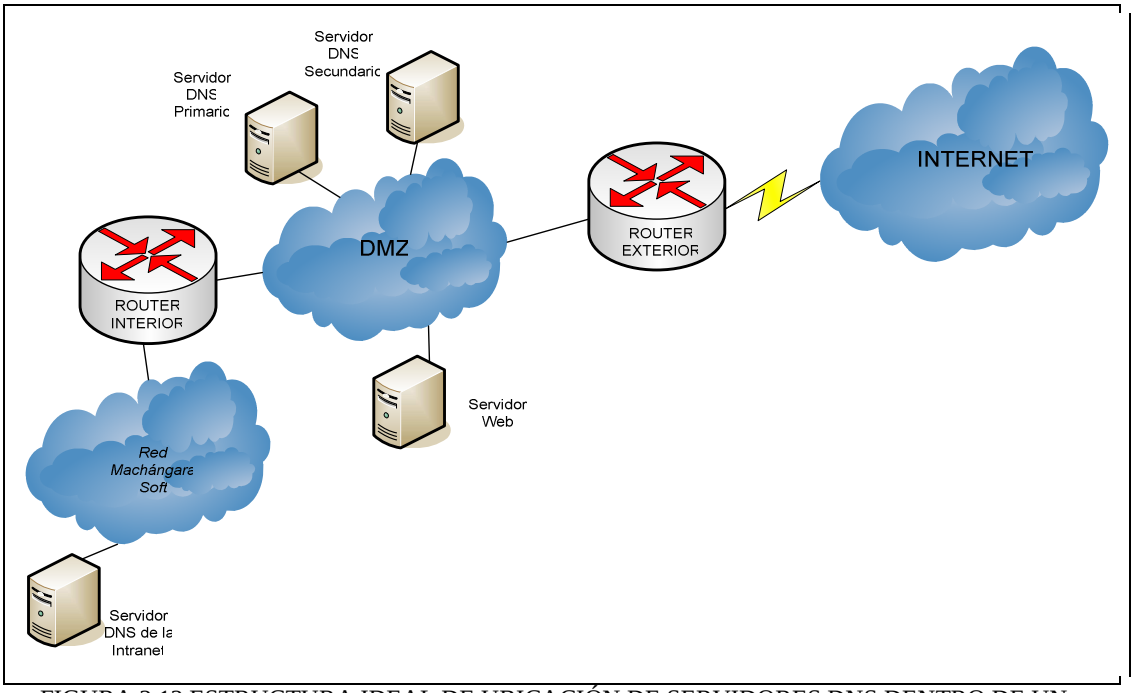


FIGURA 2.12 ESTRUCTURA IDEAL DE UBICACIÓN DE SERVIDORES DNS DENTRO DE UN SISTEMA DE FIREWALL.

2.6.4 ACCESO REMOTO (SSH, TELNET)

Como se mencionó en el primer capítulo, existe una gran variedad de protocolos y aplicaciones que permiten el acceso remoto, muy importante en el caso de mantenimiento de servidores. Lamentablemente, muchos de esos métodos de *login* remoto están asociados con problemas de seguridad, siendo el más grave de ellos la transmisión de información (incluyendo las contraseñas) en texto sin encriptar, por lo que estos datos son visibles para cualquiera que tenga acceso a una de nuestras interfaces y utilice simplemente un *sniffer*.

Entre estos protocolos se puede mencionar a *telnet.* Un protocolo de capa de aplicación muy común para acceso remoto pero sin cifrado de la información, así como otros métodos comunes en ambientes Unix como *rlogin* y *rsh*.

Así surge SSH como un protocolo que permite el acceso y ejecución de comandos de forma remota, que utiliza un nivel de cifrado elevado de la información, así como un método de autenticación basado en algoritmos de clave pública. Con todo esto, la lectura de la información privada será infructuosa ya que estará encriptada.

El puerto que emplea el servicio SSH para aceptar conexiones es el 22.

Como es comprensible, el acceso remoto a nuestro *firewall* es necesario para darle mantenimiento, especialmente cuando el técnico encargado o el administrador no están físicamente cerca del cuarto de comunicaciones o del servidor. En esa línea, se debe permitir todos los paquetes que intenten ingresar (INPUT) en nuestro *firewall* provenientes de cualquier dirección IP válida hacia el puerto 22 correspondiente a SSH.

En el caso de que existiera otro tipo de servicios en nuestra red interna (como correo electrónico, web, ftp) que deban ser visibles desde Internet, se deberá permitir también los paquetes para el acceso a ellos desde el exterior por parte de los administradores, pero únicamente a estos *hosts* que de preferencia deberán ubicarse dentro de la DMZ.

2.6.5 CORREO ELECTRÓNICO (SMTP, POP, IMAP)

No cabe duda de que el uso del servicio de correo electrónico se ha generalizado a nivel mundial al punto de haberse convertido en una herramienta fundamental

para las empresas y sus empleados. En el caso del parque tecnológico MachángaraSoft la situación no es diferente, pues cada una de las empresas utilizan el servicio de *e-mail,* como un mecanismo de comunicación esencial y muy económico. El *firewall*, por tanto, deberá permitir el paso de los paquetes correspondientes. Aún cuando, por el momento, no se disponga de un servidor de correo interno, se tomará en cuenta que existen muchos usuarios de clientes de descarga de correo POP, como *Thunderbird* u *Outlook* que utilizan puertos determinados (como el 109 y el 110) para recuperar su correo de servidores en Internet a través del Protocolo POP (*Post Office Protocol*). Este acceso será posible, permitiendo que atraviesen (FORWARD) el firewall aquellos paquetes cuyo origen sea la red interna desde un puerto aleatorio y cuyo destino sea cualquier *host* en el puerto destino 109 y 110 TCP. Algunos otros clientes de correo son utilizados también, como IMAP que utiliza otros puertos ya que incluso puede funcionar sobre SSL para mayor seguridad.

2.6.6 ACCESO A SERVIDORES DE ARCHIVOS (FTP)

Muchas de las empresas que conforman la corporación poseen sitios web o servicios de correo electrónico albergados en Internet por servidores contratados de *hosting,* a los que deben acceder mediante el protocolo FTP (*File Transfer Protocol*) para actualizar o administrar la información correspondiente de las empresas.

Debido a esto será necesario que el firewall reenvíe (FORWARD) los paquetes IP cuyo puerto destino sea el correspondiente a FTP (21), para las estaciones que forman parte de la red interna del parque.

2.6.7 MENSAJERÍA INSTANTÁNEA

La visión frente a los medios de comunicación y la filosofía de la corporación al respecto se orienta a pensar que cualquier tipo de comunicación es útil siempre y cuando se le de el uso adecuado. Por este motivo la mensajería instantánea a través de aplicaciones como MSN Messenger (Windows) y GAIM (Linux) resulta

de mucha utilidad para varios miembros de las empresas por lo que el firewall deberá permitir el reenvío (FORWARD) de los paquetes correspondientes, es decir, en el caso de MSN Messenger los que se dirijan hacia el puerto 1863 de la capa de transporte.

En determinado punto en el que el uso de cualquiera de estas herramientas interfiera con el rendimiento de los empleados, se podrá restringir su uso, simplemente eliminando la regla que permite este tipo de tráfico, gracias a la política restrictiva que se decide implementar.

2.6.8 DESCARGA P2P

Debido a que varias de las empresas trabajan instalando y probando sistemas operativos y aplicaciones de software, se requiere usualmente emplear este tipo de canales de descarga, por lo que se deberá permitir la utilización de algunas aplicaciones como *Limewire*, admitiendo el reenvío de paquetes dirigidos al puerto 6346, en este caso.

Asimismo se trata de un servicio del que se puede abusar fácilmente provocando, por ejemplo, la saturación de la conexión a Internet, por lo que quizá en determinado momento será necesario retirar esta regla que permite la utilización de descarga P2P.

2.6.9 VPN (Virtual Private Network-Red Privada Virtual)

Tal como se mencionó con anterioridad, las VPNs son infraestructuras de red que permiten interconectar dos redes privadas o *intranets* a través de otra red de tipo público como el Internet. La característica fundamental de las VPNs es que garantizan la privacidad de los datos gracias a mecanismos de cifrado elevado que evitan que la información que se transmite sea accesible a terceros.

Dependiendo de los protocolos de autenticación que se decida implementar, será necesario permitir la entrada y salida (INPUT, OUTPUT) del *firewall* de paquetes cuyo puerto de destino sea el correspondiente a estos protocolos.

2.6.10 CVS (CURRENT VERSION SYSTEM)

CVS es un sistema de mantenimiento de código fuente muy útil para grupos de desarrolladores que trabajan de forma cooperativa utilizando algún tipo de red.

Un par de empresas en el parque utilizan para el desarrollo de software este sistema que emplea el puerto 2401 para la comunicación cliente-servidor.

El firewall tendrá que permitir el reenvío (FORWARD) de paquetes cuyo puerto de destino sea el mencionado anteriormente de manera que los miembros del parque puedan acceder a este servicio.

2.6.11 VoIP

En el caso de las comunicaciones de VoIP, el flujo de paquetes depende fundamentalmente del protocolo de comunicaciones que se emplee, en primera instancia, para transmitir las comunicaciones de voz por Internet y del puerto que se configure en el servidor de voz para recibir las peticiones de registro por parte de los clientes de telefonía IP.

En el caso de que la central de telefonía IP esté ubicada en la red interna, el firewall deberá enrutar las peticiones o paquetes del exterior cuyo puerto destino sea el correspondiente al que escucha nuestro servidor y además permitir su paso a través del dispositivo de seguridad mediante la regla correspondiente.

2.6.12 OTROS

No cabe la menor duda de que al tiempo que vayan surgiendo nuevas necesidades, nuevos servicios deberán implementarse y en esa línea, el administrador de la red deberá habilitar los puertos de comunicaciones correspondientes para que estos servicios sean accesibles. Está claro que se trata de una labor continua y que involucra el contacto con los usuarios de la red de manera que no se permita otro tráfico más que el estrictamente necesario.

2.7 DIMENSIONAMIENTO DE LA CONEXIÓN DE DATOS

Una vez determinados los requerimientos de conexión de la corporación, una labor inevitable, especialmente cuando se está proyectando tráfico en tiempo real

como el de VoIP, es la de dimensionar la conexión de datos, esencialmente hacia Internet.

2.7.1 CONSIDERACIONES INICIALES

De acuerdo a los requerimientos de conexión de la corporación y a los servicios que alojaría su *intranet*, se puede diferenciar claramente entre dos tipos de tráfico que se generará: el de voz y el de datos.

El tráfico de voz es más crítico pues tiene que transmitirse en tiempo real, de modo que un mecanismo que brinde calidad de servicio sería muy importante si las aplicaciones de voz a través de Internet empiezan a utilizarse de forma común.

El tráfico de voz es especialmente complicado de manejar a través de Internet por la baja capacidad de ancho de banda de que se suele disponer.

Este tipo de tráfico se generará básicamente cuando miembros de la corporación decidan hacer llamadas internacionales a través de un proveedor de VoIP para ahorrar costos de comunicación. Del mismo modo, es posible permitir a clientes internacionales o locales, comunicarse con el parque tecnológico a través de Internet, con el inherente ahorro que eso significaría para ellos al utilizar una conexión de banda ancha disponible a Internet.

El tráfico de datos incluye esencialmente el flujo de información generado por las peticiones *web* de los usuarios de la *intranet* ("navegación"-*downloading*), acceso a los servicios internos (web, ftp, mail) de la corporación desde Internet.

El número de usuarios de la LAN es de 25; sin embargo, el número total de miembros va creciendo y, aún cuando no sean parte de la infraestructura física, forman también la comunidad de *MachángaraSoft*. En este sentido, el número total de miembros está entre los 40, que sería el número de usuarios del servidor de correo electrónico.

2.7.2 CÁLCULO DEL TRÁFICO DE VOZ

No se tienen estadísticas del tráfico de voz a través de Internet puesto que al no existir las herramientas adecuadas para su manejo no se ha hecho común "llamar por teléfono a través de Internet", sin embargo como parámetro inicial se puede

tomar el cálculo realizado para las troncales analógicas a nivel local. El tráfico de voz local será definitivamente mayor que el internacional que podría cursarse a través de Internet, en ese sentido se podría a asumir que tendremos la mitad de troncales de voz en Internet en comparación con las troncales locales a través de Andinatel. Dado que el cálculo refirió 3 troncales analógicas, para el tráfico de voz a través de Internet no se necesitarán más de dos canales de voz.

Se pretende utilizar el estándar de codificación de voz G.711, que si bien es cierto ocupa un gran ancho de banda pues casi no involucra compresión de la voz, su manejo es soportado por gran cantidad de dispositivos cliente y por la mayoría de proveedores Quizá lo más importante, no requiere un elevado procesamiento por parte del servidor por el hecho mismo de que casi no involucra compresión y por ello mismo no hay retardos en la comunicación si se dispone del ancho de banda adecuado. Este códec es utilizado por las comunicaciones a través de la PSTN y goza de una excelente calidad. Se mencionaron inicialmente otros códecs que combinan eficientemente compresión, calidad y velocidad de transmisión pero que lamentablemente no están estandarizados. Posteriormente cuando lo estén se podrá aprovechar de mejor manera el ancho de banda que ahora se reserva para la transmisión a través de G.711.

El ancho de banda que ocuparía por canal es de 84 kbps incluida la cabecera TCP/IP[22], por canal de voz. Esto significa que debería disponerse de al menos **168 kbps** para el tráfico de voz (2 canales * 84 kbps/canal).

2.7.3 CÁLCULO DEL TRÁFICO DE DATOS

Es una tarea muy interesante analizar el tipo de información que cada aplicación de red genera, de modo que pueda determinarse el ancho de banda que se consume por ello. Esta labor es necesaria de manera que puedan establecerse criterios que, en caso de que las circunstancias de tráfico cambien, estos puedan ser corregidos.

[22] http://www.voip-info.org/wiki/view/ITU+G.711

2.7.3.1 Tráfico del servidor de correo electrónico

Debido a que se trata de correo corporativo, el tamaño promedio de los mensajes recibidos y enviados puede asumirse como 50 KB, así como el tiempo máximo de retardo entre el envío y recepción del mismo se supondrá en 30 segundos. Finalmente se considera un nivel de simultaneidad del 10%, que señala que en el pero de los casos el 15% de los usuarios de correo corporativo estará enviando/recibiendo un mensaje en el servidor.

El cálculo del ancho de banda consumido en este proceso luciría así:

$$Vcorreo/usuario = 50[KB] * \frac{8[bits]}{1[byte]} * \frac{1}{30[s]} = 13.333[kbps]$$
$$Vcorreo = 25(0.15) * 13.333[kbps] = 40[kbps]$$

2.7.3.2 Tráfico del servidor web

a diferencia del tráfico de correo electrónico, el flujo al servidor web se apreciará como flujo de descarga de información de éste y por tanto como *upload* desde la *intranet*. Se asume que en el peor de los casos habrá 5 accesos simultáneos al servidor, que el tamaño promedio de las páginas que forman el *website* es de 20 KB y que el tiempo máximo en que deberían cargarse es de 10 segundos, se tiene que la velocidad de *uploading* consumida sería:

$$Vweb/usuario = 20[KB] * \frac{8[bits]}{1[byte]} * \frac{1}{10[s]} = 16[kbps]$$
$$Vweb = 5 * 16[kbps] = 80[kbps]$$

2.7.3.3 Tráfico de navegación

El tráfico de navegación originado en la *intranet* si involucra descarga de información y por lo tanto flujo de *downloading.* Se supone una simultaneidad de 20%, que asume que, en el peor de los casos, el 30% de los usuarios estarán accediendo simultáneamente a una página web de tamaño promedio 20 KB y que se cargará máximo en 10 segundos. El cálculo de tráfico sería:

$$Vnavegación/usuario = 20[KB] * \frac{8[bits]}{1[byte]} * \frac{1}{10[s]} = 16[kbps]$$
$$Vnavegación = 25(0.3)(16[kbps]) = 120[kbps]$$

2.7.3.4 Tráfico FTP

El tráfico hacia el servidor FTP presupone también un flujo de *uploading* desde el punto de vista de la *intranet*. El servidor se configurará de manera que puedan acceder máximo 3 usuarios simultáneamente y a cada uno de ellos se asignará un ancho de banda de 5 kbps de acceso al servidor. El cálculo determinará que en el peor de los casos se tendría un ancho de banda consumido de **15 kbps**.

2.7.3.5 Tráfico vía VPN

El momento en que una VPN enlace sucursales de la corporación MachángaraSoft, un cálculo pertinente sobre el ancho de banda que consumirán las aplicaciones que funcionen a través de los túneles deberá plantearse del mismo modo que se lo ha hecho para otros tipos de tráfico. El tráfico que podría cursarse es tráfico de monitoreo y acceso remoto, acceso a bases de datos, etc.

2.7.4 CONSIDERACIONES FINALES

Sin duda alguna este resulta un cálculo por demás presuntivo, pero que puede ajustarse a la realidad en la medida en que los criterios de simultaneidad se vayan adaptando a lo que efectivamente sucede.

Los cálculos anteriores determinan los siguientes resultados:

Tráfico de Voz = 168 [kbps]

***Tráfico de Datos (*download*)=* Vnavegación + Vcorreo = 160 [kbps]**

Tráfico Total (download) = **328 [kbps]**

A simple vista, la velocidad de conexión actual (400/150) que posee la corporación sería capaz de satisfacer los requerimientos de acceso a Internet. Sin embargo, se debe anotar que está capacidad de acceso es compartida, de manera que no se podrá disponer de todo el canal en cualquier instante de tiempo. Esto resulta ser un gran inconveniente, incluso en la actualidad, que el parque no utiliza aún aplicaciones de voz.

Es, por tanto, imperativo la contratación de un canal dedicado de **512 [kbps]** que garantice un valor cercano a ese de acceso a Internet para que pueda asegurarse

el desempeño adecuado de las aplicaciones de voz y datos mencionadas, en función de los cálculos realizados.

2.8 POLÍTICAS DE SEGURIDAD A APLICARSE

2.8.1 DEFINICIÓN DEL PROBLEMA

Si se analiza detenidamente el entorno que se ha descrito en párrafos anteriores, se puede observar que existen muchas debilidades que presenta la red interna de la corporación frente a su conexión a Internet. Debido al número reducido de miembros que todavía conforman el parque y a la inexistencia de delincuentes informáticos con experiencia, los efectos de una conexión vulnerable no se hacen palpables de inmediato, pero poco a poco, conforme el desarrollo tecnológico vaya involucrando a la mayoría de la población, estas vulnerabilidades se irán asociando con ataques cada vez más especializados, que requerirán de la aplicación de las políticas de seguridad que ahora se plantean.

- Para empezar, debido a que las empresas son todavía pequeñas, no requieren, por ejemplo, albergar internamente servicios de *hosting* de sitios web o de correo electrónico. Sin embargo, en poco tiempo estos servicios serán necesarios y se deberá tomar las medidas al respecto para proteger no solamente el acceso a esos servicios sino también la información que a través de ellos se maneja, en el sentido de que varios de ellos requerirán ser accesibles desde cualquier parte del Internet, lo que implica permitir explícitamente el tráfico externo hacia la *intranet* de la corporación.
- Básicamente no existe ninguna política definida con respecto a la navegación HTTP, por lo que hay una completa libertad para acceder a sitios de cualquier tipo de contenido, sea de descarga, *Chat*, pornografía, entretenimiento, etc. Esto no necesariamente es conveniente para el desempeño adecuado de los miembros de cada empresa, por lo que se deberá tomar este aspecto en cuenta al momento de generar listas de control de acceso a nivel de capa de aplicación.

- En lo referente al dispositivo que permite el enrutamiento hacia Internet, está configurado únicamente con las funcionalidades por defecto, lo cual resulta riesgoso especialmente en cuanto a las reglas de filtrado de tráfico.

Con el objetivo de permitir un nivel adecuado de protección de la *intranet* de la corporación, que involucre un costo reducido y brinde además un rendimiento satisfactorio, se plantea el diseño de un sistema de seguridad basado en software de libre distribución (Linux-*iptables*) que permita el filtrado de tráfico a través del bloqueo de todos los paquetes cuyo tránsito no esté explícitamente permitido. Este sistema estará conformado por una estación de trabajo corriendo sobre una plataforma operativa en base a Linux con las aplicaciones pertinentes que permitan el enrutamiento de los paquetes desde la interfaz de conexión a Internet hacia la interfaz interna, así como las correspondientes para el filtrado del tráfico.
Este diseño se concentrará en el establecimiento de las reglas para el manejo del tráfico antes comentado, a través de la herramienta de *netfilter/iptables* (de Linux) que se describió en el capítulo precedente. Esta herramienta ofrece un nivel elevado de control de los paquetes en cuanto a capa de enlace de datos, de red y de transporte.
A nivel de capa de aplicación se trabajará con listas de control de acceso, funcionalidad propia de la herramienta de proxificación *Squid.*

2.8.2 POLÍTICA DE SEGURIDAD DEL PARQUE TECNOLÓGICO

Así como se mencionó en el primer capítulo, existen varios aspectos que se pueden considerar dentro de una política de seguridad: la seguridad física, seguridad de red, autenticación, autorización, pero el presente diseño se concentrará en la seguridad a nivel de red basándose, como ya se mencionó, en las herramientas del sistema operativo Linux para el filtrado de tráfico.
Una política de seguridad se define, entre otras cosas, como un conjunto de requisitos que establecen lo que está y lo que no está permitido dentro de la operación de un sistema sobre el que se aplican estas políticas. De alguna forma, al determinar las aplicaciones que se requieren dentro de la Corporación

MachángaraSoft, se establece los servicios que están permitidos dentro de la operación normal de la red.

La política *restrictiva* que se plantea se caracteriza por denegar todas las conexiones (o paquetes relacionados) que no estén *explícitamente permitidas*, o que descarta cualquier paquete de comunicación relacionado con un servicio o aplicación que no concuerde estrictamente con los requerimientos explícitos de conexión de la corporación. Esto se resume en denegar todo el tráfico **por defecto** e ir abriendo paso, uno por uno, a cada servicio que forme parte de los requerimientos.

Ahora que se ha definido el mecanismo que se utilizará para definir las reglas, mediante una política restrictiva, se concentrará el diseño alrededor de esta política, estructurando las tablas de tráfico de manera que se vayan habilitando únicamente los servicios requeridos y ningún otro más.

2.8.3 TOPOLOGÍA DEL SISTEMA DE FIREWALL

Dentro de las distintas topologías analizadas anteriormente en cuanto a la estructura que puede conformarse para un sistema de seguridad en base a *firewall*, la que se empleará y quizá la más utilizada actualmente en nuestro ámbito por pequeñas y medianas empresas es la topología *Screened Host,* que consiste en un único punto de falla en el borde de la conexión a Internet, entre la *intranet* y una infraestructura de acceso público. Este punto lo constituirá una estación en la que se instalarán todas las herramientas que permitan el enrutamiento, enmascaramiento, filtrado, control y monitoreo de tráfico desde el interior de la corporación hacia el Internet y viceversa. Es verdad que no es el diseño más seguro que se puede ofrecer a la red interna, pero la adquisición de más equipos para formar una estructura de protección más sólida y la mayor inversión que esto significa, limitan enormemente las posibilidades de implementar por ejemplo una topología de tipo *Screened Subnet*, que permite incrementar los puntos de falla, haciendo mucho más compleja la labor de *crackers* y demás. Esta “cultura del ahorro” es muy común incluso en el momento de implementar servicios de correo, de alojamiento web, de resolución de nombres de dominio, a los que las empresas, por evitar el gasto económico,

concentran en un único servidor, que en el caso de fallar, se comprometería todas las comunicaciones de la empresa.

A breves rasgos, la topología en la que se basará nuestro sistema de *firewall* es la mostrada en la figura 2.13.

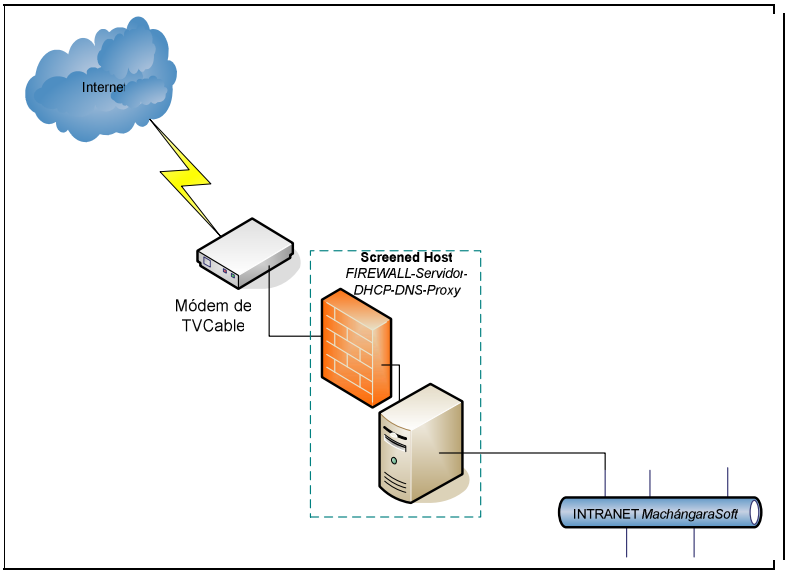


FIGURA 2.13 TOPOLOGÍA DE LA ESTRUCTURA DE FIREWALL DE LA CORPORACIÓN.

2.8.4. ESTABLECIMIENTO DE REGLAS DE TRÁFICO PARA LA CORPORACIÓN MACHÁNGARASOFT

En función de los requerimientos de tráfico anotados de la corporación y las herramientas estudiadas para el enrutamiento, enmascaramiento y filtrado IP, es posible establecer algunas de las reglas de tráfico que regirán nuestra configuración del *firewall* sobre Linux.

Es importante acotar nuevamente que los criterios de filtrado son variados: dirección IP origen, destino, puerto origen o destino, interfaz de entrada, dirección del tráfico, estado de la conexión, límite de coincidencia (match) por regla, etc. Asimismo corresponde aclarar que generalmente los paquetes TCP/UDP/IP que inician las conexiones llevan puertos origen aleatorios, comúnmente mayores a 1024 y que el puerto destino (comúnmente menor a 1024) es el que determinará el tipo de servicio al que accede un paquete.

Las reglas de enmascaramiento y filtrado de tráfico que se emplearán pueden resumirse como sigue.

- Para que *iptables* goce de algunas funcionalidades especiales deben cargarse algunos módulos al *kernel* de Linux. Esto se logra mediante el comando **modprobe**.
- Antes de iniciar cualquier configuración siempre será conveniente *borrar cualquier detalle anterior por defecto o antiguo de reglas de tráfico*, de manera que se inicie el proceso desde “cero”.
- Dentro del *kernel* de Linux se incluyen varias opciones que activan la función de enrutamiento y otras opciones que permiten asegurar el sistema de ataques muy conocidos como *spoofing* de direcciones IP y en general, varios tipos de ataques de DoS.
- Si existen intentos de vulnerar el *firewall* a través de servicios cuyo tráfico no está permitido, *iptables* cumplirá su cometido y descartará esos paquetes, sin embargo, no se tendrá información específica sobre los puertos que están siendo atacados. Por esta razón se debe filtrar tráfico característico de ataques, que generalmente van dirigidos a puertos ya identificados. De este modo, si un ataque conocido está ejecutándose, se podrá registrarlo y tomar medidas adicionales si es necesario. Estas cadenas permiten controlar también el número de conexiones que se pueden establecer en un intervalo de tiempo, cosa que no se puede hacer únicamente con las políticas por defecto.
- Para protección frente a ataques de inundación con paquetes *syn (Syn Flooding)* se genera una cadena que compruebe si el paquete pertenece a este tipo de ataque antes de aceptarlo,
- Asimismo es conveniente registrar los intentos de acceso a través de ciertos puertos *especiales* típicos de ataques *backdoor.* Estos ataques intentan establecer conexiones remotas ilícitas utilizando estos puertos conocidos. Entre los puertos se encuentran algunos característicos del sistema operativo *Windows* que se abren por defecto especialmente para compartir archivos.
- Todos los paquetes que contengan combinaciones de banderas inválidas serán descartados. Estas combinaciones son utilizadas por escáneres de puertos e incluyen por ejemplo todas las banderas activadas o ninguna para interpretar las respuestas ante estos paquetes erróneos.

- Se crearán cadenas para **filtrar paquetes ICMP** que forman parte de escaneo de conexiones y ataques *fingerprinting* a través de los que se puede determinar algunas características del sistema operativo de los *hosts* sujetos del proceso de escaneado. Estas cadenas protegerán también el sistema de seguridad de ataques *Ping Flooding* (inundación de *ping*), limitando la cantidad de paquetes que pueden atenderse en un tiempo dado.
- Se especificarán reglas de tráfico para las cadenas predefinidas INPUT, OUTPUT y FORWARD de la forma en que se describe a continuación:

2.8.4.1 Cadena INPUT (Todos los paquetes que se dirigen al *firewall-box*)

- Se aceptarán todos los paquetes entrantes en el *firewall-box* pertenecientes a conexiones ya establecidas o relacionadas.
- Se determinará si cualquier paquete entrante es inválido o tiene combinaciones de banderas erróneas para registrarlo y descartarlo.
- Se aceptarán todos los paquetes que ingresan al sistema de *firewall* a través de la interfaz virtual de *loopback* y se rechazarán todos aquellos que intenten acceder por otra interfaz hacia la red de *loopback* 127.0.0.0/8.
- Entre los paquetes que se originan en la red interna, los que se permitirá dirigirse hacia el *firewall-box* son: los paquetes ICMP válidos, las peticiones DNS (puerto 53, ya que el dispositivo también redirige estas peticiones) y las peticiones al servicio de proxy al puerto 8080. Se tomará en cuenta que estos paquetes deberán ingresar por la interfaz interna originándose en cualquier máquina de la *intranet*. Se registrará y descartará cualquier intento de ingreso de paquetes que no cumplan con estas condiciones, especialmente aquellos que pretendan ingresar por la interfaz externa con direcciones-origen falsificadas.
- Se filtrará el tráfico proveniente de la red externa que se dirige al *firewall*, permitiendo los paquetes ICMP válidos y descartando cualquier tráfico SMB sin registrarlo, además de evitar el ingreso de cualquier flujo perteneciente a *traceroute.*

- El tráfico a través de SSH (puerto 22) también se permitirá para el acceso remoto al dispositivo de *firewall*, desde el Internet a la interfaz externa y desde la *intranet* mediante la interfaz interna.
- Se rechazará (REJECT) con un mensaje TCP de RESET (RST) el tráfico correspondiente al protocolo ***Ident*** [23], para evitar los excesivos retardos en conexiones FTP, HTTP, POP, etc., que utilizan Ident para autenticación.
- Se filtrará cualquier flujo de paquetes a través de puertos típicos de escaneo y mediante los que se establecen ataques ya conocidos para acceso remoto ilícito.
- Se permitirá el tráfico a través de las interfaces virtuales que forman parte de una posible infraestructura de VPN. En el caso de que el *firewall-box* funcione como servidor, deberá aceptar las peticiones al puerto especificado por la herramienta que se utilice.
- Todos los paquetes restantes serán registrados y descartados.

2.8.4.2 Cadena OUTPUT

- Esta cadena estará conformada por todos los paquetes que se originen en el *firewall-box* y salgan por tanto de éste.
- Se permitirá el paso de paquetes que se originen en el dispositivo de *firewall* y salgan a través de la interfaz de *loopback.*
- Se permitirá la salida de cualquier paquete correspondiente a conexiones ya establecidas o relacionadas con estas últimas.
- Se aceptará el paso de todos los paquetes que se dirijan hacia la *intranet,* a través de la interfaz interna correspondiente.
- Se permitirá el tráfico ICMP válido de salida a través de la interfaz externa, descartando en cambio los paquetes del tipo SMB[24] y evitando registrarlos.

---

[23] Este protocolo es a veces utilizado por servidores de Telnet, POP, FTP y HTTP para identificar usuarios entrantes. Cuando un usuario solicita un servicio, el servicio intenta establecer una conexión de regreso al cliente que está tras un firewall para identificar el nombre de usuario del proceso que inició la conexión. El *firewall* intercepta esta conexión y la descarta silenciosamente (DROP). Esto implica que el servidor nunca reciba la respuesta esperada y podría no permitir al usuario conectarse. La mayoría de usuarios considera a IDENT como una violación de seguridad pues puede permitir a alguien en el exterior ganar acceso a información confidencial de una red privada. Los problemas de conexión relacionados con los servicios mencionados pueden solucionarse al rechazar explícitamente los paquetes de conexión IDENT enviando mensajes RST (reset) hacia el servidor para que las conexiones originales puedan continuar inmediatamente.

- Se permitirá el tráfico de todos los paquetes salientes TCP/UDP que abandonen el dispositivo de *firewall* a través de la interfaz externa con un puerto origen mayor a 1024.
- Todos los paquetes de salida restantes serán descartados y registrados.

2.8.4.3 Cadena FORWARD

Esta cadena contendrá las reglas correspondientes a aquellos paquetes que atraviesen el *firewall* desde la *intranet* hacia el Internet o viceversa.

- Se aceptarán los paquetes que atraviesen el *firewall* y pertenezcan a conexiones ya establecidas o que estén relacionados con ellas.
- Se descartarán los paquetes inválidos y con combinaciones de banderas erróneas.
- Desde la *intranet,* así como desde Internet se filtrará el tráfico SMB, descartando los paquetes correspondientes.
- Desde la *intranet* se permitirá el establecimiento de conexiones DNS, SSH, FTP, HTTPS, MSN Messenger, POP/IMAP, CVS, VPN y descarga P2P hacia el Internet, de acuerdo con los requerimientos específicos de la corporación.
- El tráfico que se filtrará desde el Internet es el SMB evitando que se registre y descartándolo si existe.
- Todo el tráfico de reenvío (FORWARD) restante será registrado y descartado.

2.8.4.4 Cadena PREROUTING

En esta cadena se enrutarán aquellos paquetes provenientes de Internet que requieran un servicio especial en la *intranet*, es decir, que sean peticiones de acceso a un servicio *web*, de correo o FTP.

En esta sección se incluirán las reglas que permiten el enrutamiento de paquetes en función del destino que tengan, permitiendo, por ejemplo la redirección de puerto necesaria para la *proxificación transparente.*

---

[24] SMB – *Server Message Block* o Bloque de Mensajes de Servidor, es un protocolo de red usado principalmente en ordenadores *Microsoft Windows*, para compartir archivos e impresoras entre nodos de una red. Los puertos de comunicaciones que se utilizan para su funcionamiento se caracterizan por ser altamente vulnerables.

2.8.4.5 Cadena POSTROUTING

Esta cadena incluirá la regla de enmascaramiento de todos los paquetes que son enrutados desde la *intranet* de modo que puedan viajar a través de Internet.

Todos los aspectos mencionados resumen brevemente las reglas de tráfico que se establecerán para el control de los paquetes que ingresarán, se originarán o atravesarán el dispositivo de *firewall*.

2.9 DISEÑO Y CONFIGURACIÓN DEL PROTOTIPO

En esta fase se plasmarán cada una de las políticas y requerimientos de conexión y seguridad de la corporación en los respectivos *scripts*, parámetros de kernel y archivos de configuración correspondientes que gobernarán los servicios de enrutamiento, enmascaramiento y *firewall*, así como otros servicios complementarios para nuestro sistema de seguridad.

2.9.1. CARACTERÍSTICAS DEL EQUIPO Y DEL DIRECCIONAMIENTO

En general, el sistema operativo Linux se caracteriza por un gran rendimiento sobre *hardware* bastante modesto. CentOS, por ejemplo funciona con procesadores desde *Pentium* II pero siempre es una buena costumbre sobredimensionar los equipos para mejorar en disponibilidad.

El servicio que fundamentalmente estará funcionando en este equipo será el de *firewall* y seguramente el servicio de *proxy.* Estos servicios no consumen por si mismos mucha memoria, disco duro o procesador, por lo que hay casos en los que una máquina de 64-128 MB de RAM con procesador de 700 MHz funciona perfectamente para un medio centenar de usuarios. A estos servicios podría agregarse el de DHCP, control de ancho de banda y, aunque no es recomendable en el borde de la red, un sistema de DNS. Aún con ellos, el servidor funcionaría de manera relativamente estable. Sin embargo, aún cuando suceda todo lo descrito, a menos que se disponga de una máquina relativamente antigua con esas características, será más conveniente adquirir un modelo moderno cuyo costo es bastante reducido en comparación a las capacidades de procesamiento y memoria que actualmente están disponibles.

Otro requisito indispensable del *hardware* será desde luego el que posea al menos dos interfaces de red para que se pueda conectar en una de ellas el enlace WAN hacia el ISP o Internet y en la otra la red interna de la corporación.

En ese sentido será sencillo adquirir un CPU con las siguientes características:

- Procesador: Intel Pentium IV 2.8 GHz
- Disco Duro: 80GB
- RAM: 512MB
- 2 tarjetas de red 10/100/1000 Mbps

Cabe resaltar que estás características serán más que suficientes si trabajan los servicios antes mencionados y a menos que el número de usuarios crezca a varias centenas, el equipo no deberá presentar inconvenientes.

El direccionamiento interno se estructurará utilizando la red 10.0.0.0 con máscara de red 255.255.255.0. La dirección IP de la interfaz interna será 10.0.0.1 y la dirección IP de la interfaz externa se asignará dinámicamente por parte del ISP a través del módem.

2.9.2. LA DISTRIBUCIÓN DEL SISTEMA OPERATIVO LINUX

En realidad el núcleo (*kernel*) de cualquier distribución de Linux es esencialmente el mismo, siempre y cuando se trate de la misma versión. Ahora, existe una diversidad muy amplia de 'sabores' de Linux, distribuciones o 'distros' cuya diferencia radica en las herramientas de configuración y los sistemas de paquetes de software que se instalan.

Para el sistema de seguridad en base a *firewall* para la corporación MachángaraSoft se utilizará la distribución **CentOS 4.3** (*Community Enterprise Operating System*) pues es un clon binario sumamente confiable de *Red Hat Enterprise Linux 4 update 3*, quizá la versión corporativa más estable del núcleo Linux.

Además existen muchos proyectos específicos de software que se basan en *CentOs*, tales como: *Asterisk@Home* y *Trixbox* para telefonía IP, *Rocks v4.1* para clústers, *BlueQuartz* para administración de cyber-cafés, etc.

Es común de esta distribución, así como de *Red Hat,* la herramienta de administración de paquetes ***rpm*** (*Red Hat Package Manager*) que permite instalar, actualizar, desinstalar y consultar características de paquetes y programas. Además es muy útil la herramienta de gestión de paquetes RPM *yum* (*Yellow Dog Updater, Modified*) que permite la , actualización y consulta de versiones de paquetes RPM organizados en un repositorio interno o en repositorios de software conocidos en Internet.

Todas estas herramientas reducen en gran medida los problemas de dependencias en el momento de instalar software en el sistema operativo.

2.9.3. INSTALACIÓN SEGURA DE LA PLATAFORMA OPERATIVA LINUX PARA EL SISTEMA DE *FIREWALL*

Es necesario que la plataforma operativa de Linux se instale de tal manera que el sistema instalado sea lo suficientemente seguro y disponga de las características suficientes para que pueda ser configurado como un *firewall*.

La instalación de Linux en un *firewall* se la realiza como cualquier otro servidor, a través de un medio óptico como un CD-ROM o mediante una instalación vía red.

Es recomendable que se establezcan particiones separadas para cada sistema de archivos diferente en el *firewall*. Esto ayudará a contener los efectos de un intento de acceso a nuestro sistema.

Cuando se instala un *firewall* es comprensible que se deseará instalar el *menor número de servicios posible*. La razón es que cada servicio que se instale puede contener un agujero de seguridad, de modo que mientras menos programas se instalen, menor será la posibilidad de que algo salga mal.

Otra consideración importante al momento de instalar el sistema operativo es la asignación de una contraseña de administrador segura o lo que se conoce como el *password de root*. Esto es verdad en cada una de las estaciones de nuestra red, y especialmente en el *firewall*. Esto se relaja un poco en ciertos entornos pues debido a la dificultad de recordad diferentes contraseñas, se opta por utilizar *passwords* similares o incluso el mismo en todas las máquinas.

Está claro que el *firewall* debe ser la máquina más segura de la organización y, en ese sentido, un muy buen *password* es tan sólo la base de su seguridad.

2.9.4. APLICACIÓN DE PARCHES

En una etapa posterior a la instalación del sistema operativo es necesario visitar la página web de quienes desarrollaron la distribución y descargar todos los paquetes de actualización para la versión respectiva. Se debe asegurar la descarga de parches únicamente provenientes de quien desarrolló la versión de sistema operativo pues aquellos distribuidos por terceros podrían contener troyanos.

En el caso de *CentOS* y otras distribuciones como *Red Hat* y *Fedora*, los paquetes son distribuidos en forma de RPMs. Para instalarlos, podemos usar las herramientas que se mencionaron anteriormente: **rpm** o **yum**.

El comando **rpm** puede utilizarse con los siguientes parámetros en la consola de Linux:

- `-q` Para consultar al sistema si un paquete determinado está instalado
- `-i` Para instalar un paquete *.rpm.
- `-v` Para obtener mensajes de información sobre el proceso que se realiza.
- `-h` Imprime en pantalla un conjunto de *hashs* (#) que indica el avance en el proceso de instalación o desinstalación de un paquete.
- `-e` Desinstala un paquete especificado.
- `-U` Instala la actualización de un paquete ya existente.

Siempre será adecuado descargar la última versión disponible del *kernel* de Linux e instalarla en nuestro sistema inicial para evitar cualquier hueco de seguridad que no haya sido corregido por falta de actualización.

En el caso particular de la actualización del kernel, nunca debe instalarse con el comando **`rpm -U`** sino con **`rpm -i`**, luego de lo cual se debe reiniciar el sistema y desinstalar la antigua versión de nuestro kernel mediante el comando **`rpm -e`**.

Si se está descargando los paquetes de Internet, que es la situación más común, se deberá verificar el *checksum MD5* y la firma GPG de los paquetes RPM, utilizando el siguiente proceso:

- Montar el CD correspondiente a la distro que se instaló y agregar la clave pública GPG en ese CD-ROM al *keyring* (llavero) del sistema operativo, utilizando el comando:

```
# rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Si el punto de montaje de la unidad de CD es `/mnt/cdrom`.
- Verificar el *checksum* del paquete que se desea comprobar utilizando el comando:

```
rpm -K <nombre del paquete>
```

2.9.5. RECOMPILACIÓN DEL KERNEL

Luego de la instalación de Linux, puede ser necesario recompilar el kernel. Entre las razones que puede haber para esto están:

- Es posible que se desee deshabilitar el soporte de *hardware* que no esté disponible en el sistema. Esto no sólo desperdicia espacio de disco y memoria sino que puede ser utilizado en ataques de intrusión DoS si un *bug* es encontrado en esa pieza de código.
- No todas las distribuciones poseen un kernel que esté optimizado para el procesador que se esté utilizando, de manera que recompilando el kernel para el procesador específico podría resultar en un significativo incremento en el rendimiento.
- Al recompilar el kernel se puede tener acceso a opciones que permiten optimizar el funcionamiento del sistema como *router*, lo que quiere decir que se aumenta el rendimiento para los paquetes que serán enrutados a través del sistema de *firewall*.
- Si se recompila el kernel Linux es comprensible desear mantenerlo muy pequeño y esto se logra deshabilitando los módulos que no son necesarios y que podrían contener código inseguro que podría ser utilizado por hackers.
- Al recompilar el kernel se puede activar algunas características de seguridad adicionales a las que posee por defecto. Por ejemplo podrían activarse módulos de detección de escaneo de puertos cuyo código no está integrado en el kernel por defecto.
- En el caso de que se necesite soporte adicional para hardware especial, será necesario recompilar el kernel para activar este soporte.

2.9.6. ASEGURAMIENTO DEL SISTEMA OPERATIVO

El aseguramiento del sistema operativo se refiere a modificar algunas de las configuraciones por defecto del sistema de manera que el sistema sea tan seguro como sea posible. Esta labor involucra varios aspectos.

2.9.6.1 Asegurando el BIOS

Esto significa asegurar el BIOS (*Basic Input Output System*) de la estación al cambiar su configuración de modo que un hacker que logre ganar acceso físico no pueda ingresar en el sistema o deshabilitarlo.

Esto se puede lograr mediante varias formas:

- Cualquier BIOS permite especificar el orden de arranque de los dispositivos. Se debe configurar este sistema de tal manera que sea posible arrancar únicamente desde el disco duro, pues el arranque desde un CD-ROM o un diskette podría darle a un intruso la posibilidad de arrancar el sistema en modo de rescate.
- Algunos BIOS gozan de soporte para protección anti-virus. Esto significa obtener una advertencia cuando el Registro de Arranque Maestro (MBR) ha cambiado.
- Es una buena idea deshabilitar el APM (*Advanced Power Management* – Administración Avanzada de Energía) del BIOS pues ya que se está instalando un servidor, éste siempre deberá estar encendido, por lo que no será necesario administrar el consumo de energía dado que su actividad será constante.
- Finalmente, y quizá lo más importante es establecer una contraseña de ingreso al BIOS de manera que sus opciones de configuración no sean alteradas.

2.9.6.2 Asegurando el Boot Loader

El *Boot Loader* es el siguiente paso en el proceso de arranque y debe estar muy asegurado también. Afortunadamente, los *boot loader* más comunes en Linux permiten este nivel de seguridad. Ambos, LILO (*Linux Loader*) y GRUB (*Grand Unified Boot Loader*) permiten especificar una contraseña en su archivo de configuración. Esta contraseña no se requiere cuando un arranque regular se lleva a cabo, pero se necesita antes de que un usuario pueda arrancar el sistema

pasando ciertos parámetros al kernel como los que se necesitan para arrancar en modo *single*. Este modo es especialmente peligroso pues permite el acceso al sistema como administrador sin que se requiera ninguna contraseña a menos que el *password* para el *boot loader* se haya configurado. Esto puede permitir que personal ajeno a la administración tenga acceso al sistema de archivos y cambien incluso la contraseña de administración.

Con GRUB, es necesario especificar la contraseña en el *path* **`/boot/grub/menú.lst.`** Este *boot loader* permite encriptar las contraseñas con MD5 antes de que sean almacenadas en el archivo mencionado. Esto se puede lograr con el comando **`md5crypt.`** Es necesario también tener cuidado con la seguridad de este archivo. Algunos de los pasos para configurar este *password* son los siguientes:

```
# grub
grub> md5crypt
Password: secret
Encrypted: $1$24QV1/$ecUahVmWxCDBU3k5Mzmjy/
grub> quit
# vi /boot/grub/menu.lst
default=0
timeout=10
password=$1$24QV1/$ecUahVmWxCDBU3k5Mzmjy/
...
# chmod 600 /boot/grub/menu.lst
```

2.9.6.3 Agregar/Cambiar/Borrar Cuentas de usuario

Al igual que cada servicio instalado, cada cuenta de usuario que se crea es un potencial problema de seguridad, pues cada cuenta de usuario requiere una contraseña que puede adivinarse. Por esto es recomendable tener tan pocos usuarios como sea posible en el *firewall.* Es importante echar un vistazo de las cuentas de usuario y borrarlas o deshabilitarlas si ya no son útiles.

En realidad en el sistema de *firewall* las cuentas de usuario no son necesarias del todo, pues los únicos usuarios en el sistema son sus administradores que muy probablemente utilizarán la cuenta **root** para ingresar.

Hay varias posibilidades en este sentido, algunos consideran que es conveniente dejar a los administradores ingresar al sistema como **root** directamente, pues esto evita crear más cuentas para cada administrador, lo que resulta en un problema

de seguridad si uno de ellos es poco cuidadoso en mantener la contraseña en secreto. Otros, en cambio, afirman que es adecuado crear cuentas para los administradores y que luego de acceder como usuarios accedan al sistema como *root*, pues esto implicaría que un *hacker* adivine dos *passwords* en lugar de uno solo.

Otra medida interesante que puede tomarse es cambiar la cuenta **root** a un nombre diferente, o configurar varias cuentas **root**, una para cada administrador. Esto es factible de realizar debido a que el usuario *root* recibe sus privilegios no por su nombre sino porque su identificación de usuario es cero. Por tanto, cualquier cuenta de usuario con una identificación de cero tiene privilegios de **root** aunque esa cuenta no se llame *root.* Esto nos da una perspectiva bastante interesante, pues es posible, por ejemplo, crear una cuenta para cada administrador, pero todas con identificación cero. De esta forma, cada administrador puede acceder al sistema con su propio nombre y *password* y obtener privilegios de *root* inmediatamente. La cuenta de *root* no es necesaria pero puede configurarse de tal manera que se envíe una advertencia si alguien gana acceso a esa cuenta.

2.9.6.4 Deshabilitar servicios innecesarios

Una distribución regular inicia una gran cantidad de servicios por defecto luego de la instalación. La mayoría de estos servicios no son necesarios en un *firewall* y, por eso, deben ser deshabilitados.

Algunas de las formas en las que los servicios pueden iniciarse son:

- Los servicios que no son usados usualmente se inician a través del súper demonio *xinetd*. Para deshabilitar estos servicios se debe editar el archivo `/etc/xinetd.conf` y deshabilitar todos los servicios que no sean necesarios que ahí están listados. La distribución puede también estar configurada para incluir todos los archivos en el directorio `/etc/xinetd.d`.
- Los servicios que pueden utilizarse muchas veces por segundo o aquellos que necesitan su propio demonio funcionando todo el tiempo son usualmente iniciados en un puerto directamente. Estos servicios pueden deshabilitarse

removiendo los enlaces de inicio ubicados en los directorios **`/etc/rc.d/rc3.d`** o **`/etc/rc.d/rc5.d`**.

Esto puede realizarse utilizando el comando **`chkconfig`**, Por ejemplo si se desea desactivar el demonio de impresión:

```
# chkconfig cupsd off
```

Verificar qué servicios están activos es necesario y se logra con el comando `ps –aux` que lista los procesos que están trabajando y también a través del comando `netstat –a`, que muestra todos los puertos que están abiertos.

2.9.6.5 Deshabilitar la combinación Ctrl-Alt-Delete

Es conveniente alterar el comportamiento de la combinación de teclado *Ctrl-Alt-Delete* o deshabilitarla completamente. Generalmente el efecto asociado es el reinicio de la máquina, lo cual no es nada recomendable en un servidor de *firewall* que debe estar activo las 24 horas del día. Esta combinación se deshabilita al comentar la entrada correspondiente en el archivo **`/etc/inittab`**.

2.9.6.6 Cambiar los archivos /etc/issue y /etc/issue.net

Por defecto, estos archivos muestran cierta información acerca del tipo de sistema operativo y versión antes de que un usuario haga *login* en el sistema. Es conveniente cambiar su contenido de modo que muestre una advertencia que alerte del acceso no autorizado.

2.9.6.7 Cambiar el archivo /etc/motd

`/etc/motd` es el archivo cuyo contenido se muestra luego de que un usuario ha ingresado al sistema. Puede utilizarse para advertir a usuarios no autorizados a usar este acceso de manera ilícita.

2.9.6.8 Establecer la variable de entorno $TMOUT

Es muy común que un usuario haga *login* en un sistema como *root* y luego se vaya sin hacer un *login out*. La variable del *shell* de Linux **`$TMOUT`** permite especificar el tiempo máximo (medido en segundos) que una sesión puede

permanecer en estado inactivo, luego del cual la sesión es terminada automáticamente. Para establecer el valor de esta variable, se puede agregarla al archivo **`/etc/profile`**, si se desea que tenga ese valor para todos los usuarios, o a **`$HOME/.bash_profile`** si esa variable se asignará a un usuario específico.

2.9.6.9 Asegurar el sistema de archivos

Hay algunas medidas de protección que deben tomarse en cuenta para asegurar el sistema de archivos. Cada sistema de archivos, cuando es montado, puede soportar un número de opciones que pueden hacer un poco más difícil que *crackers* pueda dañarlo. Entre estas opciones están:

- **`noexec`**.- Evita que se puedan ejecutar comandos en este sistema de archivos, aún cuando estén configurados con los permisos de ejecución.
- **`nodev`**.- Evita que algunos archivos especiales que representan a dispositivos físicos.
- **`ro`**.- El sistema de archivos es de sólo lectura.

Todas estas opciones pueden especificarse en el archivo **`/etc/fstab.`**

Un ejemplo de este archivo podría ser:

```
/dev/hda1   /boot       ext2     defaults,noexec,nosuid,nodev     2 2
/dev/hda5   /           ext2     defaults                         1 1
/dev/hda6   /usr        ext2     defaults,ro,nodev                2 2
/dev/hda11  /usr/local  ext2     defaults,ro,nodev                2 2
/dev/hda7   /tmp        ext2     defaults,nosuid,nodev            2 2
/dev/hda8   /home       ext2     defaults,nosuid,nodev            2 2
/dev/hda9   /var        ext2     defaults,noexec,nosuid,nodev     2 2
/dev/hda10  swap        swap     defaults                         0 0
/dev/hdc    /mnt/cdrom  iso9660  noauto,owner,nosuid,noexec,nodev 0 0
/dev/fd0    /mnt/floppy auto     noauto,owner,nosuid,noexec,nodev 0 0
```

FIGURA 2.14 EJEMPLO DE UN ARCHIVO fstab

2.9.7. CONFIGURACIÓN DEL *FIREWALL* PARA EL MACHANGARASOFT

La configuración del mecanismo de *firewall* a través de *iptables* se basará en la generación de un *script* cuya ejecución iniciará el servicio de filtrado y control de tráfico, o en su defecto lo detendrá o reiniciará.

Algunas de las características de este *script* serán las que se detallan a continuación:

- Reconocimiento de parámetros a los que se aplicarán las reglas, dirección IP, puerta de enlace, máscara de red, perteneciente a las interfaces (interna y externa) a las que se aplicarán las reglas.
- Configuración de parámetros de kernel y carga de módulos especiales de la plataforma operativa.
- Creación y aplicación de cadenas conformadas por las reglas de tráfico y registro (*logging*).

2.9.7.1 Verificación de Parámetros

El nombre del *script* será *iptfirewall* y tendrá un parámetro que deberá pasarse obligatoriamente y dos parámetros opcionales. El parámetro **obligatorio** indicará la acción que se desee ejecutar con el servicio de *firewall* y las posibilidades serán: `start` (iniciar), `stop` (detener), `restart` (reiniciar) o `status` (mostrar estado). Los dos parámetros **opcionales** indicarán las interfaces (interna y externa) a las que se aplicarán cada una de las reglas de *firewall*. El parámetro que identifique a la interfaz externa deberá indicarse primero y el parámetro correspondiente a la interfaz interna se deberá indicar después. Si estos no se especifican al ejecutar el *script*, la estructura del mismo ubicará valores por defecto.

La sección que se describe es la siguiente:

```
# Ubicacion en el SO del comando iptables
IPTABLES="/sbin/iptables"

# Validacion de parametros del comando
case "$1" in  # Parametro stop -> Flush completo de reglas + politicas
por defecto ACCEPT
   stop)
      echo "Desactivando el Firewall..."
      $IPTABLES –F           #Flush de todas las cadenas
      $IPTABLES -F -t mangle  #Flush de tabla mangle
      $IPTABLES -F -t nat     #Flush de tabla nat
      $IPTABLES –X           #Flush de cadenas de usuario
      $IPTABLES -X -t mangle  #Flush de cadenas tabla mangle
      $IPTABLES -X -t nat     #Flush de cadenas de usuario tabla nat
#Politica Restrictiva por defecto (Descartar todos los paquetes no
explicitamente permitidos)
      $IPTABLES -P INPUT ACCEPT
      $IPTABLES -P OUTPUT ACCEPT
      $IPTABLES -P FORWARD ACCEPT
      echo "...Desactivado"
      ;;
status) # Parametro status -> Lista contenido de tablas filter, nat y
mangle
      echo $"Tabla: filter"
```

```
      iptables --list
      echo $"Tabla: nat"
      iptables -t nat --list
      echo $"Tabla: mangle"
      iptables -t mangle --list
      ;;
restart|reload) # Parametro restart|reload -> Permite reiniciar la carga
de reglas
      $0 stop
      $0 start
      ;;
start)  # Parametro start -> inicio de configuracion y activacion del
firewall
    echo "Activando el Firewall..."
    echo ""
```

2.9.7.2 Configuración Inicial

Dentro del proceso de configuración inicial, el *script* permitirá determinar automáticamente algunos valores de variables que se emplearán posteriormente para el establecimiento de las reglas de filtrado. Entre estas variables están por ejemplo algunos rangos de puertos de comunicaciones específicos, direcciones IP de las interfaces y límite de flujo de algún tipo de paquetes. Asimismo se cargarán módulos y se activarán opciones de kernel para el funcionamiento adecuado de la estación de *firewall*.

Inicialmente se determina el valor de las interfaces por defecto de la estación, que serán útiles en el caso de que no se especifiquen al ejecutar el *script.*

```
## Interfaz Externa por Defecto
DEFAULT_EXTERNAL_INT="eth1"
## Interfaz Interna por Defecto
DEFAULT_INTERNAL_INT="eth0"
```

Se configuran tres variables especiales como se indica a continuación:

```
# Variable que identifica a todas las direcciones IP
ALL="0.0.0.0/0"
# Especificacion de puertos no privilegiados
HIGHPORTS="1024:65535"
# Especificacion de puertos para el sistema X Window (Interfaz Grafica)
XWINPORTS="6000:6063"
```

Se determinan algunas *variables de inundación* cuyos valores especificarán los límites dentro de los que se detectarán ataques de DoS típicos a través del envío progresivo de determinadas clases de paquetes. Se identifican dos variables bastante similares, la una relacionándose con LIMIT y la otra con LIMITBURST.

La primera determina el número de coincidencias (*matchs*) en promedio medidos en un intervalo de tiempo, mientras que la segunda determina el número de coincidencias simultáneas.

```
# Estas variables representan limites en el flujo de paquetes
# Limite para deteccion de Inundacion de paquetes TCP-SYN
TCPSYNLIMIT="5/s"
# Limite de desborde para deteccion de Inundacion de paquetes TCP-SYN
TCPSYNLIMITBURST="10"
# Limite de Logging en las Cadenas de Log
LOGGINGLIMIT="2/s"
# Limite de desborde para las Cadenas de Log
LOGGINGLIMITBURST="10"
# Limite para la Deteccion de Inundacion de Ping
PINGLIMIT="5/s"
# Limite de Desborde para la deteccion de Inundacion de Ping
PINGLIMITBURST="10"
```

En la sección resaltada, por ejemplo se determinará en el primer caso que el límite de *pings* será de 5 por segundo **en promedio**, mientras que en segunda instancia, el límite de *pings* en una sola ráfaga será de 10.

La detección de información respecto de las interfaces se hará de forma individual para cada una de ellas, mediante un proceso equivalente utilizando ciertos comandos útiles del lenguaje *shell* de Linux y algunas herramientas especiales.
Inicialmente se determina si se ingresó un valor que indique la interfaz (eth0, eth1, ppp0), si no es así le asigna el valor por defecto. Luego de ello, se obtiene y se muestra los valores de dirección IP, puerta de enlace y máscara de red de cada interfaz, si no se hubieren asignado, se imprime un mensaje de error.

```
if [ "x$2" != "x" ]; then
   EXTERNAL_INT=$2
else
   EXTERNAL_INT=$DEFAULT_EXTERNAL_INT
fi
echo Interfaz Externa: $EXTERNAL_INT

## Determinar direccion IP de la Interfaz Externa
EXTERNAL_IP="`ifconfig $EXTERNAL_INT | grep inet | cut -d : -f 2 | cut -d
\  -f 1`"
  if [ "$EXTERNAL_IP" = '' ]; then
     echo "Error:Imposible determinar la direccion IP de $EXTERNAL_INT !"
     exit 1
  fi
echo IP Externa: $EXTERNAL_IP
## Determinar direccion IP del gateway externo
EXTERNAL_GW=`route -n | grep -A 4 UG | awk '{ print $2}'`
echo Default Gateway: $EXTERNAL_GW
```

Es necesario cargar algunos módulos del kernel de Linux que permiten el funcionamiento de la herramienta de *iptables*, el filtrado de paquetes, el mantenimiento del estado de las conexiones y el funcionamiento de FTP a través de NAT.

```
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp
```

Tal como se señaló anteriormente, es recomendable iniciar el proceso de establecimiento de reglas con una configuración limpia. Para esto se borrará y reiniciará cualquier tipo de reglas y cadenas previamente establecidas. Asimismo se establece la política por defecto de descarte de paquetes que no estén explícitamente permitidos de las cadenas de entrada (INPUT), de salida (OUTPUT) y de reenvío (FORWARD).

```
$IPTABLES –F            #Flush de todas las cadenas
$IPTABLES -F -t mangle  #Flush de tabla mangle
$IPTABLES -F -t nat     #Flush de tabla nat
$IPTABLES –X            #Flush de cadenas de usuario
$IPTABLES -X -t mangle  #Flush de cadenas tabla mangle
$IPTABLES -X -t nat     #Flush de cadenas de usuario tabla nat
#Politica Restrictiva por defecto (Descartar todos los paquetes no
explicitamente permitidos)
$IPTABLES -P INPUT ACCEPT     #Politica cadena INPUT
$IPTABLES -P OUTPUT ACCEPT    #Politica cadena OUPUT
$IPTABLES -P FORWARD ACCEPT   #Politica cadena FORWARD
```

Dentro de la configuración inicial se incluye también la activación o desactivación de algunas **opciones del kernel** que permiten el enrutamiento de paquetes y le dan a este proceso algo más de seguridad a través del kernel mismo del sistema operativo.

- Se activa la opción para que la plataforma operativa Linux permita el enrutamiento de los paquetes desde la red interna hacia Internet y viceversa.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Al activar la opción `rp_filter` se protege al equipo de ataques de *spoofing* o falsificación de direcciones. Permite verificar si la dirección origen con la que ingresan los paquetes se pueden alcanzar a través de la interfaz por la que ingresaron.

```
echo 2 > /proc/sys/net/ipv4/conf/all/rp_filter
```

- Se activa la opción del kernel para ignorar *pings* de *broadcast*.

```
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

- Se prohíbe paquetes enrutados desde el origen[25] pues este mecanismo podría utilizarse para alcanzar máquinas dentro de una red privada a través de dispositivos intermedios.

```
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
```

- Se deshabilitan las marcas de tiempo de los paquetes TCP, evitando mediante ello que se pueda averiguar el *uptime* (tiempo de funcionamiento) del sistema.

```
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
```

- Se habilitan las *Syn Cookies* como parte de un mecanismo para evitar ataques de inundación *Syn* que provoquen Denegación de Servicio.

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

- No se acepta la redirección de paquetes ICMP, pues podría utilizarse para alterar tablas de enrutamiento. Deberían utilizarlos sólo los dispositivos de *gateway*.

```
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
```

- Se habilita la protección contra mensajes de error falsos de modo que el kernel los ignore e impida que se llenen los *logs*.

```
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

- Se activa la opción de registro de paquetes falsificados (*spoofed*), enrutados desde el origen y paquetes de redirección.

```
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
```

- Se determinará el rango de puertos locales que serán permitidos de usar por las aplicaciones del dispositivo.

```
echo "32768 61000" > /proc/sys/net/ipv4/ip_local_port_range
```

- Al habilitar las siguientes opciones se reduce la posibilidad de ataques DoS al disminuir los *timeouts*, es decir el tiempo en el que Linux determinará que una conexión ha finalizado. La primera opción `tcp_fin_timeout` establece el tiempo en el que Linux terminará una conexión y `tcp_keepalive_time` determina el tiempo para finalizar una conexión que ya no está activa.

```
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 2400 > /proc/sys/net/ipv4/tcp_keepalive_time
```

2.9.7.3 Cadenas de Usuario

En esta sección del *script* se crean reglas propias que posteriormente permitirán el filtrado de tráfico, estas reglas conformarán cadenas adicionales a las existentes por defecto.

Se generan reglas que por un lado permitirán el registro (*logging*) de paquetes pertenecientes a conexiones especiales y por otro lado controlarán la intensidad de registro de estos eventos de modo que no se produzca una inundación de *logs*.

- Se crea la cadena `LINVALID` que detectará paquetes inválidos[26]. En el caso de que las coincidencias estén dentro del límite de velocidad de registro establecido, estos eventos se almacenarán, de otra forma simplemente serán descartados.

```
#Se crea la cadena
$IPTABLES -N LINVALID
#Se agrega a LINVALID una regla que registra el intento de acceso del
paquete si la velocidad de coincidencia está dentro del límite.
$IPTABLES -A LINVALID -m limit --limit $LOGGINGLIMIT --limit-burst
     $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: INVALID:1 DROP"
#Se descarta finalmente el paquete
$IPTABLES -A LINVALID -j DROP
```

[25] El *source routing* o enrutamiento desde el origen es una técnica en la que quien envía un paquete puede especificar la ruta que deberá tomar a través de la red.

[26] Se entiende como paquetes inválidos (INVALID) a aquellos que no posean el estado ESTABLISHED, RELATED o NEW.

La estructura de las siguientes cadenas que se crean será similar, es decir, se generará la cadena, se determinará si un paquete coincide con la regla en el tiempo dentro del límite de velocidad de registro y finalmente se descartará.

- Se crea la cadena `LBADFLAG` que detectará una combinación incorrecta de banderas en el paquete TCP y lo descartará.

```
$IPTABLES -N LBADFLAG
$IPTABLES -A LBADFLAG -m limit --limit $LOGGINGLIMIT --limit-burst
       $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: BADFLAG:1 DROP"
 $IPTABLES -A LBADFLAG -j DROP
```

- Se crea la cadena `LSPECIALPORT` para registrar y descartar intentos de conexión a través de puertos de ataque ya conocidos.

```
$IPTABLES -N LSPECIALPORT
$IPTABLES -A LSPECIALPORT -m limit --limit $LOGGINGLIMIT --limit-burst
   $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: SPECIALPORT:1 DROP"
$IPTABLES -A LSPECIALPORT -j DROP
```

- Se crea la cadena `LSYNFLOOD` que registra y descarta paquetes que forman parte de un ataque de inundación de peticiones `SYN`.

```
$IPTABLES -N LSYNFLOOD
$IPTABLES -A LSYNFLOOD -m limit --limit $LOGGINGLIMIT --limit-burst
      $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: SYNFLOOD:1 DROP"
$IPTABLES -A LSYNFLOOD -j DROP
```

- Se genera también la cadena `LPINGFLOOD` cuyas reglas registran y descartan paquetes ICMP que formen parte de un ataque de inundación ICMP.

```
$IPTABLES -N LPINGFLOOD
$IPTABLES -A LPINGFLOOD -m limit --limit $LOGGINGLIMIT --limit-burst
     $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: PINGFLOOD:1 DROP"
$IPTABLES -A LPINGFLOOD -j DROP
```

- Se crea la cadena **`LDROP`** y sus reglas que descartan y registran paquetes tcp, udp, icmp o fragmentos y la cadena **`LREJECT`** muy similar pero que devuelve un mensaje específico de rechazo hacia quien envió el paquete.

```
$IPTABLES -N LDROP
$IPTABLES -A LDROP -p tcp -m limit --limit $LOGGINGLIMIT --limit-burst
        $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: TCP:1 DROP"
$IPTABLES -A LDROP -p udp -m limit --limit $LOGGINGLIMIT --limit-burst
        $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: UDP:2 DROP"
$IPTABLES -A LDROP -p icmp -m limit --limit $LOGGINGLIMIT --limit-burst
       $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: ICMP:3 DROP"
```

```
$IPTABLES -A LDROP -f -m limit --limit $LOGGINGLIMIT --limit-burst
      $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: FRAGMENT:4 DROP"
$IPTABLES -A LDROP -j DROP
```

```
$IPTABLES -N LREJECT
$IPTABLES -A LREJECT -p tcp -m limit --limit $LOGGINGLIMIT --limit-burst
         $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: TCP:1 REJECT"
$IPTABLES -A LREJECT -p udp -m limit --limit $LOGGINGLIMIT --limit-burst
         $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: UDP:2 REJECT"
$IPTABLES -A LREJECT -p icmp -m limit --limit $LOGGINGLIMIT --limit-burst
        $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: ICMP:3 REJECT"
$IPTABLES -A LREJECT -f -m limit --limit $LOGGINGLIMIT --limit-burst
    $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: FRAGMENT:4 REJECT"
$IPTABLES -A LREJECT -p tcp -j REJECT --reject-with tcp-reset
$IPTABLES -A LREJECT -p udp -j REJECT --reject-with icmp-port-unreachable
$IPTABLES -A LREJECT -j REJECT
```

- Se crea también una cadena que permite detector si un paquete TCP pertenece a un ataque de inundación de peticiones SYN antes de permitir su paso. Se puede notar que para ello se utiliza la cadena **LSYNFLOOD** antes definida. Si el paquete o flujo de paquetes no pertenece a este tipo de ataque, es aceptado.

```
$IPTABLES -N TCPACCEPT
$IPTABLES -A TCPACCEPT -p tcp --syn -m limit --limit $TCPSYNLIMIT –
                              limit-burst $TCPSYNLIMITBURST -j ACCEPT
$IPTABLES -A TCPACCEPT -p tcp --syn -j LSYNFLOOD
$IPTABLES -A TCPACCEPT -p tcp ! --syn -j ACCEPT
```

2.9.7.4 Cadenas de usuario especiales

En esta sección se crean cadenas que detectarán paquetes erróneos o dirigidos hacia puertos característicos por ser el objetivo de ataques *backdoor*[27].

- Se crea la cadena **CHECKBADFLAG** cuyas reglas detectan combinaciones erróneas de banderas en los paquetes tcp y en caso de existir, se aplica la cadena **LBADFLAG** que descarta esos paquetes.

```
$IPTABLES -N CHECKBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL FIN,URG,PSH -j LBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j LBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL ALL -j LBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL NONE -j LBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags SYN,RST SYN,RST -j LBADFLAG
$IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags SYN,FIN SYN,FIN -j LBADFLAG
```

[27] Los ataques *backdoor* representan mecanismos de control remoto de sistemas que utilizan determinados puertos por defecto.

- Se crea la cadena SMB que descartará sin registrar todos los paquetes tcp y udp que se originen o se dirijan en/hacia un servicio SMB[28] para evitar revelar información de este tipo hacia Internet. Todos los puertos detallados podrían ser utilizados por las distintas versiones de este protocolo para compartir archivos e incluso virus.

```
$IPTABLES -N SMB
$IPTABLES -A SMB -p tcp --dport 137 -j DROP
$IPTABLES -A SMB -p tcp --dport 138 -j DROP
$IPTABLES -A SMB -p tcp --dport 139 -j DROP
$IPTABLES -A SMB -p tcp --dport 445 -j DROP
$IPTABLES -A SMB -p udp --dport 137 -j DROP
$IPTABLES -A SMB -p udp --dport 138 -j DROP
$IPTABLES -A SMB -p udp --dport 139 -j DROP
$IPTABLES -A SMB -p udp --dport 445 -j DROP
$IPTABLES -A SMB -p tcp --sport 137 -j DROP
$IPTABLES -A SMB -p tcp --sport 138 -j DROP
$IPTABLES -A SMB -p tcp --sport 139 -j DROP
$IPTABLES -A SMB -p tcp --sport 445 -j DROP
$IPTABLES -A SMB -p udp --sport 137 -j DROP
$IPTABLES -A SMB -p udp --sport 138 -j DROP
$IPTABLES -A SMB -p udp --sport 139 -j DROP
$IPTABLES -A SMB -p udp --sport 445 -j DROP
```

- Es necesario crear una cadena cuyas reglas registrarán y descartarán paquetes que se dirijan hacia puertos especiales utilizados generalmente por troyanos a través de los que se pueden establecer conexiones remotas a los equipos.

```
$IPTABLES -N SPECIALPORTS
#Escaneo Deepthroat
$IPTABLES -A SPECIALPORTS -p  tcp --dport 6670 -j LSPECIALPORT
#Escaneo Subseven
$IPTABLES -A SPECIALPORTS -p tcp --dport 1243 -j LSPECIALPORT
$IPTABLES -A SPECIALPORTS -p udp --dport 1243 -j LSPECIALPORT
$IPTABLES -A SPECIALPORTS -p tcp --dport 27374 -j LSPECIALPORT
$IPTABLES -A SPECIALPORTS -p udp --dport 27374 -j LSPECIALPORT
$IPTABLES -A SPECIALPORTS -p tcp --dport 6711:6713 -j LSPECIALPORT
#Escaneo Netbus
$IPTABLES -A SPECIALPORTS -p tcp --dport 12345:12346 -j LSPECIALPORT
$IPTABLES -A SPECIALPORTS -p tcp --dport 20034 -j LSPECIALPORT
#Escaneo Back Orifice
$IPTABLES -A SPECIALPORTS -p udp --dport 31337:31338 -j LSPECIALPORT
#X-Windows
$IPTABLES -A SPECIALPORTS -p tcp --dport $XWINPORTS  -j LSPECIALPORT
#Hack'a'Tack 2000
```

[28] SMB-*Server Message Block* es un protocolo que se utiliza para compartir archivos e impresoras en redes Windows. Su uso hacia Internet puede considerarse una vulnerabilidad por la facilidad para compartir estos archivos e impresoras.

```
$IPTABLES -A SPECIALPORTS -p udp --dport 28431 -j LSPECIALPORT
```

Resulta conveniente también registrar y descartar las conexiones hacia puertos que son utilizados por el protocolo Xwindow[29]. En general todos estos intentos de conexión serán descartados por la política por defecto, sin embargo es útil registrar estos intentos para determinar el posible origen de los ataques.

Para finalizar con la creación de cadenas específicas de usuario, se generarán dos cadenas que permitan controlar el filtrado de paquetes de control muy importantes como los correspondientes al protocolo ICMP y una herramienta basada en éste como es **`traceroute`**. Los mensajes ICMP, de la misma forma en que son de mucha utilidad para determinar si determinados *hosts* en Internet o en la *intranet* pueden alcanzarse, pueden utilizarse también para ejecutar ataques que saturen el dispositivo de seguridad o los servidores de comunicaciones e incluso mediante los cuales se revele información sobre los sistemas que pueda ser utilizada para vulnerarlos. Este último caso se puede aplicar también a la herramienta **`traceroute`** que podría utilizarse para descubrir la estructura de direccionamiento interno de la corporación.

Por estas razones se incluyen dos cadenas que filtrará adecuadamente el flujo de mensajes ICMP que ingresan al *firewall-box* (cadena ICMPINBOUND) y de aquellos que salen de él (cadena ICMPOUTBOUND).

```
$IPTABLES -N ICMPINBOUND
#Proteccion contra Inundacion Ping - Se acepta un limite de
peticiones/segundo y el resto se descartan
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type echo-request -m limit –-
                limit $PINGLIMIT --limit-burst $PINGLIMITBURST -j ACCEPT
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type echo-request -j LPINGFLOOD
#Bloquear ICMP-Redirects - Aun cuando ya deberian filtrarse gracias a las
opciones de kernel
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type redirect -j LDROP
#Bloquear paquetes ICMP-Timestamp - deberian ya filtrarse gracias a las
opciones de kernel
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type timestamp-request -j LDROP
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type timestamp-reply -j LDROP
#Bloquear mensajes ICMP-address-mask (puede evitar problemas de OS-
fingerprinting[30])
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type address-mask-request -j LDROP
$IPTABLES -A ICMPINBOUND -p icmp --icmp-type address-mask-reply -j LDROP
```

[29] Xwindow es un protocolo que permite la interacción gráfica en red entre un usuario y una o más computadoras.

[30] *OS-fingerprinting* se refiere a la posibilidad de determinar el tipo de plataforma operativa que se utiliza en un sistema, a partir de la forma de responder a determinado tipo de petición.

```
#Permitir todos los paquetes ICMP restantes que ingresan
$IPTABLES -A ICMPINBOUND -p icmp -j ACCEPT
```

```
$IPTABLES -N ICMPOUTBOUND
# Bloquear paquetes ICMP-Redirect (ya deberian descartarse mediante las
opciones de kernel)
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type redirect -j LDROP
#Traceroute MS (MS usa ICMP en lugar de UDP para hacer tracert)
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type ttl-zero-during-transit -j LDROP
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type ttl-zero-during-reassembly -j LDROP
#Bloquear paquetes ICMP-Parameter-Problem
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type parameter-problem -j LDROP
#Bloquear paquetes ICMP-Timestamp (ya deberian descartarse mediante las
opciones de kernel)
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type timestamp-request -j LDROP
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type timestamp-reply -j LDROP
#Bloquear paquetes ICMP-address-mask (permite prevenir OS-fingerprinting)
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type address-mask-request -j LDROP
$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type address-mask-reply -j LDROP
#Aceptar el resto de paquetes ICMP salientes
$IPTABLES -A ICMPOUTBOUND -p icmp -j ACCEPT
```

2.9.7.5 Aplicación de Cadenas y Reglas de Firewall

2.9.7.5.1 Cadena INPUT

Como ya se indicó, esta cadena está conformada por reglas de tráfico de los paquetes que se dirigen hacia el *firewall-box*, es decir que tienen éste dispositivo como su destino, ya sea hacia a través de la interfaz interna desde la *intranet* o a través de la interfaz externa desde Internet.

Esta sección iniciará con dos reglas de *filtrado general* que determinarán si los paquetes que intentan entrar en el dispositivo de *firewall* son inválidos o si poseen combinaciones de banderas erróneas, mediante la utilización de dos de las cadenas creadas en el inicio de la configuración.

```
 # Descartar paquetes invalidos (que no son del tipo ESTABLISHED, RELATED
o NEW)
  $IPTABLES -A INPUT -m state --state INVALID -j LINVALID
  # Descartar paquetes con banderas erroneas
  $IPTABLES -A INPUT -p tcp -j CHECKBADFLAG
```

En relación a la interfaz virtual de *loopback* se deben tomar algunas consideraciones especiales y permitir el tráfico a través de ella pues permite realizar pruebas de diagnóstico y conectividad del propio dispositivo de red y de la validez del protocolo de comunicación.

```
$IPTABLES -A INPUT -i lo -j ACCEPT
```

Una regla importante para el funcionamiento adecuado de las conexiones establecidas es la que debe permitir el ingreso de los paquetes que pertenecen a una conexión ya establecida o a una que esté relacionada con una conexión existente.

```
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ahora, es necesario hacer un análisis de los paquetes que intentan ingresar al *firewall-box,* provenientes de la *intranet*. Se inicia filtrando el tráfico ICMP mediante las cadenas antes creadas y descartando el tráfico `traceroute` al bloquear los puertos que esta herramienta pueda utilizar.

```
#Filtrar ICMP
$IPTABLES -A INPUT -i $INTERNAL_INT -p icmp -j ICMPINBOUND
#Bloquear traceroute UDP
$IPTABLES -A INPUT -p udp --dport 33434:33523 -j LDROP
```

En este caso, el dispositivo de *firewall* redirigirá las peticiones DNS de la LAN hacia Internet, permitirá conexiones vía SSH y atenderá peticiones de proxificación en su interfaz interna. También funcionará como servidor VPN y se permitirá el tráfico por las interfaces virtuales que este esquema utiliza.

```
##Permitir el ingreso de peticiones DNS
$IPTABLES -A INPUT -i $INTERNAL_INT -p udp -s $INTERNAL_LAN -d $ALL --
dport 53 -j ACCEPT
$IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN -d $ALL --
dport 53 -j ACCEPT
##Permitir el ingreso de peticiones al servicio de PROXY SQUID
$IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN --sport
$HIGHPORTS --dport 8080 -j TCPACCEPT
##SSH desde INTRAnet
$IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN --sport
$HIGHPORTS --dport 22 -j TCPACCEPT
##OpenVPN desde la INTRAnet e INTERNET
$IPTABLES -A INPUT -i tun+ -j ACCEPT
$IPTABLES -A INPUT -p udp --dport 1194 -j ACCEPT #Puerto VPN
```

Cualquier paquete que se origine en la *intranet* (`INTERNAL_LAN`) y que trate de ingresar al dispositivo a través de una interfaz diferente a la interna se rechazará y registrará, previniendo la falsificación de paquetes.

```
$IPTABLES -A INPUT -s $INTERNAL_LAN -j LREJECT
```

El flujo de paquetes provenientes de Internet debe filtrarse cuidadosamente también. Como en el caso anterior se controla el tráfico ICMP y de `traceroute`,

descartando el tráfico SMB y rechazando silenciosamente las conexiones *Ident*[31]. Asi también se permite el acceso vía SSH mediante la interfaz externa y se registran y descartan los intentos de conexión a través de puertos especiales, señalados en la cadena LSPECIALPORTS.

```
#Filtrar ICMP
$IPTABLES -A INPUT -i $EXTERNAL_INT -p icmp -j ICMPINBOUND
#Bloquear traceroute UDP
$IPTABLES -A INPUT -p udp --dport 33434:33523 -j LDROP
#Descartar todo el trafico SMB
$IPTABLES -A INPUT -i $EXTERNAL_INT -j SMB
#Rechazar Ident de forma silenciosa
$IPTABLES -A INPUT -i $EXTERNAL_INT -p tcp --dport 113 -j REJECT --
reject-with tcp-reset
##SSH desde INTERnet
$IPTABLES -A INPUT -i $EXTERNAL_INT -p tcp --dport 22 -j TCPACCEPT
##Registro de intentos de conexion o escaneo de puertos especiales
$IPTABLES -A INPUT -i $EXTERNAL_INT -j SPECIALPORTS
##Descartar el resto de paquetes entrantes
$IPTABLES -A INPUT -j LDROP
```

2.9.7.5.2 Cadena OUTPUT

El filtrado de tráfico en esta cadena es más relajado que en la anterior pues sobre la salida de paquetes que se originan en el *firewall-box* se tiene mucho más control.

Se puede iniciar habilitando el tráfico de salida para la interfaz de *loopback*.

```
$IPTABLES -A OUTPUT -o lo -j ACCEPT
```

Es posible permitir todo el tráfico saliente hacia la red interna, ya que esto no implica ningún riesgo, siempre y cuando salga a través de la interfaz interna.

```
$IPTABLES -A OUTPUT -o $INTERNAL_INT -d $INTERNAL_LAN -j ACCEPT
```

Entre los paquetes que se dirigen hacia Internet, se permitirá la salida a través de la interfaz virtual (túnel) y aquellos cuyo puerto de origen represente el servicio de VPN (1194 en este caso). En el anexo D se puede observar una pequeña guía de cómo configurar una VPN en Linux.

[31] *Ident* es un protocolo de Internet que permite identificar el usuario de una conexión particular TCP y en ese sentido puede representar un problema de seguridad. Sin embargo descartar estas conexiones podría provocar retardos significativos en el proceso de conexión al usar otros protocolos como FTP.

También se filtrará el tráfico ICMP, SMB, Ident aceptando todo flujo de paquetes cuyos puertos origen sean aleatorios y mayores a 1024[32].

```
#Permitir flujo OpenVPN
$IPTABLES -A OUTPUT -o tun+ -j ACCEPT
$IPTABLES -A OUTPUT -p udp --sport 1194 -j ACCEPT
#ICMP y Traceroute
$IPTABLES -A OUTPUT -o $EXTERNAL_INT -p icmp -j ICMPOUTBOUND
#Filtrado SMB
$IPTABLES -A OUTPUT -o $EXTERNAL_INT -j SMB
#Filtrado Ident
$IPTABLES -A OUTPUT -o $EXTERNAL_INT -p tcp --sport 113 -j REJECT --
reject-with tcp-reset
#Aceptar todo el trafico tcp/udp saliente en puertos no privilegiados
$IPTABLES -A OUTPUT -o $EXTERNAL_INT -s $EXTERNAL_IP -p tcp --sport
$HIGHPORTS -j ACCEPT
$IPTABLES -A OUTPUT -o $EXTERNAL_INT -s $EXTERNAL_IP -p udp --sport
$HIGHPORTS -j ACCEPT
```

Y se descarta todo el tráfico de salida restante.

```
$IPTABLES -A OUTPUT -j LDROP
```

2.9.7.5.3 Cadena FORWARD

Esta cadena de reenvío se encargará de filtrar y discriminar todos aquellos paquetes que intenten atravesar el dispositivo de *firewall*.

Se realizará un filtrado general de estos paquetes mediante reglas que descarten paquetes inválidos y aquellos que posean combinaciones de banderas inválidas, propias de ataques de escaneo.

```
$IPTABLES -A FORWARD -m state --state INVALID -j LINVALID
$IPTABLES -A FORWARD -p tcp -j CHECKBADFLAG
```

- Se permite el paso de paquetes pertenecientes a conexiones válidas establecidas o relacionadas con una de ellas.

```
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

A continuación se detallan las reglas que permitirán (aceptarán) las conexiones hacia determinados servicios requeridos por la *intranet* de la corporación.

Estos servicios se identifican por el puerto destino de los paquetes de petición de quienes solicitan el servicio, que son en este caso los usuarios de la *intranet.*

Se permite el paso de tráfico ICMP hacia Internet.

[32] Se refiere a los puertos mayores a 1024 que identifican al cliente en una conexión en el campo de puerto

```
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN – p icmp -j ACCEPT
```

- Se descarta el tráfico de paquetes SMB a través de la regla creada con ese nombre.

```
$IPTABLES -A FORWARD -o $EXTERNAL_INT -j SMB
```

- En caso de que el servicio de reenvío de peticiones DNS ubicado en el *firewall* falle, podría ser útil permitir que estas peticiones puedan ser atendidas por servidores en Internet, siempre que estas peticiones se originen en la *intranet*.

```
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p udp --sport $HIGHPORTS -d $ALL --dport 53 -j ACCEPT
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p tcp --sport $HIGHPORTS -d $ALL --dport 53 -j ACCEPT
```

- Se permitirá el tráfico SSH para el acceso remoto y monitoreo de hosts a través de Internet.

```
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      –p tcp --sport $HIGHPORTS -d $ALL --dport 22 -j ACCEPT
```

- El tráfico FTP, como se analizó con anterioridad, deberá poder atravesar el *firewall* de la corporación.

```
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p tcp --sport $HIGHPORTS -d $ALL --dport 20 -j ACCEPT
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p tcp --sport $HIGHPORTS -d $ALL --dport 21 -j ACCEPT
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p udp --sport $HIGHPORTS -d $ALL --dport 20 -j ACCEPT
$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN
      -p udp --sport $HIGHPORTS -d $ALL --dport 21 -j ACCEPT
```

- El tráfico HTTPS (puerto 443), POP (puertos 109-110), IMAP (puerto 143), SMTP (puerto 25), se permitirá para que funcionen algunas aplicaciones importantes que se manejan en la corporación como el correo electrónico, además de otras variaciones que le dan más seguridad a la transmisión de información (SSL).

```
#Reenvio de trafico HTTPS
$IPTABLES -A FORWARD -m state --state NEW -p tcp --dport 443 -j ACCEPT
#Trafico POP3
```

origen del paquete TCP o UDP.

```
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL
      --dport 110 -j ACCEPT
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL
      --dport 109 -j ACCEPT
#Trafico SMTP
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL
      --dport 25 -j ACCEPT
$IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 25 -j ACCEPT
#Trafico IMAP2
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 143 -j ACCEPT
$IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 143 -j ACCEPT
#Trafico IMAP3
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 220 -j ACCEPT
$IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 220 -j ACCEPT
#Trafico IMAPS (IMAP sobre SSL)
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 993 -j ACCEPT
$IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 993 -j ACCEPT
#Trafico POP3S (POP3 sobre SSL)
$IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 995 -j ACCEPT
$IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL
      --dport 995 -j ACCEPT
```

- En el caso de existir servicios que se presten a Internet como Web o FTP, deberán generarse reglas que permitan el reenvío de este tipo de comunicación.
- Finalmente se filtra el tráfico de reenvío que ingresa a la *intranet*, registrando los paquetes que pertenecen a conexiones SMB, descartándolos junto con los restantes.

```
#Trafico SMB
$IPTABLES -A FORWARD -i $EXTERNAL_INT -j SMB
#Descartar paquetes de FORWARDING restantes
$IPTABLES -A FORWARD -j LDROP
```

2.9.7.5.4 Cadena PREROUTING

Esta cadena contendrá reglas que procesarán paquetes antes de que estos sean enrutados. En general, permitirán la redirección del tráfico que va al *firewall* hacia servidores públicos en la *intranet*. Por ejemplo una petición hacia el puerto 80 en la interfaz externa del *firewall* será re-dirigida hacia el servidor web de la corporación si existiese.

Primeramente se creará la regla que permita la proxificación al redirigir todas las peticiones desde la *intranet* que van puerto 80 (peticiones HTTP) hacia el puerto 8080 en el que funciona el servicio de *proxy*.

```
$IPTABLES -t nat -A PREROUTING -p tcp -s $INTERNAL_LAN -d $ALL --dport 80
      -j REDIRECT --to-port 8080
```

Si existiese un servidor Web al interior de la red de la corporación, se redirigirán los paquetes que intenten acceder al *router* con peticiones HTTP hacia el equipo servidor correspondiente. Lo mismo se aplicará para los servicios FTP y de correo electrónico (SMTP).

```
# Redireccion de puerto de acceso al servidor Web Hacia el servidor
correspondiente
$IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP
--dport 80 -j DNAT --to $HTTP_IP
# Redireccion de peticion ftp hacia el servidor ftp
$IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP
      --dport 20 -j DNAT --to $FTP_IP
$IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP
      --dport 21 -j DNAT --to $FTP_IP
```

2.9.7.5.5 Cadena POSTROUTING

Esta cadena permitirá procesar paquetes luego de que estos hayan sido enrutados. En este caso es útil para “enmascarar” todos los paquetes que se originan en la *intranet* y tienen como destino Internet, de manera que puedan ser enrutados y viajar por la red de redes, con la dirección IP de la interfaz pública del *router* en el campo de dirección IP origen de la cabecera de los paquetes.

```
$IPTABLES -A POSTROUTING -t nat -o $EXTERNAL_INT -j MASQUERADE
```

CAPÍTULO 3

IMPLEMENTACIÓN, PRUEBAS Y COSTO DEL PROTOTIPO

3.1 IMPLEMENTACIÓN DEL PROTOTIPO DE VOZ

3.1.1 CARACTERÍSTICAS DE HARDWARE DEL PROTOTIPO DE VOZ

El elemento fundamental del prototipo es una **estación de trabajo** con las siguientes características:

- Procesador AMD Athlon 1600 de 1500 MHz
- Memoria RAM de 768 MB
- Disco Duro de 40 GB
- 2 Tarjetas de Red 10/100 Mbps

Se usará como interfaz con la red telefónica pública conmutada una tarjeta genérica del modelo X100P de Digium (ver anexo C) con un módulo FXO, conectada a una ranura PCI en la tarjeta madre de la estación.

Asimismo se empleará un switch de marca D-Link y un adaptador de teléfono analógico (ATA) de marca Linksys cuyas especificaciones están indicadas también en el anexo C.

3.1.2 INSTALACIÓN DEL SISTEMA OPERATIVO

Tal como se recomendó en la etapa de diseño, se instalará la distribución CentOS versión 4.3, un clon binario de RedHat, muy sencillo de actualizar y con soporte de paquetes, al menos durante 5 años más.

La instalación debe llevarse a cabo, de acuerdo a los parámetros indicados en la sección 2.8.3. Una instalación adecuada y segura es vital cuando se trate de cualquier clase de servidor de comunicaciones.

3.1.3 INSTALACIÓN DE SOFTWARE NECESARIO PARA EL FUNCIONAMIENTO DE LA CENTRAL IP

Tal y como se indicó en el capítulo anterior, específicamente en la sección 2.2.2.3, es necesario instalar una serie de paquetes de audio, de bases de datos, del kernel, etc., sin los cuales la instalación del servidor *Asterisk* no sería posible.

3.1.4 INSTALACIÓN DEL SOFTWARE DE *ASTERISK* Y PAQUETES ADICIONALES

Con las fuentes de los paquetes **libpri, asterisk, zaptel, asterisk-addons** y **asterisk-sounds**, el proceso de compilación e instalación es muy sencillo e incluso rápido. En el anexo B se describe claramente el proceso de instalación de cada uno de estos paquetes, así como aquellos que son prerrequisito para esta instalación.

3.1.5 INSTALACIÓN DE HARDWARE

El proceso de instalación de *hardware* no es otro más que el típico relacionado con agregar dispositivos PCI a la tarjeta madre de una estación de trabajo (tarjeta X100P o cualquier otra de *Digium*) o conectar dispositivos de telecomunicaciones a través de un cable telefónico y un cable de red CAT5.

Eso a breves rasgos es lo que involucra la interconexión de los dispositivos que permitirán a los clientes interactuar con el servidor *Asterisk* y viceversa.

- La tarjeta X100P se conecta a una ranura PCI de la tarjeta madre de la PC, como una tarjeta de red común o una tarjeta de sonido.
- En el puerto FXO de la tarjeta se conecta cualquier cable telefónico de 2 hilos y conectores RJ11 hacia la línea telefónica (PSTN).
- El servidor de telefonía irá conectado a Internet a través de un cable de red directo.
- El adaptador ATA PAP2 *Linksys* se comunica con la LAN interna a través de un cable de red directo CAT5 conectado a un puerto Ethernet en un switch D-Link en un extremo y en el otro, desde luego, al puerto Ethernet correspondiente al adaptador. Mediante un puerto RJ11 el adaptador se conecta también con un teléfono analógico común y corriente.
- Finalmente se conectan varias computadoras portátiles y PCs con Sistemas operativos Windows y Linux para que funcionen dentro del entorno de VoIP como clientes a través de *softphones.*

3.1.6 CONFIGURACIÓN DEL SISTEMA

El sistema se configura inicialmente creando las extensiones SIP correspondientes a cada usuario que podrá registrarse en el servidor de telefonía. Estas extensiones se crean en el archivo **sip.conf**, dentro de la ruta */etc/asterisk*, y la estructura de extensiones SIP del prototipo, puede apreciarse dentro del anexo A. Se ubicarán únicamente las extensiones correspondientes a la operadora, al Administrador de la corporación y los miembros de 2 de las empresas del parque.

Se debe configurar adecuadamente los archivos que rigen el funcionamiento de la interfaz analógica X100P mediante la edición de los archivos **zaptel.conf** y **zapata.conf**, verificando antes el reconocimiento de la tarjeta por parte del sistema operativo tal y como se muestra en la sección 2.3.2.1**.**

Una vez configuradas las interfaces de comunicación, se debe especificar el plan de marcado (ver anexo A) en el archivo **extensions.conf**. La funcionalidad de la central de telefonía IP se centrará en esta configuración y en ella se detallará el manejo de las llamadas a través del servidor. Se incluirá en este plan todas las aplicaciones que se analizaron en el levantamiento de requerimientos del capítulo anterior.

3.2 IMPLEMENTACIÓN DEL PROTOTIPO DE FIREWALL

3.2.1 CARACTERÍSTICAS DE HARDWARE

El *hardware* para el prototipo de *firewall* involucra únicamente una PC que dispone de dos tarjetas de red. En este caso, dado que el módem de conexión a Internet dispone de un puerto USB es posible conectarlo a la PC sin necesidad de una tarjeta de red. La tarjeta de red de que dispone la PC permitirá conectarla a la red interna a través de un *switch* D-Link.

Las características de la estación que se utilizará en el prototipo son las siguientes:

- Procesador AMD Athlon de 3.0 GHz
- Memoria RAM de 2 GB
- Disco Duro de 120 GB
- Tarjeta de Red 10/100 Mbps

3.2.2 INSTALACIÓN DEL SISTEMA OPERATIVO

Al igual que para cualquier servidor, y muy especialmente en uno que brinda protección a una red privada, deben tomarse las medidas de precaución adecuados, de manera que se haga más difícil el trabajo de *hackers* y *crackers* al intentar acceder a nuestros sistemas. La actualización de la plataforma operativa, la implementación de parches y el manejo de contraseñas fuertes, tal como se indica en la sección 2.8.3, harán de la barrera cortafuegos un dispositivo más blindado.

3.2.3 CONFIGURACIÓN DEL SCRIPT DE FIREWALL

Luego del análisis detallado de las reglas a aplicarse, de acuerdo a los requerimientos básicos de conexión de la corporación, lo que resta es la aplicación de estas reglas a través de la ejecución del *script* generado en el capítulo anterior y que se adjunta en el anexo E.

3.3 PRUEBAS DEL PROTOTIPO DE VOZ

3.3.1 DEFINICIÓN DEL AMBIENTE DE PRUEBAS

En esta sección se define el esquema de pruebas del prototipo. Para ello se dispondrá de un servidor de pruebas de voz sobre IP, con las características mencionadas anteriormente.

Se simulará una LAN formada por 5 dispositivos que estarán conectados a un *switch* y, a través de éste a un servidor de acceso a Internet mediante NAT y proxificación. Estos 5 dispositivos representarán los usuarios internos de telefonía IP del servidor *Asterisk,* 4 de los cuales serán computadores con *software telefónico* (3 portátiles con S.O. Linux y una PC con Windows) y el quinto será el adaptador ATA *Linksys* al que se conectará un teléfono analógico convencional. El *softphone* que se utilizará para Linux es *Ekiga,* que viene por defecto en la distribución *Ubuntu 6.10.* El software telefónico correspondiente para *Windows* será la versión no comercial de *Xten-XLite*.

Un último usuario se conecta a través de Internet y la interfaz externa del servidor.

El proceso de configuración de estos clientes está detallado en el capítulo anterior, en la sección 2.3.5.1.

Las direcciones IP que se asignarán estarán desde la 10.0.0.101 a la 10.0.0.105 con una máscara de 24 bits, y la puerta de enlace predeterminada (la dirección IP del servidor proxy/NAT) será 10.0.0.1/24.

El servidor *Asterisk* estará conectado a Internet con una dirección 190.10.228.28/24 y a la *intranet* con dirección 10.0.0.3/24 a través de las dos interfaces de red que posee.

Esta conexión especial permitirá que los usuarios internos se conecten al servidor *Asterisk* a través de la LAN de la que forman parte, y los usuarios que no se encuentren en la *intranet* se podrán conectar a través de la dirección IP pública mencionada, evitando que el tráfico de voz proveniente de redes externas tenga que ingresar en la LAN, en el caso de que el servidor de voz fuese solamente interno. Esta situación generaría muchos problemas, especialmente en lo que al manejo del *firewall* y el protocolo SIP se refiere, pues SIP maneja puertos aleatorios para la transferencia de información, lo que se vuelve muy difícil de manejar a través de un dispositivo de filtrado.

Los departamentos a los que representará cada uno de los clientes/dispositivos están indicados en la tabla 3.1, así como se refiere también la información de direccionamiento, nombres de usuario y contraseña de cada uno.

Las pruebas consistirán básicamente en comprobar el funcionamiento de todas las aplicaciones cuya configuración se propuso con anterioridad, en base al siguiente proceso:

- Registro de clientes en el servidor.
- Establecimiento de llamadas internas.
- Establecimiento de llamadas hacia la PSTN.
- Ingreso de llamadas desde la PSTN.
- Aplicaciones de voz adicionales.

TABLA 3.1 ESQUEMA DE DIRECCIONAMIENTO Y AUTENTICACIÓN CON EL SERVIDOR ASTERISK

Dpto.	Username	Password	Tipo de cliente	Dirección IP
Gerencia	gerente1_ref	gerente1_ref	ATA+teléfono	10.0.0.101/24
Pagos	pagos_ref	pagos_ref	Softphone	10.0.0.102/24
Ventas	ventas_ref	ventas_ref	Spftphone	10.0.0.103/24
Soporte	soporte_ref	soporte_ref	Softphone	10.0.0.104/24
Operación	operadora	operadora	Softphone	10.0.0.105/24
Cliente	cliente	cliente	softphone	201.217.88.104/24
Asterisk	-	-	-	190.10.228.28/24 10.0.0.3/24

El escenario creado tendrá la estructura mostrada en la figura 3.1.

3.3.2 DESARROLLO DE PRUEBAS

3.3.2.1 Registro de clientes

En cada uno de los dispositivos mencionados se configuran los clientes *softphones* y el ATA, tal como se indicó en la sección 2.3.5, tomando en cuenta el direccionamiento de la tabla 3.1 y el esquema de la figura 3.1 y sin olvidar que la dirección IP del servidor con la que se configurarán estos clientes internos deberá ser 10.0.0.3/24 (interfaz interna).

Se puede observar en la tabla 3.2 el proceso de registro exitoso (capturado con la herramienta *Ethereal*) del usuario `soporte_ref`, perteneciente al departamento de Soporte de la empresa *Refundation Consulting Group.* En el caso de los teléfonos analógicos, la configuración respectiva debe hacerse en el ATA, que es el dispositivo que se autentica en el servidor *Asterisk*. De este modo, se configurará el *username,* contraseña y parámetros IP correspondientes de los clientes, como se muestra en la sección 2.3.5.2.

Resultados.- *El registro de cada una de las cuentas SIP de prueba es exitoso, incluyendo los softphones y el ATA Linksys*.

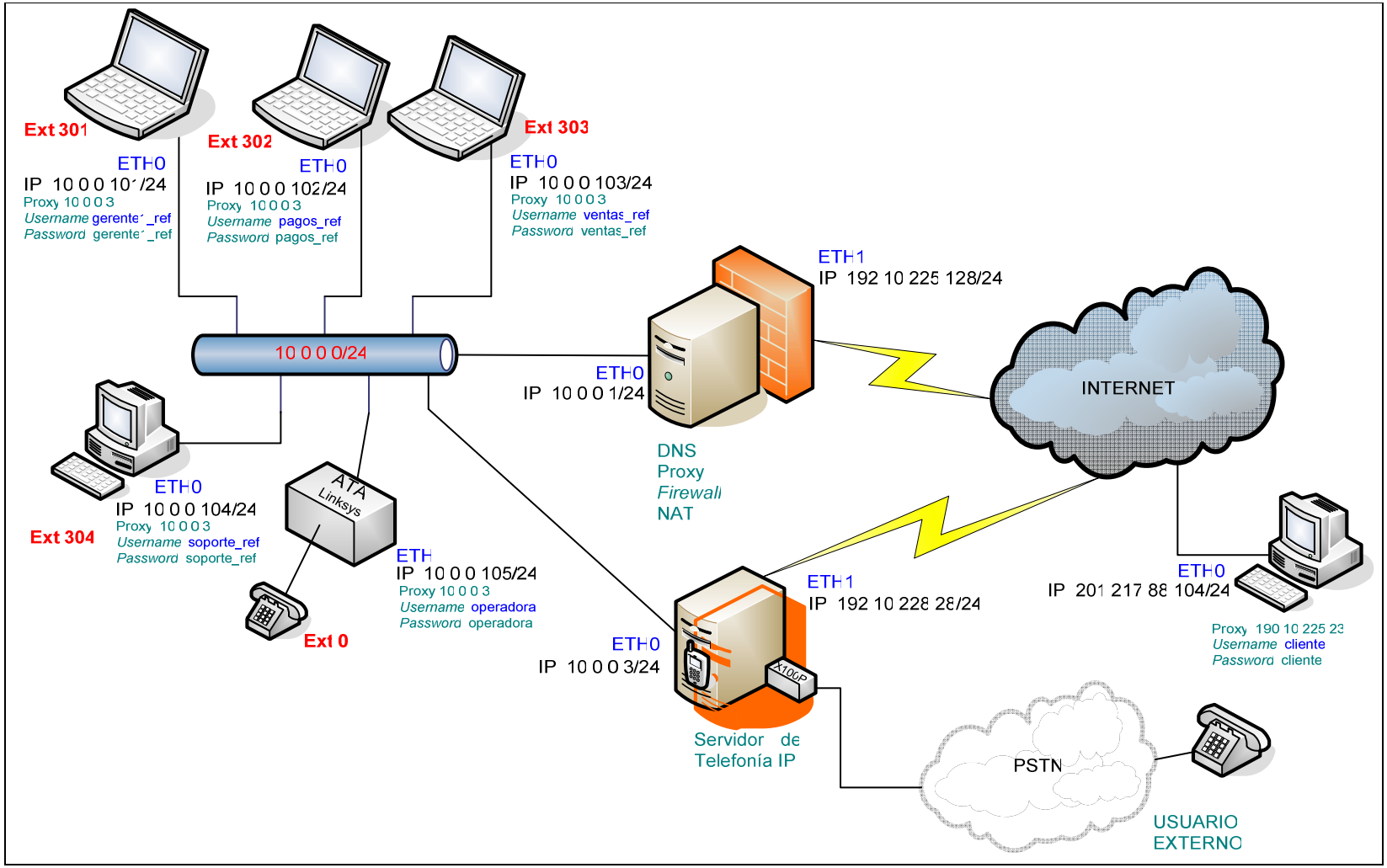


FIGURA 3.1 ESQUEMA DE RED DEL ESCENARIO DE PRUEBAS DEL PROTOTIPO DE VOZ

TABLA 3.2 MENSAJES DE REGISTRO DEL CLIENTE `soporte_ref` EN EL SERVIDOR *ASTERISK*

IP ORIGEN	IP DESTINO	Mensaje
10.0.0.104	10.0.0.3	Request: REGISTER sip:10.0.0.3
10.0.0.3	10.0.0.104	Status: 100 Trying (1 bindings)
10.0.0.104	10.0.0.3	Request: REGISTER sip:10.0.0.3
10.0.0.3	10.0.0.104	Status: 100 Trying (1 bindings)
10.0.0.3	10.0.0.104	Status: 200 OK (0 bindings)
10.0.0.104	10.0.0.3	Request: REGISTER sip:10.0.0.3
10.0.0.3	10.0.0.104	Status: 100 Trying (1 bindings)
10.0.0.104	10.0.0.3	Request: REGISTER sip:10.0.0.3
10.0.0.3	10.0.0.104	Status: 100 Trying (1 bindings)
10.0.0.3	10.0.0.104	Request: OPTIONS sip:soporte_ref@10.0.0.104:25496;…
10.0.0.3	10.0.0.104	Status: 200 OK (1 bindings)
10.0.0.104	10.0.0.3	Status: 200 OK

3.3.2.2 Establecimiento de llamadas

El objetivo básico de un sistema de voz es permitir a los usuarios de este sistema comunicarse entre sí y con otros sistemas de voz. El servidor de telefonía dentro del escenario que se plantea, permite comunicar a los clientes directos de la central IP, en este caso las áreas de Gerencia, Soporte, Ventas, Pagos y la operadora.

Las pruebas que verifican este funcionamiento son:

1. Establecimiento de llamada de gerencia a ventas durante 3 minutos
2. Establecimiento de llamada de la operadora a soporte durante 2 minutos
3. Establecimiento de llamada de Ventas a la operadora durante 1 minuto

***Resultados*.-** Las llamadas individuales son exitosas, se establecen normalmente pues quienes las reciben están registrados en el servidor de Voz y contestan cuando empieza el timbrado de sus terminales (*softphones* y ATA).

Por otro lado, al disponer de una tarjeta con un módulo analógico FXO en el servidor, éste debe ser capaz de funcionar como *gateway* y permitir la salida de llamadas desde el interior de la red LAN hacia la PSTN, así como el ingreso de comunicaciones desde la red telefónica pública al interior de la LAN a través de la misma interfaz.

En este sentido, las pruebas que se realizan son las siguientes:

- Se realiza una llamada desde todos los departamentos hacia un número local (el escenario dispone de dos líneas telefónicas, una conectada al servidor y otra para simular un cliente externo).
- Se realiza una llamada desde un número telefónico dentro de la ciudad (a través de la línea telefónica de pruebas) a la *troncal* (línea telefónica a la que está conectado el servidor de telefonía) y se digita un patrón de dígitos para comunicarse con cada uno de los departamentos de los que está formado el escenario.

Resultados.- Las dos pruebas se ejecutan de manera satisfactoria, las llamadas desde la LAN hacia la PSTN se enrutan adecuada y transparentemente, permitiendo que se establezca la comunicación. Del mismo modo, las llamadas hacia la *troncal* son atendidas por el mecanismo de IVR que guía a los llamantes para alcanzar cada uno de los departamentos a través del marcado de su número de extensión.

Dado que el servidor también se encuentra conectado a Internet, existe la posibilidad de que cualquier persona, incluso un cliente, pueda comunicarse a bajo costo con las empresas a través de este servidor en Internet, únicamente registrándose en él.

- Se realiza una prueba de registro y establecimiento de llamada desde otro punto en Internet con un softphone *Xlite* en *Windows* a través de la que se comunica con los departamentos del escenario de pruebas.

Resultados.- El registro del dispositivo es exitoso, si bien pueden generarse inconvenientes si los clientes en Internet se encuentran detrás de dispositivos que hacen NAT. Las llamadas se establecen sin problemas y con resultados favorables. Un inconveniente notorio es la presencia de retardos cuanto menos ancho de banda existe entre los dispositivos que forman parte de la llamada.

3.3.2.3 Acceso a Aplicaciones de Voz

3.3.2.3.1 *IVR/Control de horario*

- Para comprobar el funcionamiento del servicio de IVR (respuesta de voz interactiva), se realiza una llamada desde una línea convencional hacia la *troncal* conectada al servidor *Asterisk* y para verificar el servicio de atención en función del horario se manipula la hora dentro de la función `GoToIfTime( )` del plan de marcado.

***Resultados*.-** La llamada es contestada automáticamente y una grabación de voz indica las extensiones a marcarse para alcanzar a los departamentos en la corporación. Asimismo, el control de horario, verifica exitosamente si la comunicación se realiza en horarios de oficina pues si no es así la llamada es terminada luego de que otro mensaje de voz indica la razón por la que el llamante ni puede ser atendido.

3.3.2.3.2 *Casos Especiales*

- Para verificar el funcionamiento de la central en algunos casos especiales, se realiza una llamada interna o desde la PSTN y se digitan patrones que no existen dentro del *dialplan*.
- Asimismo, en una llamada que ingresa a través de la troncal analógica se espera sin realizar ninguna acción luego del mensaje de voz del IVR (sin digitar ningún patrón o extensión).
- Por último se mantiene establecida una llamada realizada desde la LAN interna de forma prolongada.

Resultados.- Al generar estos pequeños escenarios especiales pero que pueden suceder, se tiene que cuando un usuario (interno o externo) digita un patrón que no corresponde a ninguna extensión creada dentro del plan de marcado, un mensaje de voz indica el error y el IVR vuelve a detallar las indicaciones de direccionamiento. Asimismo, si al recibir las instrucciones, el llamante no digita ningún patrón, la llamada se transfiere automáticamente (luego de pocos segundos) a la operadora. Finalmente, la llamada cuya duración es prolongada, el

momento que llega al límite definido mediante la aplicación `AbsoluteTimeOut( )`, un mensaje de alerta se reproduce y finalmente se cuelga la llamada.

3.3.2.3.3 *Conferencias*

- Se realizan varias llamadas a la extensión 600 para establecer una conferencia entre las áreas de Gerencia, Ventas, Pagos y Soporte.

Resultados.- La conferencia se establece satisfactoriamente hasta el momento en el que el representante del área de Soporte quiere ingresar. En ese instante recibe un mensaje de que la conferencia está llena. Esto comprueba el funcionamiento de la limitación de participantes de la sala en la extensión 600.

3.3.2.3.4 *Parqueo y Transferencia de Llamadas*

Se realiza una llamada desde el cliente externo (a través de la línea analógica) hacia la extensión del departamento de Ventas, pero resulta que la llamada estaba en realidad dirigida al departamento de Pagos. Entonces el representante de Ventas tiene dos opciones: la primera que transfiera la llamada a la extensión correspondiente a Pagos o que la parquee en la extensión 700 y se comunique con Pagos para que su representante atienda la llamada marcando a esa extensión.

Resultados.- La solución más directa resulta transferir la llamada a la extensión de pagos y al hacerse, inmediatamente empieza a timbrar la extensión correspondiente. La llamada parqueada resulta de mayor utilidad cuando el objetivo es, por ejemplo, mantener la llamada en espera para luego retomarla en otro terminal. De todos modos la transferencia y el parqueo se ejecutan sin problemas.

3.3.2.3.5 *Servicio de Directorio*

- Para verificar el servicio de directorio, simplemente se realiza una llamada a la *troncal* y se marca la extensión 501 donde se alberga este servicio.

Resultados.- Una vez que se estableció comunicación con la extensión 501, un mensaje de voz indica que digitemos las 3 primeras letras del nombre de la persona con la que se desea establecer la llamada. El servidor reconocerá los patrones digitados y un nuevo mensaje de vos pregunta si deseamos comunicarnos con una persona cuyo registro concuerda con las letras que se marcó. Finalmente si el nombre es correcto la llamada es dirigida automáticamente y sin que se necesite saber la extensión de su departamento. La prueba fue satisfactoria.

3.3.2.3.6 *Buzón de Voz, Envío a correo electrónico, configuraciones*

- Para la comprobación del buzón de voz se hace una llamada a una extensión cuyo dispositivo no esté registrado, en este caso el departamento de educación. También se hace una llamada desde Gerencia hacia Ventas pero no se contesta la llamada.

Resultados.- En ambos casos la llamada fue redirigida al buzón en dónde se grabó un mensaje. Luego se accedió desde la extensión de Ventas a la extensión 500 para escuchar los mensajes de voz que, en efecto, allí permanecían. Asimismo, el archivo de voz correspondiente al mensaje de voz se envió como archivo adjunto a la dirección de correo electrónico que se configuró el servidor *Asterisk*.

3.3.2.3.7 *Recepción de Fax*

- Se solicitó a una persona en otro edificio que envíe un fax marcando 300 una vez conectada con el IVR, para obtener tono de fax.

Resultados.- El fax se recibió digitalizado en la cuenta de correo configurada para ese efecto en el servidor *Asterisk.*

3.3.2.3.8 *Restricciones de acceso*

- Para verificar las restricciones de llamada que posee cada uno de los departamentos se realizaron algunas pruebas:

- Llamada desde gerencia a un número en Ibarra.
  Resultado: Establecida
- Llamada desde Gerencia a un teléfono celular.
  Resultado: Establecida
- Llamada desde Gerencia a un teléfono local.
  Resultado: Establecida
- Llamada desde Pagos a un teléfono en Ibarra.
  Resultado: No establecida
- Llamada desde Soporte a un teléfono celular.
  Resultado: No establecida
- Llamada desde la Operadora a un número celular.
  Resultado: Establecida
- Llamada desde Ventas a un número de emergencia.
  Resultado: Establecida

3.3.2.3.9 *Emisión de registro de llamadas*

- Luego de todas las pruebas mencionadas se verificó un *log* que mantiene *Asterisk* de todas y cada una de las llamadas que se realizaron a través de este servidor. Entre los detalles que se pueden apreciar son *origen, destino, hora de inicio, hora de fin, etc*. Este archivo se encuentra en `/var/log/asterisk/cdr-csv/Master.csv` y es un archivo delimitado por comas que puede observarse claramente a través de cualquier hoja de cálculo.

3.3.3 DEFINICIÓN DEL AMBIENTE DE PRUEBAS – PROTOTIPO DE FIREWALL

Para realizar las pruebas de firewall se creará un escenario similar al mostrado en la figura 3.2.

Fundamentalmente se generará tráfico a través de 2 dispositivos de *firewall* y se verificará que las aplicaciones de red requeridas estén disponibles, de acuerdo a las políticas definidas inicialmente. Esta labor también se ejecutará comentando las reglas específicas y comprobando mediante esto que son efectivas y las únicas que permiten un tráfico determinado.

En la oficina donde se realizan las pruebas se dispone de un cable-módem de SATNET, a través del cual se puede obtener dos direcciones IP públicas para acceder a Internet. En cada interfaz de conexión del módem se conecta una estación de trabajo (cada una con dos salidas de red Ethernet), que cumplirán la función de *firewall-NAT* y *proxy.* Ambas estaciones están configuradas como dispositivos de firewall de la forma que se indicó en el capítulo anterior, es decir, con el *script* ***firewall*** iniciado.

Como se puede apreciar en la figura 3.2, las dos estaciones (*routers*) deberán permitir el enrutamiento de tráfico entre las redes internas que estarán conectadas a ellas, ya sea de servicio web, correo electrónico, VPN, FTP, etc. Los servicios web, de correo electrónico, FTP, VPN, SSH, deberán estar disponibles desde el Internet y, por tanto, se debe comprobar que se permita este flujo a través del *firewall*. Esto es especialmente importante pues se establece una política restrictiva que descarta cualquier tipo de flujo que no esté explícitamente permitido y el mayor inconveniente que se tendrá habilitando un servicio, será porque no se ha abierto el puerto o puertos través de los que ese servicio envía información.

En la tabla 3.3 se puede observar un registro del direccionamiento que se verifica también en el esquema antes mostrado.

Luego de comprobar que los servicios de comunicaciones que la corporación requiere del prototipo se pueden acceder desde la *intranet* A, y aquellos como correo electrónico, SSH, FTP, HTTP, VPN pueden accederse desde Internet (*router* B), se realiza un escaneo de los puertos en el borde de acceso de la *intranet* A hacia Internet.

TABLA 3.4 SERVICIOS QUE SE PRESTAN A INTERNET DESDE LA *INTRANET* A

Servicio	**IP/Host**
FTP	10.0.0.4/24
SSH	190.10.225.128/24
HTTP	10.0.0.4/24
Correo	10.0.0.4/24

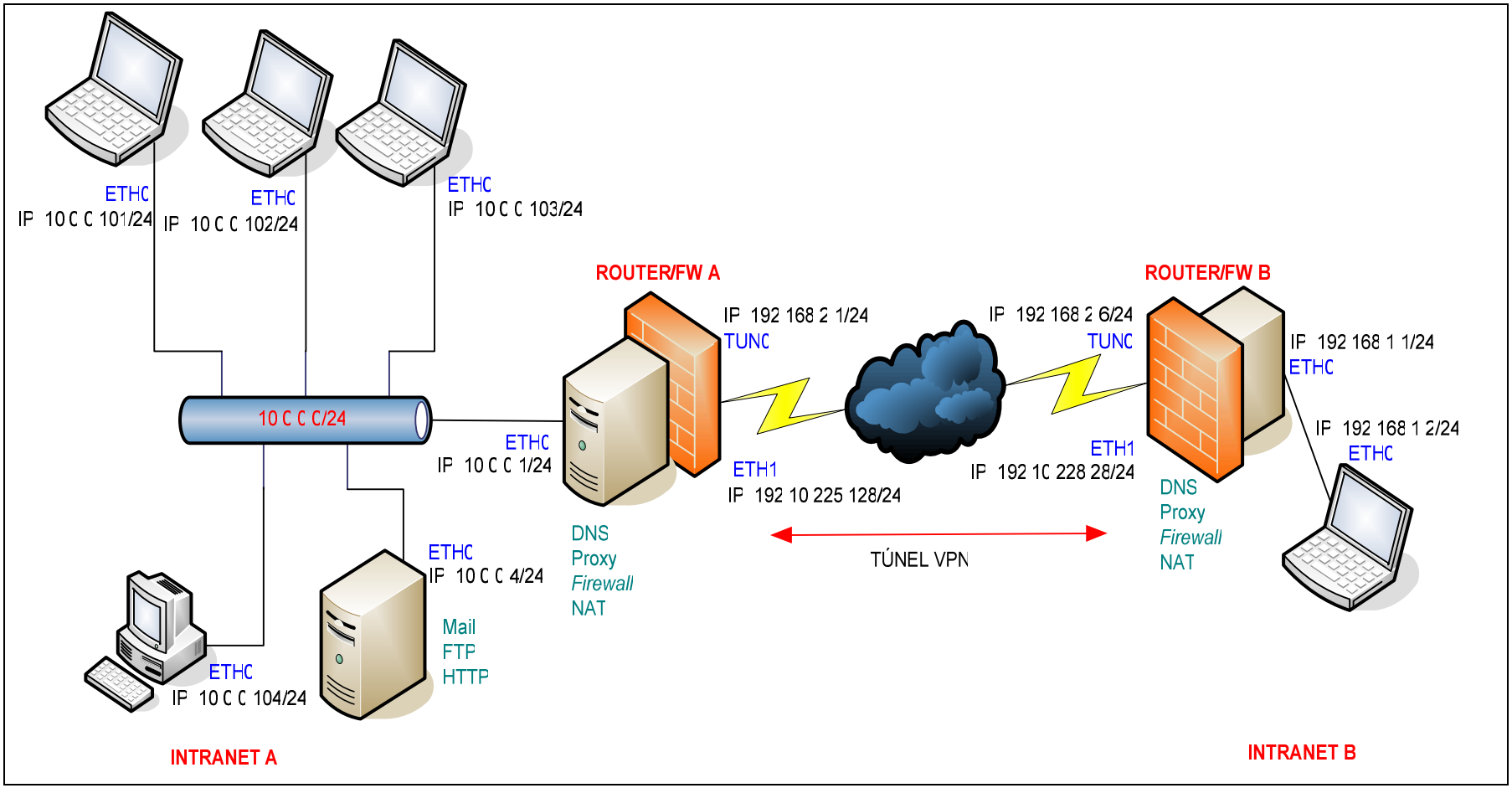


FIGURA 3.2 ESQUEMA DE RED DEL ESCENARIO DE PRUEBAS DEL PROTOTIPO DE FIREWALL

3.3.4 DESARROLLO DE PRUEBAS – PROTOTIPO DE FIREWALL

3.3.4.1 Reporte de Errores vía ICMP

Para verificar que se pueden ejecutar pruebas de conexión a dispositivos en Internet se utiliza la aplicación ***ping*** para el envío de peticiones de eco ICMP. Se hace *ping* desde los usuarios a www.google.com, a www.yahoo.com y, finalmente al otro *router* de nuestro escenario (*router* B) que dispone también de una dirección IP pública.

Resultados.- El *ping* es exitoso, lo que significa que el *firewall* enruta y permite el paso de este tipo de paquetes.

Por otro lado, las peticiones de eco dirigidas a la interfaz externa del *firewall* deben ser atendidas, se prueba esto ejecutando un *ping* desde el *Router* B del escenario al *Router* A.

Resultados.- La petición de eco es exitosa.

3.3.4.2 Navegación HTTP y HTTPS

Se verifica la navegación vía HTTP (puerto 80) y HTTPS (puerto 443) accediendo desde la red A (figura 3.2, escenario de *firewall*) a páginas como http://www.ecualug.org (puerto 80) y a páginas como https://www.gmail.com, https://www.hotmail.com (puerto 443). Para verificar las reglas específicas que permiten el paso este tipo de tráfico se comentan las reglas correspondientes en el *script* de *iptables.*

El servidor web interno se configuró para que escuche las peticiones de acceso en el puerto 8081 (en lugar del 80 estándar) pues el proveedor de Internet de la conexión que se utilizó durante las pruebas bloquea en sus *routers* el tráfico hacia puertos como el 80 y 25 en las direcciones que asigna a sus clientes residenciales. Estos cambios se reflejaron igualmente en el *script* de configuración del *firewall*. Para comprobar el funcionamiento del servicio se accedió desde el dispositivo de *firewall* de la *intranet* B (190.10.228.28/24) hacia el servidor web ubicado la *intranet* A (10.0.0.4/24) pero apuntando hacia la interfaz externa del *firewall* en la *intranet* A (190.10.225.128/24). El direccionamiento y la estructura se observa más claramente en la figura 3.2.

Resultados.- El acceso a las páginas vía HTTP y HTTPS es exitoso. El momento en que se comentan las reglas que permiten este tráfico, el acceso es bloqueado. En el caso el servidor web en la *intranet* A, el acceso es exitoso desde la *intranet* B (ver figura 3.2).

3.3.4.3 Resolución de Nombres (DNS)

El *router* A además de realizar el proceso de NAT y de servir de *firewall* de la *intranet* A es su servidor de Proxificación HTTP, servidor DHCP y de Resolución de dominios. En el caso del DNS, el *router* A básicamente reenvía las peticiones a otros servidores en Internet.

Para comprobar que este servicio está disponible para la *intranet* A, se hacen algunas consultas vía línea de comandos desde los clientes internos a través del comando **host**. Sin embargo, el instante anterior en el que se usó la aplicación *ping* a google.com o yahoo.com ya se corroboró esta resolución de nombres.

Resultados.- Las consultas DNS se ejecutan exitosamente desde los clientes de la Red A y, asimismo, al comentar las reglas que permiten este tráfico las consultas ya no tienen respuesta, lo que señala que son esas las únicas reglas que permite el flujo de peticiones DNS.

3.3.4.4 Acceso Remoto

El acceso remoto vía SSH se comprueba mediante el siguiente proceso:

- Acceso remoto desde la *intranet* a un servidor externo (**larchmont.dreamhost.com**).
- Acceso remoto desde Internet (*router B*) hacia el *router A*.

***Resultados*.-** El acceso es exitoso, sin inconvenientes. Al comentar las reglas correspondientes a estas comunicaciones, como es de esperarse, el acceso remoto y sus paquetes son descartados, tanto desde la red interna hacia Internet como desde Internet al *router A*.

3.3.4.5 Correo electrónico

El correo electrónico involucra varios servicios de los que se ejecutan las siguientes pruebas.

- Descarga y envío de correo a través del cliente POP *ThunderBird* para Linux.
- Descarga y envío de correo a través del cliente POP *Outlook*.
- Luego de solicitar, vía reclamo a la *Suptel*, la apertura del puerto 25 se abrió el tráfico y se probó el funcionamiento del servidor de correo, enviando *e-mail* vía línea de comandos desde el *firewall-proxy* de la intranet B (ver figura 3.2).

Resultados.- La descarga y envío de correo, a través de los clientes mencionados, es exitosa, así como el funcionamiento del servidor de correo en la *intranet* A.

3.3.4.6 Acceso a Servidores de Archivos (FTP)

- Primeramente se intenta establecer una conexión FTP desde un cliente Linux y uno Windows hacia un servidor FTP externo (**larchmont.dreamhost.com**).
- Desde Internet *(router B)* se realiza una conexión FTP hacia el servidor interno de la *intranet* A (ver figura 3.2).

Resultados.- Ambas conexiones FTP son exitosas. En la primera se comprueba el reenvío del tráfico FTP al exterior y en la segunda el ingreso de tráfico hacia nuestro servidor FTP. Se comentan las reglas respectivas en el *script* de *firewall* y las conexiones son descartadas, como era de esperarse.

3.3.4.7 Mensajería Instantánea

El servicio de mensajería instantánea, provisto por *Hotmail* en este caso, puede comprobarse simplemente estableciendo una conexión con el servidor, a través de la utilización de cualquier cliente como GAIM o MSN Messenger.

Resultados.- El resultado es exitoso. Si se quisiera bloquear versiones nuevas de este cliente de mensajería la labor no implica únicamente eliminar la regla que controla el puerto si no que se debe hacer un control a nivel de aplicación *(proxy)*

pues estas versiones utilizan otros métodos para conectarse y registrarse a nivel de web.

3.3.4.8 Descarga P2P

Para comprobar la regla que permite la descarga P2P por parte de los usuarios se utiliza el cliente de descarga *Limewire* desde la *intranet* para la descarga de un archivo de video.

Resultados.- La descarga tiene éxito, por otro lado, si se elimina la regla correspondiente del *script* el proceso es infructuoso.

3.3.4.9 Servicio de VPN

Desde el dispositivo con dirección IP 10.0.0.4/24 en la *intranet* A se hace *ping* a la estación con dirección IP 192.168.1.2 en la *intranet* B. Se prueba también en acceso SSH y Web directo.

Resultados.- Los resultados son todos positivos. La red 10.0.0.0/24 puede verse directamente y de forma transparente con la red 192.168.1.0/24 como si todos los dispositivos de ellas estuviesen conectados al mismo *switch*.

3.3.4.10 Voz Sobre IP

Para verificar que el *firewall* permite el paso de paquetes de VoIP, hacemos que los clientes de telefonía se registren en nuestro servidor *Asterisk* pero ahora a través de su interfaz externa. En este sentido, como ya se mencionó, el protocolo SIP presenta la gran desventaja de que para el flujo RTP emplea puertos aleatorios entre el 10000 y el 20000, de modo que el *firewall* que esté delante de dispositivos de VoIP deberá permitir el reenvío de paquetes a través de estos puertos, para permitir la comunicación con el servidor *Asterisk* a través de su interfaz en Internet, y a través de él, con otros dispositivos registrados.

La prueba consiste en hacer que el usuario **operadora** se registre en el servidor *Asterisk* pero no a través de la interfaz interna del mismo sino de la externa, de manera que todas las comunicaciones que se originen en el dispositivo de la **operadora** tengan que atravesar el *firewall,* viajar por Internet y llegar finalmente

al servidor de telefonía IP. Una vez que la **operadora** se ha registrado, se realiza una llamada desde el dispositivo del usuario externo hacia la central y cuando se activa el IVR, se redirige la llamada hacia la extensión de la operadora para que suene su dispositivo y sea contestado.

Resultados.-El registro y el establecimiento de la llamada tienen éxito, y aún más importante, el flujo de información a través de la PSTN e Internet se produce con normalidad, pues en ambos lados de la comunicación la voz se escucha normalmente.

3.3.4.11 Escaneo y *Sniffing* - HERRAMIENTAS HACKER

Estas herramientas son generalmente las básicas que utilizan los atacantes principiantes pero pueden ser de mucha utilidad para que los administradores de red detecten ciertos niveles de ataques e incluso para que puedan poner a prueba la seguridad de los sistemas que tienen a cargo.

3.3.4.11.1 Sniffers

Los sniffers son herramientas que permiten a un hacker monitorear el tráfico de red que pasa por el host y revisar el contenido de varios paquetes. Esto es útil para recuperar información, por ejemplo nombres de usuario y contraseñas.

La mayoría de sniffers que funcionan sobre Linux usan una librería llamada libcap que usualmente está disponible en la distribución. Esta librería contiene todas las rutinas de bajo nivel para colocar al adaptador de red en modo *promiscuo* de manera que pueda recibir todos los paquetes en la red.

Existen varios sniffers disponibles para Linux, de los que el más común que viene por defecto es *tcpdump*, pero no es muy sofisticado.

Hay también sniffers más sofisticados como *Ethereal* y *Sniffit* por ejemplo. Existe un sniffer mucho más sofisticado que se llama *Ethercap* que incluso permite diseccionar varios protocolos TCP que se conocen por contener contraseñas en texto plano (como Telnet, POP3, e incluso SSHv1). Este sniffer graba automáticamente e imprime estos nombres de usuario y las contraseñas.

Aunque parezca extraño, existe también software *anti-sniffer*.

3.3.4.11.2 Escáner de Puertos

Un escáner de puertos es una herramienta que se conecta a cada puerto que el usuario desea y mira los paquetes que retornan. Basado en estos paquetes, el programa puede determinar qué puerto está abierto (aceptando conexiones) y la versión del software que está siendo utilizado. Un hacker puede utilizar este conocimiento para realizar un ataque más dirigido hacia un servicio específico.

La desventaja de un escáner de puertos es que usualmente deja un log que puede ser descubierto por el administrador de sistemas.

Algunos ejemplos de escáner de puerto son: *netcat* y *Nmap.*

3.3.4.11.3 Escáner de Intrusión

El escaneo de intrusión es completamente diferente al escaneo de puertos, esencialmente porque trabaja mediante el uso de una lista de vulnerabilidades conocidas e intenta aprovecharse de ellas en el sistema objetivo, sin importar los puertos abiertos o la versión del software que se está utilizando.

El escaneo de intrusión suele dejar un rastro en la bitácora. Algunos de los escáneres de intrusión más conocidos son: *Satan, Saint* y *Nessus*, siendo este último el más sofisticado.

Una vez que se ha realizado la labor más compleja, que supone la habilitación de servicios, a través del *firewall* que maneja una política restrictiva de filtrado, se necesita emplear un mecanismo de escaneo para verificar que únicamente los puertos necesarios, y que por ello se han habilitado a través del *firewall,* estén abiertos. Este escaneo es muy útil pues además de identificar los puertos de comunicaciones abiertos, revelan algunos detalles o información sobre los servicios o el sistema operativo que está funcionando en un dispositivo ubicado en Internet. Es por esta razón que en el caso de brindar servicio de correo, HTTP, FTP u otros comunes, se debe asegurar adecuadamente también los dispositivos que ejecutan estas funciones pues, estarán al alcance de cualquiera.

La prueba de escaneo se realiza (tomando en cuenta la figura 3.2) desde un cliente en la *intranet* B, utilizando herramientas muy conocidas de “*hacking* ético” como *Nmap y Nessus.*

Las herramientas de monitoreo de paquetes, *hacking* ético y *sniffing* que se utilizaron antes y durante la etapa de pruebas son:

- *tcpdump*
- *Ethereal*
- *Logs* del sistema operativo
- *Nmap*
- *Nessus*

Resultados.- El resultado de las pruebas es satisfactorio, pues el escaneo reporta como abiertos únicamente los puertos a través de los que se prestan servicios en Internet como correo electrónico (25), FTP (20, 21), Web (8081 en el caso de la prueba) y SSH (22). Asimismo, el firewall evita que se revele información como el tipo de plataforma operativa sobre el que está funcionando, sin que se determine ningún factor de riesgo.

El resultado del escaneo utilizando las aplicaciones mencionadas puede observarse en el anexo F.

3.3.4.12 Otras Herramientas de verificación

El manejo de nuevas aplicaciones que impliquen la apertura de puertos adicionales puede llegar a ser un dolor de cabeza si no se manejan las herramientas adecuadas.

En el caso del *script* de *firewall* propuesto, los registros de *log* que generan las reglas pueden ser aprovechados a través del archivo */var/log/messages* donde el kernel de Linux detalla los eventos relacionados con paquetes descartados o rechazados por *iptables*. Es por esto que se utilizó el siguiente comando para verificar el bloqueo de algunos puertos y a través de ello el funcionamiento de varios de los servicios a través del *firewall*.

```
# tail -f /var/log/messages
```

La salida de este comando mostrará todos los paquetes que están siendo descartados o rechazados por el *firewall* además de información muy útil como la dirección IP origen y destino y también puertos origen y destino.

La salida de este comando puede además ser filtrada mediante la siguiente variación:

```
# tail -f /var/log/messages | grep [patron de filtrado]
```

El `[patron de filtrado]` puede ser una dirección IP o un puerto, de modo que la salida se refiera únicamente a los paquetes relacionados con esa dirección IP o ese puerto.

3.4 COSTOS

Determinar el presupuesto referencial del presente diseño toma en cuenta que no hay licencias de software por los módulos que se necesite y sin duda el *hardware* resulta mucho menos costoso que otras soluciones propietarias.

El hardware necesario para la implementación del sistema de voz y datos, en resumen, sería el siguiente:

TABLA 3.5 COSTOS DEL SISTEMA DE VOZ Y DATOS

Cant.	Item	Descripción	Costo Unitario	Total (USD)
2	PC HPDHC5700	HD: 80GB RAM:2GB Proc: 3 GHz CD+DVD ROM NIC:2	900	1800
1	Tarjeta Digium TDM400P	Tarjeta para 4 módulos FXO/FXS (incluye 2 FXO+2FXS)	350	350
6	Horas Técnicas	Instalación y configuración del sistema operativo y *firewall*. Instalación y configuración de la central de VoIP, habilitación de aplicaciones y servicios.	50	300

Total: USD 2450

En la generalidad de los casos, el dispositivo de *firewall* en una empresa pequeña o mediana incluirá además otras aplicaciones como servidor de correo, ftp, o web, pues el gasto de dispositivos de *hardware* individuales no se justifica en su presupuesto.

Sería posible incluir el sistema de voz y datos en un solo PC, lo que significaría un ahorro significativo a una empresa pequeña; sin embargo, es recomendable separar las funciones de modo que se distribuya la carga y no se tenga un solo punto de falla.

CAPÍTULO 4
CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La integración de software libre y telecomunicaciones (telefonía IP usando Linux), permite que los costos inherentes de hardware de telefonía se reduzcan al máximo pues es posible aprovechar la disponibilidad de procesamiento cada vez más potente y menos costoso de los computadores actuales. Esto sin duda aportará a la reducción de precios a nivel del mercado de VoIP, gracias a la competencia en costos y funcionalidades que la utilización de software libre representa.
- La típica central propietaria de hardware se convierte, gracias a la utilización de *Asterisk*, en una central de software, mucho más flexible, mejor dotada de aplicaciones cuya utilización no requiere de licencias, y más sencilla de administrar y utilizar.
- Al basarse en estándares, el sistema de voz de *Asterisk* puede interactuar con gran variedad de tecnologías de hardware (tarjetas, interfaces, adaptadores) y protocolos de comunicación y codificación.
- Al manejar un esquema no privativo, a través de código abierto, el sistema de voz puede integrarse con otros servicios como correo electrónico, bases de datos, formatos de sonido, aprovechando la amplia gama de aplicaciones de comunicación integradas a través de Internet.
- El sistema de voz está estructurado por un amplio conjunto de servicios que incluye la posibilidad de interconexión de usuarios a través de la red interna de datos, la PSTN e incluso el Internet. A esto se acompaña de varias aplicaciones como buzón de voz, mensajes de voz enviados a una dirección de correo electrónico, recepción de fax, IVR, transferencia/parqueo de llamadas, llamada en espera, servicio de directorio, conferencia, y además, gracias al manejo de contextos, permite controlar la salida de llamadas dependiendo de las facultades que se hayan asignado en el plan de marcado.
- La utilización del protocolo SIP para manejo de VoIP puede representar un pequeño riesgo de seguridad a través del *firewall* si el servidor *Asterisk* se encuentra dentro de la red interna, por su manejo aleatorio de una gran

cantidad de puertos de comunicaciones para el transporte de la voz. Sin embargo es un protocolo muy utilizado en sistemas para VoIP.

- Este sistema de voz aprovecha la infraestructura de red existente para ese tipo de tráfico, permitiendo un ahorro considerable en cableado telefónico y reduciendo el trabajo necesario para el direccionamiento y manejo controlado de las llamadas telefónicas.
- El sistema de voz en base a *Asterisk* brinda a la corporación una imagen empresarial y tecnológica que se orienta a la utilización de herramientas no privativas y cuyo desarrollo beneficia ampliamente a sociedades en desarrollo como la nuestra.
- La política restrictiva implementada en el sistema de *firewall* es la forma más segura de proteger la red de datos de la corporación, aunque eso implique una labor más ardua de administración cuando se requiera habilitar un nuevo servicio a través del cortafuegos.
- El sistema de *firewall* es capaz de filtrar tráfico a nivel de capa de enlace (capa 2), de red (capa3) y de transporte (capa 4) a través del módulo de *iptables* y a nivel de capa de aplicación gracias al sistema de proxy. En este sentido y gracias a esta capacidad es posible incluso implementar mecanismos de de priorización de tráfico que permitan controlar el ancho de banda para brindar calidad de servicio.
- Las aplicaciones de administración gráfica de *iptables* como *firestarter* o *phpfwgen* y otras distribuciones dedicadas para ello gozan de bastante popularidad por la sencillez de su configuración, pero el precio de esta sencillez es la limitada flexibilidad que ofrecen para la construcción de reglas de tráfico que se orienten a necesidades específicas.
- El sistema de *firewall* de forma general brinda protección frente a todo el tráfico que no está explícitamente permitido; sin embargo, nada podrá hacer con el flujo autorizado de paquetes que deberá ser controlado por los servicios que los manejan, como otro nivel de seguridad.
- El *script* generado cumple los requerimientos de conexión de la corporación y supera exitosamente las pruebas realizadas al evitar conexiones en puertos que no sean aquellos en los que se estén prestando servicios a Internet que por esa razón están abiertos.

- El comportamiento de las reglas del *firewall* puede monitorearse a través de los mensajes del kernel generados gracias a algunas reglas del *script,* de modo que cada vez que un paquete sea descartado, este evento será registrado en un archivo de *log* junto con información del origen de estos paquetes. Otras herramientas de *hacking* ético y escaneo son muy útiles para la verificación del funcionamiento de estas reglas de filtrado de tráfico.
- El sistema de *firewall* a través de *iptables* es sumamente flexible y el nivel de filtrado puede compararse fácilmente con el provisto por soluciones propietarias de *hardware* dedicado, como por ejemplo Cisco.
- El *firewall* es una parte del sistema de seguridad, otras medidas como la seguridad física, la administración adecuada de contraseñas, la actualización del sistema operativo y una política de uso de equipos son medidas que también deben tomarse en cuenta si se considera poseer un esquema de red asegurado.
- El costo del prototipo involucra básicamente *hardware*, no hay costos de licenciamiento de software ni limitaciones en la creación de aplicaciones adicionales o en cuanto al número de usuarios que podrán utilizarlo. Esto significa que, al representar una solución global de seguridad de datos y de voz sobre IP, el costo que involucra es sumamente menor que el que se podría pagar por una solución de hardware propietaria en la que usualmente cada módulo (aplicación) tiene su costo individual y mientras más usuarios, más elevado será su costo.

4.2 RECOMENDACIONES

- Hay una gran cantidad de distribuciones de Linux en Internet, que se han creado para instalar todo el *software* necesario para el funcionamiento de *Asterisk* y una interfaz gráfica que puede hacer de la administración y configuración del sistema una tarea muy intuitiva y sencilla. Es recomendable probar su funcionamiento aún cuando, de entrada puede esperarse algunas limitaciones en cuanto a la flexibilidad de que goza la manipulación vía archivos de configuración y líneas de comando. A pesar de esta limitación siempre será posible acceder directamente a los archivos o ejecutar los comandos vía consola.

- Es recomendable utilizar tecnologías y protocolos estándar que faciliten la instalación de dispositivos como teléfonos IP, o adaptadores ATA. En la actualidad se utiliza mucho SIP y H.323, sin embargo este último es propietario y su uso tiene un costo.
- El protocolo IAX de *Asterisk* es mucho más manejable en entornos en los que el servidor o los dispositivos cliente estén detrás de elementos que empleen NAT y mecanismos de filtrado de tráfico como un *firewall*, debido a que el tráfico de voz y señalización es enviado a través de un mismo puerto de comunicación. Aún cuando no haya una gran variedad de dispositivos que manejen este protocolo, es recomendable la migración y su estandarización pues implica además un manejo más seguro y eficiente (en el caso de troncalización) del tráfico de voz.
- En caso de usar SIP, no es recomendable que el sistema de voz acepte registros de usuarios desde Internet a través del *firewall,* pues implicaría que este último deba aceptar conexiones entrantes en gran cantidad de puertos aleatorios, lo que implica un riesgo de seguridad.
- Al escoger los dispositivos cliente no se recomienda la utilización de *software* telefónico o *softphones* pues las tarjetas de sonido de las PCs normalmente no brindan la misma calidad que los elementos destinados para la comunicación telefónica, como teléfonos analógicos o teléfonos IP.
- La solución más rentable para la corporación resulta la adquisición de adaptadores ATAs, que permiten el funcionamiento de 2 teléfonos analógicos en una red IP debido al menor costo que la opción de adquirir teléfonos IP. Si bien esta última sería la opción idónea, muchas veces el elevado costo de estos dispositivos impide que puedan ser implementados a gran escala.
- Es recomendable verificar los protocolos que manejan los dispositivos de *hardware*, especialmente los teléfonos IP y adaptadores ATA de modo que sean compatibles con los que se utilizan en el servidor. Asimismo, especialmente en el caso de los adaptadores ATA *Linksys*, verificar que el acceso a su configuración no esté bloqueado.
- En el caso de que se curse el tráfico de voz a través de Internet es imperativo que se utilicen conexiones de banda ancha pues, de otro modo, las comunicaciones se percibirán entrecortadas debido al retardo.

- Es recomendable también la implementación de un sistema que brinde alta disponibilidad y calidad de servicio, de modo que el tráfico de voz sea prioritario, especialmente si las comunicaciones se realizan a través de Internet.
- El sistema podría ser dotado de soporte para la transmisión de vídeo. Software telefónico recomendado para esto sería *Xlite* para *Windows* y *Ekiga* para Linux que tienen soporte nativo.
- Si se quiere orientar el sistema al ahorro en llamadas a la red celular, éste puede mejorarse al conectar una o varias bases celulares que funcionen como *gateways* hacia las distintas operadoras, de modo que las llamadas sean enrutadas dependiendo de la operadora a la que éstas se dirigen.
- La instalación de una interfaz gráfica también sería recomendable si se requiere que el tráfico de voz sea administrado en algún punto por una persona. Existen herramientas como *frepbx* que permiten el encolamiento, parqueo y transferencia de llamadas con un simple *drag&drop*.
- El sistema podría incluir además un *gateway* SMS, es decir, por ejemplo conectarse a él un dispositivo celular (vía *bluetooth* o serial) de modo que se envíen comunicaciones de este tipo a través del servidor de comunicaciones desde la *intranet.*
- Al implementar las herramientas, interfaces y tecnologías adecuadas como *gateways* hacia la PSTN, red celular e Internet y un sistema de tarifación, es posible implementar un sistema de locutorio a muy bajo precio y con la flexibilidad y facilidad que el código abierto proporciona.
- Una vez que se han configurado correctamente las reglas y políticas de *firewall*, es especialmente recomendable “blindar” los servicios que se prestan a través de Internet como el servidor de correo, servidor web, FTP, etc. Esto incluye actualizar el *software* que paulatinamente va corrigiendo *bugs* de seguridad. Un simple escáner de intrusión como *Nessus* podría determinar la versión de *software* y, por tanto, las posibles vulnerabilidades. En el caso del correo electrónico, la implementación de mecanismos anti-*spam* y antivirus son obligatorios.
- Al agregar reglas al *script* de *firewall* es recomendable mantener un orden adecuado en relación a cada cadena de manera que sea sencillo identificarlas

y depurar el proceso de filtrado. En la labor de depuración, el uso de *sniffers* como *Ethereal* o *tcpdump* y escáneres como *Nessus* y *nmap* son de mucha utilidad, especialmente cuando se trata de habilitar un servicio a través del *firewall*.

- Debido a que el sistema de *firewall* no es más que una parte del sistema de seguridad de una organización, es recomendable establecer políticas de gestión de usuarios, *hardware, software* y contraseñas, de manera que la seguridad de la red no sea una política escrita, sino una cultura.
- Es recomendable distribuir los servicios como correo electrónico, DNS, HTTP, *firewall, proxy*, etc, de manera que no exista un único punto de falla y se pueda repartir la carga. Lamentablemente, aún en empresas grandes es común en el país utilizar un solo dispositivo para todos los servicios, lo que representa un alto riesgo.
- La seguridad física debe tomarse muy en cuenta en la medida que crezca la organización. El cableado estructurado debe estar de acuerdo a las normas establecidas para el transporte de información y especialmente si se trata de tráfico en tiempo real como el flujo de voz. Se recomienda al menos categoría 5E en el cableado *Ethernet*. Además, detectores de humo y una ventilación acorde debe proveerse al cuarto de comunicaciones y servidores que se implemente.
- Aún cuando hay soluciones basadas en Linux que permiten el manejo de políticas de *firewall* de manera gráfica, es limitado su alcance, en comparación con la utilización de la herramienta *iptables* y sus parámetros de forma directa.
- Es recomendable el monitoreo continuo de los *logs* del sistema operativo referentes a los paquetes que están siendo rechazados por *iptables* de modo que se puedan tomar medidas cautelares o reactivas en caso de que algún intruso logre acceder al sistema. Existen herramientas de detección de intrusos que permiten el registro de eventos irregulares; entre estas herramientas están: *Snort* y *tripwire* (para el sistema de archivos). Estas aplicaciones incluso realizan reportes gráficos y pueden enviar una alarma mediante correo electrónico.
- En el caso del servidor Asterisk de VoIP, debido a que maneja un tráfico crítico para la empresa como es la voz, es recomendable utilizar un servidor dedicado

de altas prestaciones de *hardware* con capacidad de redundancia y alta disponibilidad, de modo que las comunicaciones de voz no se vean interrumpidas por un fallo de los equipos.

- En cuanto a la conexión de acceso a Internet, de acuerdo a los cálculos realizados para dos troncales, se recomienda la contratación de un canal dedicado de 512 kbps de manera que el tráfico de voz pueda cursarse adecuadamente a través de la red de redes, junto con el tráfico adicional de datos. En la medida en la que pueda utilizarse códecs más eficientes y calidad similar a G711, podrán establecerse un mayor número de troncales, de otro modo, la única solución será incrementar el tamaño del canal dedicado.

ANEXOS

ANEXO A

Archivos de Configuración – Prototipo de Central Telefónica IP PBX

Archivo de configuración `zapata.conf`

Ruta: **`/etc/asterisk/zapata.conf`**

```
#################################################################################
#########
; Configuracion del archivo zapata.conf – MachangaraSoft
; Definicion de canales analogicos
[channels]
language=es
usecallerid=yes     ;Si se usa o no identificador de llamadas
threewaycalling=yes ;Habilita tono para conferencia de llamada
transfer=yes        ;Habilita transferencia de llamada
echocancel=yes      ;Habilita cancelacion de eco
txgain=4.0          ;Ganancia en tx
rxgain=4.0          ;Ganancia en rx
faxdetect=incoming  ;Para detectar la llegada de fax
busydetect=yes      ;Detecta la señal de ocupado y cuelga la llamada
hunguponpolarityswitch=yes       ;Cuelga si cambia la polaridad de la señal
context=entrante-pstn     ;Las llamadas entrantes por el canal 1 van al
                          ;contexto [entrante-pstn] en el archivo
                          ;extensions.conf.
signalling=fxs_ks       ;Se usa señalizacion FXS en un canal FXO
channel => 1            ;El canal 1 esta conectado a la PSTN
```

Archivo de configuración `sip.conf`

Ruta: `/etc/asterisk/sip.conf`

```
################################################################################
#########
; Configuracion del Archivo SIP - MachangaraSoft

; CONTEXTO GENERAL
; Determina las opciones que se utilizaran si no se especifican explicitamente
; en la definicion de las extensiones
[general]
context=interno
port=5060           ;puerto UDP en el que se escucharan las conexiones SIP
bindaddr=0.0.0.0    ;se recibiran peticiones de registro en todas las interfaces
srvlookup=yes       ;permite que se puedan establecer llamadas SIP en base a
nombres de                    ;dominio
disallow=all        ;deshabilita todos los codecs antes configurados
;allow=ulaw         ;habilita la utilizacion de codificación con ley u
allow=all           ;habilita la utilizacion de cualquier tipo codificacion
;allow=ilbc
language=es         ;establece el lenguage de los sonidos de la contestadora
rtptimeout=60       ;termina la llamada si hay mas de 60 seg sin flujo de info
(RTP)
rtpholdtimeout=600  ;termina la llamada si hay mas de 300 seg sinflujo RTP,
cuando la
                    ;llamada esta en espera
nat=yes             ;permite la comunicacion de dispositivos que se comunican
mediante NAT
canreinvite=no      ;tipicamente en NO cuando los dispositivas se comunican via
NAT

; Se configuran las extensiones que se registraran con el servidor Asterisk
;
;EXTENSIONES SIP
;
; Extension de la Operadora
;
[operadora]
type=friend         ;puede hacer y recibir llamadas
username=operadora  ;nombre de usuario para el registro
secret=operadora    ;password del usuario
host=dynamic        ;obliga a que el usuario tenga que registrarse con el server
qualify=yes         ;permite mantener abiertas las sesiones NAT
nat=yes
canreinvite=no
context=operadora   ;contexto al que pertenece el usuario en el dialplan
disallow=all
allow=ulaw          ;permitir codec G.711 ulaw
allow=gsm           ;permitir codec GSM
allow=speex         ;permitir codec Speex
;
;Extension Area Administrativa de la Corporacion
;
[admincorp]
type=friend
username=admincorp
secret=admincorp
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=gerencia
disallow=all
allow=ulaw          ;permitir codec G.711 ulaw
allow=gsm           ;permitir codec GSM
allow=speex         ;permitir codec Speex
;
;EXTENSIONES DE LAS EMPRESAS MIEMBROS
;
;Extensiones REFUNDATION CONSULTING GROUP [Modelo para el resto de empresas]
;
[pagos_ref]
type=friend
```

```
username=pagos_ref
secret=pagos
host=dynamic_ref
qualify=yes
nat=yes
canreinvite=no
context=pagos
disallow=all
allow=ulaw          ;permitir codec G.711 ulaw
allow=gsm           ;permitir codec GSM
allow=speex         ;permitir codec Speex
;
[soporte_ref]
type=friend
username=soporte_ref
secret=soporte_ref
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=soporte
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[ventas_ref]
type=friend
username=ventas_ref
secret=ventas_ref
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=ventas
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[gerente1_ref]
type=friend
username=gerente1_ref
secret=gerente1_ref
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=gerencia
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[gerente2_ref]
type=friend
username=gerente2_ref
secret=gerente2_ref
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=gerencia
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
;Extensiones NDEVELOPER
;
[pagos_nde]
type=friend
username=pagos_ref
secret=pagos
host=dynamic_ref
qualify=yes
```

```
nat=yes
canreinvite=no
context=pagos
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[soporte_nde]
type=friend
username=soporte_nde
secret=soporte_nde
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=soporte
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[ventas_nde]
type=friend
username=ventas_nde
secret=ventas_nde
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=ventas
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[gerente1_nde]
type=friend
username=gerente1_nde
secret=gerente1_nde
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=gerencia
disallow=all
allow=ulaw
allow=gsm
allow=speex
;
[gerente2_nde]
type=friend
username=gerente2_nde
secret=gerente2_nde
host=dynamic
qualify=yes
nat=yes
canreinvite=no
context=gerencia
disallow=all
allow=ulaw
allow=gsm
allow=speex

[cliente]
type=friend
username=cliente
secret=cliente
host=dynamic
qualify=yes
nat=yes
canreinvite=yes
context=clientes
allow=all
```

Archivo de configuración extensions.conf

Ruta: **/etc/asterisk/extensions.conf**

```
##########################################################################
#########
; Configuracion del Plan de Marcado – MachangaraSoft
;
; CONTEXTO GENERAL
; Define variables generales que se usaran en todo el plan de marcado a menos que
en cada
; contexto individual se especifiquen otros valores
[general]
static=yes          ;Permite que el archivo se recargue cada vez que se reinicia
asterisk
writeprotect=no     ;Se complementa con la opcion anterior
autofallthrough=yes ;Cuelga la extensión cuando se ejecutan acciones no
contempladas
clearglobalvars=no  ;No resetea las variables globales luego de un reload
priorityjumping=no  ;Evita el salto entre prioridades que utilizan algunas
aplicaciones
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; CONTEXTO GLOBALS
; Incluye variables y opciones globales a las que se
; puede hacer referencia dentro del dialplan
;
[globals]
```

```
TRONCAL_ANDINATEL1=Zap/1  ;Identifica la interfaz de salida a la PSTN de
Andinatel
OPER=SIP/operadora        ;Identifica la extension de la operadora (IAX2/0)
ADMINCORP=SIP/admincorp   ;Identifica la extension del Administrador de la
Corporacion
;
;-------Extensiones Refundation----------
VENTAS_REF=SIP/ventas_ref
SOPORTE_REF=SIP/soporte_ref
PAGOS_REF=SIP/pagos_ref
GERENTE1_REF=SIP/gerente1_ref
GERENTE2_REF=SIP/gerente2_ref
;
;-------Extensiones NDeveloper-----------
VENTAS_NDE=SIP/ventas_nde
SOPORTE_NDE=SIP/soporte_nde
PAGOS_NDE=SIP/pagos_nde
GERENTE1_NDE=SIP/gerente1_nde
GERENTE2_NDE=SIP/gerente2_nde
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;
; CONTEXTO DEFAULT
; Incluye las capacidades por defecto que debe tener cualquier extension
; En este caso unicamente tendran por defecto la salida a llamadas de emergencia
;
[default]
exten => 911,1,Dial(${TRONCAL_ANDINATEL1}/911)  ;Si se marca la extension 911 o
9911 se                                                  ;comunica con
el servicio de emergencia
exten => 9911,1,Dial(${TRONCAL_ANDINATEL1}/911)
exten => 101,1,Dial(${TRONCAL_ANDINATEL1}/911)  ;Si se marca la extension 101 o
9101 se                                                  ;comunica con
el servicio de emergencia
exten => 9101,1,Dial(${TRONCAL_ANDINATEL1}/911)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;
; CONTEXTO ENTRANTE-PSTN
; Incluye las extensiones a las que tendran acceso las llamadas entrantes a
traves de la
; PSTN
; el IVR incluyendo informacion de direccionamiento de las extensiones

[entrante-pstn]
;Control de llamadas fuera  de horario de atencion (lunes a viernes)
exten => s,1,GotoIfTime(08:00-16:59|mon-fri|*|*?abierto)
;Control de llamadas fuera de horario de atencion (dias sabado)
exten => s,n,GotoIfTime(09:00-11:59|sat|*|*?abierto)
;Mensaje de informacion horario de atencion
exten => s,n,Playback(mens_horario_mach_es)
;Se cuelga la llamada si esta fuera del horario de atencion
exten => s,n,Hangup()
;Se contesta la llamada
exten => s,n(abierto),Answer()
;Se espera 1 segundo
exten => s,n,Wait(1)
;Se establece el tiempo máximo de llamada
;exten => s,n,Set(TIMEOUT(absolute)=30)
;Se establece 5 segundos como limite de tiempo para el marcado de una extension
luego de lo ;cual los digitos marcados se evaluaran como extension marcada
exten => s,n,Set(TIMEOUT(digit)=5)
;Se establece 15 segundos como limite para que quien llame pueda escoger una
opcion y ;marcarla antes de que la llamada se trasfiera a la extension de timeout
luego de lo cual la ;llamada se colgara
exten => s,n,Set(TIMEOUT(response)=15)
;Se reproduce el mensaje de bienvenida (español/ingles)
;exten => s,n(bienvenida),Background(mens_bienvenida_mach)
;Se reproduce el mensaje de direccionamiento de extensiones (español)
exten => s,n(reiniciar),Background(mens_ivr_mach_es)
;Se espera por el marcado de la opcion (con musica en espera)
exten => s,n,WaitExten(15|m)
;
;Si se marca 0, se transfiere la llamada a la extension de la operadora.
;Si la extension esta ocupada o nadie contesta, se va a un buzon de voz
exten => 0,1,Macro(voicemail,${OPER})
       ;
```

```
;Extensiones Generales
;Si presiona 1 se redirige la llamada a la Administración de la corp.
exten => 1,1,Macro(voicemail,${ADMINCORP})

;Si presiona 2, se reproducen indicaciones para comunicarse con una empresa
miembro
exten => 2,1,Background(mens_empresas_mach_es)
exten => 2,n,WaitExten(10|m)
;Mensaje "Se ha comunicado con Refundation ...."
exten => 201,1,Background(mens_bienvenida_ref_mach_es)
;Mensaje de direccionamiento de ext. en las empresas
exten => 201,n,Background(mens_ivr_deptosref_mach_es)

exten => 201,n,WaitExten(10|m)
exten => 300,1,Dial(IAX2/300)

;Extensiones de las Empresas
;REFUNDATION
exten => 301,1,Macro(voicemail,${VENTAS_REF})
exten => 302,1,Macro(voicemail,${SOPORTE_REF})
exten => 303,1,Macro(voicemail,${EDUCACION_REF})
exten => 304,1,Macro(voicemail,${PAGOS_REF})
exten => 305,1,Macro(voicemail,${GERENTE1_REF})
exten => 306,1,Macro(voicemail,${GERENTE2_REF})
;
;NDEVELOPER
;exten => 311,1,Macro(voicemail,${VENTAS_NDE})
;exten => 322,1,Macro(voicemail,${SOPORTE_NDE})
;exten => 313,1,Macro(voicemail,${EDUCACION_NDE})
;exten => 314,1,Macro(voicemail,${PAGOS_NDE})
;exten => 315,1,Macro(voicemail,${GERENTE1_NDE})
;exten => 316,1,Macro(voicemail,${GERENTE2_NDE})

;Extensiones Especiales VOICEMAIL DIRECTORIO INVALID TIMEOUT
;Extensión que permite manejar el correo de voz de sus usuarios
exten => 500,1,VoiceMailMain()
;Extension para el acceso a traves del directorio

exten => 501,1,Directory(default,entrante-pstn,f)
;CONFERENCE ROOM 1-------
;Se limita el espacio de conferencia a 3 participantes
exten => 600,1,MeetMeCount(600,CONFCOUNT)
exten => 600,2,GotoIf($[${CONFCOUNT} <= 2]?3:100)
exten => 600,3,MeetMe(600,ip,54321)
exten => 600,100,Playback(conf-full)
;
;CONFERENCE ROOM 2-------
; Se limita el espacio de conferencia a 4 participantes
exten => 601,1,MeetMeCount(601,CONFCOUNT)
exten => 601,2,GotoIf($[${CONFCOUNT} <= 2]?3:100)
exten => 601,3,MeetMe(601,ip,54321)
exten => 601,100,Playback(conf-full)

;Extensiones INVALID y TIMEOUT
;Se reproduce un mensaje de invalidez si se presiono una combiancion incorrecta
exten => i,1,Playback(mens_ext_invalida)
;Se regresa a la seccion de bienvenida
exten => i,2,Goto(s,reiniciar)
;Se reproduce el mensaje de agradecimiento si la espera por un comando fue
demasiado larga ;se cuelga la llamada
exten => t,1,Playback(mens_gracias_mach_es)
exten => t,2,Hangup()
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;
;CONTEXTO INTERNO
; Incluye las extensiones y el proceso de comunicacion interno entre las empresas
que estan dentro de la LAN
; Contendra todas las extensiones de la empresa (incluyendo algunas que no pueden
alcanzarse ; directamente desde otros contextos) como las correspondientes a los
gerentes.
[interno]
;Se incluye el contexto de manejo de llamadas por defecto
include => default
;Se incluye el contexto de parqueo de llamadas (ubicado en features.conf)
```

```
include => parkedcalls
;Se llama al macro voicemail para el manejo de la extensión de la operadora
exten => 0,1,Macro(voicemail,${OPER})
exten => 1,1,Macro(voicemail,${ADMINCORP})
;
;Extensiones Empresas
;
;REFUNDATION
exten => 301,1,Macro(voicemail,${VENTAS_REF})
exten => 302,1,Macro(voicemail,${SOPORTE_REF})
exten => 303,1,Macro(voicemail,${EDUCACION_REF})
exten => 304,1,Macro(voicemail,${PAGOS_REF})
exten => 305,1,Macro(voicemail,${GERENTE1_REF})
exten => 306,1,Macro(voicemail,${GERENTE2_REF})
;
;NDEVELOPER
;exten => 311,1,Macro(voicemail,${VENTAS_NDE})
;exten => 322,1,Macro(voicemail,${SOPORTE_NDE})
;exten => 313,1,Macro(voicemail,${EDUCACION_NDE})
;exten => 314,1,Macro(voicemail,${PAGOS_NDE})
;exten => 315,1,Macro(voicemail,${GERENTE1_NDE})
;exten => 316,1,Macro(voicemail,${GERENTE2_NDE})

;Extensiones Especiales VOICEMAIL DIRECTORIO CONFERENCIA
;
;Extensión de acceso y manejo del correo de voz
exten => 500,1,VoiceMailMain()
;Extension para el acceso a traves del directorio
exten => 501,1,Directory(default,entrante-pstn,f)
;
;CONFERENCE ROM 1
;
exten => 600,1,MeetMeCount(600,CONFCOUNT)
;Se limita el conference room a 3 participantes
exten => 600,2,GotoIf($[${CONFCOUNT} <= 3]?3:100)
exten => 600,3,MeetMe(600,ip,54321)
exten => 600,100,Playback(conf-full)

;CONFERENCE ROM 2
;
exten => 600,1,MeetMeCount(600,CONFCOUNT)
exten => 600,2,GotoIf($[${CONFCOUNT} <= 3]?3:100)
exten => 600,3,MeetMe(600,ip,54321)
exten => 600,100,Playback(conf-full)
;Si la espera por un patron de marcado ha tomado mucho tiempo, se agradece y se
cuelga la ;llamada
exten => t,1,Playback(mens_gracias_mach_es)
exten => t,2,Hangup()
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;
;
; CONTEXTOS de manejo de DialPlan
;
; CONTEXTO LOCAL
;
; Define el acceso hacia la PSTN mediante llamadas locales (dentro de la
provincia)
; Se restringe el acceso a otro tipo de trafico mediante los patrones de marcado
;
[local]
ignorepat => 9            ; Se ignora el patron inicial 9 para que el tono no se
altere
include => default        ; Se incluye el contexto por defecto
include => interno        ; Se incluye el contexto default
;Salida localexten => _9[23]XXXXXX,1,Dial(${TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _9[23]XXXXXX,n,Congestion()
exten => _9[23]XXXXXX,102,Congestion()
;Salida a numeros gratuitos
exten => _91800XXXXXX,1,Dial(${TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _91800XXXXXX,n,Congestion()
exten => _91800XXXXXX,102,Congestion()
;
; CONTEXTO REGIONAL
;
```

```
; Define el acceso hacia la PSTN mediante llamadas regionales (a otras
provincias)
;
[regional]
ignorepat => 9
include => default
include => interno
include => local

exten => _90[3-7][23]XXXXXX,1,Dial(${TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _90[3-7][23]XXXXXX,n,Congestion()
exten => _90[3-7][23]XXXXXX,102,Congestion()

; CONTEXTO INTERNACIONAL
;
; Define el acceso hacia la PSTN mediante llamadas a otros países
; Este contexto deberá asignarse con mucho cuidado pues puede permitir el acceso
a cualquier tipo de llamadas
;
[internacional]
ignorepat => 9
include => default
include => interno
include => local

exten => _900.,1,Dial(${TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _900.,n,Congestion()
exten => _900.,102,Congestion()
;
; CONTEXTO CELULAR
;Define el acceso hacia dispositivos celulares mediante la linea telefonica de la
PSTN
;
[celular]
ignorepat => 9
include => default
include => interno
include => local

exten => _90[89]XXXXXXX,1,Dial(${TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _90[89]XXXXXXX,n,Congestion()
exten => _90[89]XXXXXXX,102,Congestion()

[comerciales]
exten => _91[79]00XXXXXX,1,Dial($TRONCAL_ANDINATEL1}/${EXTEN:1})
exten => _91[79]00XXXXXX,n,Congestion()
exten => _91[79]00XXXXXX,102,Congestion()

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;
; DEFINICION DE CONTEXTOS ESPECIFICOS
; En este espacio se detallan contextos mas especificos, relacionados con la
estructura de
; la orgnizacion
; Una vez que se han definido contextos generales como los anteriores, es mas
sencillo
; conformar estos contextos, incluyendo unicamente los anteriores.
;
; CONTEXTO GERENCIA
; Este contexto representa a los "privilegios" a nivel de comunicacion telefonica
tienen los ; gerentes
; De manera comun y en honor a su jerarquia, estos gozan de acceso total, de modo
que es
; coherente incluir aqui a todos los contextos antes definidos.
;
[gerencia]
include => default
include => regional
include => internacional
include => celular
include => comerciales
;
; CONTEXTO OPERADORA
; Las operadoras, dentro de su funcion de intercomunicacion de llamadas necesitan
igualmente tener
; acceso a todos los contextos.
```

```
;
[operadora]
include => default
include => regional
include => internacional
include => celular
include => comerciales
;
; CONTEXTO SOPORTE
; En general el area de Soporte recibira llamadas, no las hara, debe sin embargo
incluirse
; el contexto default para llamadas de emergencia y quizá el local.
;
[soporte]
include => default
include => local
;
; CONTEXTO VENTAS
; El departamento de ventas deberia tener acceso a llamadas a nivel local,
regional e
; incluso quiza internacional
;
[ventas]
include => default
include => regional
;
; CONTEXTO PAGOS
; El departamento de pagos generalmente recibira llamadas de los proveedores.
;
[pagos]
include => default
include => local

; CONTEXTO PARA LOS CLIENTES
; Este contexto podra contactarse unicamente con la parte interna de la
corporacion
[clientes]
include => interno
;
;DEFINICION DE MACROS
;
; MACRO VOICEMAIL.- Define una subrutina que se encarga de marcar a una extension
y enviarla ; al correo de voz en caso de que la extension este ocupada o que no
este disponible.
[macro-voicemail]
exten => s,1,Dial(${ARG1},15,tT)
exten => s,2,VoiceMail(u${MACRO_EXTEN}@default)
exten => s,102,VoiceMail(b${MACRO_EXTEN}@default)
```

Archivo de configuración `voicemail.conf`

Ruta: **`/etc/asterisk/voicemail.conf`**

```
#############################################################################
#########
; Configuracion del Archivo voicemail.conf – MachangaraSoft
[general]
servermail=asterisk@refundation.ec      ;Se determina el e-mail del servidor de
VoIP
attach=yes          ;Se activa la opción de enviar los mensajes de buzon de voz
en un                    ;mensaje de correo electronico, como archivo de sonido
adjunto
maxmsg=100          ;Se define el maximo numero de mensajes de voz por carpeta
maxmessage=180      ;Maxima duracion del mensaje de voz, medido en segundos
minmessage=3        ;Tiempo minimo de mensaje de voz para ser considerado como
tal
maxsilence=10       ;Numero de segundos de silencio antes de dar fin a la
grabación
maxlogins=3         ;Numero maximo de intentos fallidos de acceder a la cuenta de
buzon
directoryintro=mens_conf_llena_es       ;archivo de sonido que indica que una sala
de                                            ;conferencia esta llena

;Contexto default de buzon de voz
[default]
0 => 0,Operadora
1 => 1,Hernando Lopez
;
;Voicemail Refundation
301 => 301,Paola Bolanos                                    ;ventas
302 => 302,Jose Estrada,jestrada@gateway.refundation.ec ;soporte
303 => 303,Diego Pullas                                     ;educacion
304 => 304,Paola Pullas                                     ;pagos
305 => 305,Christian Pazmino                                ;gerente1

;Voicemail Ndeveloper
;Voicemail SoporteLibre
;Voicemail MagmaSoft
;Voicemail SanteFe
;Voicemail DreamQuest
;Voicemail Decuador
;Voicemail LogicStudio
;Voicemail NeoQuality
```

Puede observarse que para que cada cliente de *Asterisk* pueda obtener servicio de buzón de voz, la extensió correspondiente debe estar definida en este archivo y junto a ella una clave numérica y, si se desea que el mensaje de voz sea enviado mediante correo electrónico, se deberá definir la dirección de correo a la que se deberá ser enviado.
Esta por demás mencionar que el servidor de correo de la estación deberá estar correctamente configurado de manera que pueda enviar el mensaje a través de Intenet.

Archivo de configuración `features.conf`

Ruta: **`/etc/asterisk/features.conf`**

```
#############################################################################
; Configuracion archivo features.conf – MachangaraSoft
[general]
parkext => 700                  ;Extension a marcar para parquar una llamada
parkpos => 701-720              ;Extension donde se parquearan las llamadas
```

```
                                ;y debe ser un valor numerico
context => parkedcalls          ;Contexto al que pertenecen las llamadas
                                ;parqueadas
```

Se puede observar que este archivo es importante para la configuración del servicio de parqueo de llamadas para el sistema IP PBX *Asterisk*.

ANEXO B

Complementos – Prototipo de Central IP PBX MachangaraSoft

Aplicaciones del *dialplan* de A*sterisk*

Listado obtenido a partir de la secuencia de comandos:

```
[root@gateway ~]# asterisk -r
Asterisk 1.2.13, Copyright (C) 1999 - 2006 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'show warranty' for details.
This is free software, with components licensed under the GNU General Public
```

```
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'show license' for details.
=========================================================================
Connected to Asterisk 1.2.13 currently running on gateway (pid = 31911)
Verbosity is at least 3
gateway*CLI> show applications
```

El listado de todas las aplicaciones instaladas en el servidor *Asterisk* para su funcionamiento con el *dialplan* son:

```
-= Registered Asterisk Applications =-
       AbsoluteTimeout: Set absolute maximum time of call
        AddQueueMember: Dynamically adds queue members
               ADSIProg: Load Asterisk ADSI Scripts into phone
    AgentCallbackLogin: Call agent callback login
            AgentLogin: Call agent login
  AgentMonitorOutgoing: Record agent's outgoing call
                   AGI: Executes an AGI compliant application
         AlarmReceiver: Provide support for receving alarm reports from a burglar
or fire alarm panel
                Answer: Answer a channel if ringing
    AppendCDRUserField: Append to the CDR user field
          Authenticate: Authenticate a user
            BackGround: Play a file while awaiting extension
      BackgroundDetect: Background a file with talk detect
                  Busy: Indicate the Busy condition
         ChangeMonitor: Change monitoring filename of a channel
           ChanIsAvail: Check channel availability
               ChanSpy: Listen to the audio of an active channel

            CheckGroup: Check the channel count of a group against a limit
            Congestion: Indicate the Congestion condition
       ControlPlayback: Play a file with fast forward and rewind
                  Curl: Load an external URL
                   Cut: Splits a variable's contents using the specified
delimiter
              DateTime: Says a specified time in a custom format
                 DBdel: Delete a key from the database
            DBdeltree: Delete a family or keytree from the database
                 DBget: Retrieve a value from the database
                 DBput: Store a value in the database
               DeadAGI: Executes AGI on a hungup channel
                  Dial: Place a call and connect to the current channel
              Dictate: Virtual Dictation Machine
          DigitTimeout: Set maximum timeout between digits
             Directory: Provide directory of voicemail extensions
                  DISA: DISA (Direct Inward System Access)
               DumpChan: Dump Info About The Calling Channel
           DUNDiLookup: Look up a number with DUNDi
                  EAGI: Executes an EAGI compliant application
                  Echo: Echo audio read back to the user
              EndWhile: End A While Loop
            EnumLookup: Lookup number in ENUM
                  Eval: Evaluates a string
                  Exec: Executes internal application
                ExecIf: Conditional exec
            ExecIfTime: Conditional application execution based on the current
time
           ExternalIVR: Interfaces with an external IVR application
              Festival: Say text to the user
                 Flash: Flashes a Zap Trunk
               ForkCDR: Forks the Call Data Record
              GetCPEID: Get ADSI CPE ID
         GetGroupCount: Get the channel count of a group
    GetGroupMatchCount: Get the channel count of all groups that match a pattern
                 Gosub: Jump to label, saving return address
               GosubIf: Jump to label, saving return address
                  Goto: Jump to a particular priority, extension, or context
                GotoIf: Conditional goto
            GotoIfTime: Conditional Goto based on the current time
                Hangup: Hang up the calling channel
       HasNewVoicemail: Conditionally branches to priority + 101 with the right
options set
```

```
          HasVoicemail: Conditionally branches to priority + 101 with the right
options set
         IAX2Provision: Provision a calling IAXy with a given template
                  ICES: Encode and stream using 'ices'
             ImportVar: Import a variable from a channel into a new variable
       LookupBlacklist: Look up Caller*ID name/number from blacklist database
         LookupCIDName: Look up CallerID Name from local database
                 Macro: Macro Implementation
              MacroExit: Exit From Macro
                MacroIf: Conditional Macro Implementation
          MailboxExists: Check to see if Voicemail mailbox exists
                  Math: Performs Mathematical Functions
                   MD5: Calculate MD5 checksum
               MD5Check: Check MD5 checksum
                 MeetMe: MeetMe conference bridge
            MeetMeAdmin: MeetMe conference Administration
            MeetMeCount: MeetMe participant count
              Milliwatt: Generate a Constant 1000Hz tone at 0dbm (mu-law)
             MixMonitor: Record a call and mix the audio during the recording
                Monitor: Monitor a channel
              MP3Player: Play an MP3 file or stream
            MusicOnHold: Play Music On Hold indefinitely
                  MYSQL: Do several mySQLy things
                NBScat: Play an NBS local stream
                  NoCDR: Tell Asterisk to not maintain a CDR for the current call
                   NoOp: Do Nothing
                   Page: Pages phones
                   Park: Park yourself
       ParkAndAnnounce: Park and Announce
            ParkedCall: Answer a parked call
      PauseQueueMember: Pauses a queue member
                Pickup: Directed Call Pickup
              Playback: Play a file
             PlayTones: Play a tone list
        PrivacyManager: Require phone number to be entered, if no CallerID sent
              Progress: Indicate progress
                 Queue: Queue a call for a call queue
                Random: Conditionally branches, based upon a probability
                  Read: Read a variable
              ReadFile: ReadFile(varname=file,length)
              RealTime: Realtime Data Lookup
        RealTimeUpdate: Realtime Data Rewrite
                Record: Record to a file
     RemoveQueueMember: Dynamically removes queue members
              ResetCDR: Resets the Call Data Record
       ResponseTimeout: Set maximum timeout awaiting response
             RetryDial: Place a call, retrying on failure allowing optional exit
extension.
                Return: Return from gosub routine
               Ringing: Indicate ringing tone
              SayAlpha: Say Alpha
            SayCountPL: Say the counting word the fits to a number
             SayDigits: Say Digits
             SayNumber: Say Number
           SayPhonetic: Say Phonetic
           SayUnixTime: Says a specified time in a custom format
              SendDTMF: Sends arbitrary DTMF digits
             SendImage: Send an image file
              SendText: Send a Text Message
               SendURL: Send a URL
                   Set: Set channel variable(s) or function value(s)
            SetAccount: Set the CDR Account Code
          SetAMAFlags: Set the AMA Flags
          SetCallerID: Set CallerID
        SetCallerPres: Set CallerID Presentation
       SetCDRUserField: Set the CDR user field
            SetCIDName: Set CallerID Name
             SetCIDNum: Set CallerID Number
          SetGlobalVar: Set a global variable to a given value
              SetGroup: Set the channel's group
           SetLanguage: Set the channel's preferred language
        SetMusicOnHold: Set default Music On Hold class
               SetRDNIS: Set RDNIS Number
  SetTransferCapability: Set ISDN Transfer Capability
                SetVar: Set channel variable(s)
         SIPAddHeader: Add a SIP header to the outbound call
```

```
           SIPDtmfMode: Change the dtmfmode for a SIP call
          SIPGetHeader: Get a SIP header from an incoming call
                   SMS: Communicates with SMS service centres and SMS capable
analogue phones
            SoftHangup: Soft Hangup Application
                  Sort: Sorts a list of keywords and values
              StackPop: Remove one address from gosub stack
      StartMusicOnHold: Play Music On Hold
           StopMonitor: Stop monitoring a channel
       StopMusicOnHold: Stop Playing Music On Hold
         StopPlayTones: Stop playing a tone list
                System: Execute a system command
            TestClient: Execute Interface Test Client
            TestServer: Execute Interface Test Server
              Transfer: Transfer caller to remote extension
             TrySystem: Try executing a system command
            TXTCIDName: Lookup caller name from TXT record
    UnpauseQueueMember: Unpauses a queue member
             UserEvent: Send an arbitrary event to the manager interface
               Verbose: Send arbitrary text to verbose output
        VMAuthenticate: Authenticate with Voicemail passwords
             VoiceMail: Leave a Voicemail message
         VoiceMailMain: Check Voicemail messages
                  Wait: Waits for some time
             WaitExten: Waits for an extension to be entered
           WaitForRing: Wait for Ring Application
        WaitForSilence: Waits for a specified amount of silence
       WaitMusicOnHold: Wait, playing Music On Hold
                 While: Start A While Loop
            Zapateller: Block telemarketers with SIT
              ZapBarge: Barge in (monitor) Zap channel
                ZapRAS: Executes Zaptel ISDN RAS application
               ZapScan: Scan Zap channels to monitor calls
    -= 163 Applications Registered =-
```

Se puede observar que existe una cantidad inmensa se aplicaciones de las que el servidor y específicamente en *dialplan* puede sacar provecho.
Si desea información más detallada (uso o parámetros) de alguna de estas aplicaciones, basta con ejecutar el comando:

```
gateway*CLI> show application BackGround
```

Que obtendrá la siguiente salida:

```
  -= Info about application 'BackGround' =-

[Synopsis]
Play a file while awaiting extension

[Description]
  Background(filename1[&filename2...][|options[|langoverride][|context]]):
This application will play the given list of files while waiting for an
extension to be dialed by the calling channel. To continue waiting for digits
after this application has finished playing files, the WaitExten application
should be used. The 'langoverride' option explicity specifies which language
to attempt to use for the requested sound files. If a 'context' is specified,
this is the dialplan context that this application will use when exiting to a
dialed extension.  If one of the requested sound files does not exist, call
processing will be
terminated.
  Options:
    s - causes the playback of the message to be skipped
          if the channel is not in the 'up' state (i.e. it
          hasn't been answered yet.) If this happens, the
          application will return immediately.
    n - don't answer the channel before playing the files
    m - only break if a digit hit matches a one digit
          extension in the destination context
```

Instalación de *ASTERISK*, herramienta de Central IP

Se tomará como base la versión 4.3 de la plataforma operativa CentOS para la instalación.

Requerimientos

Para que *Asterisk* pueda instalarse adecuadamente y pueda gozar de todas las funcionalidades de que dispone, deben instalarse previamente algunos paquetes. De manera que se pueda consultar si estos paquetes se encuentran instalados se utilizará la herramienta RPM, de la siguiente manera:

```
# rpm -q gcc cvs libxml2-devel libtiff-devel mysql-server php-gd php-mysql kernel-devel bison ncurses-devel audiofile-devel subversion libogg-devel zlib-devel lame
```

Una vez ejecutado el comando en una sola línea, el sistema operativo nos indicará de entre los paquetes listados, cuáles están instalados y cuáles no. Todos estos paquetes a excepción de **lame** se encuentran dentro de los medios de instalción de CentOS.
Para instalar cada uno de los paquetes se ejecutará:

```
# rpm -ihv [nombre-del-paquete-sin-extension]
```

Y para instalar el paquete **lame** bastará con digitar el comando:

```
# yum install lame
```

Instalación de componentes

Ya instalados los paquetes base, se procede a descargar e instalar el *software* y los componentes de instalación de *Asterisk*. Los paquetes son:

```
zaptel-1.2.11.tar.gz
libpri-1.2.4.tar.gz
asterisk-1.2.13.tar.gz
asterisk-sounds-1.2.1.tar.gz
asterisk-addons-1.2.5.tar.gz
```

La instalación de los *drivers* zaptel, necesarios para la interacción de *Asterisk* con los módulos analógicos sse relizará de la siguiente secuencia de comandos:

```
# cd /usr/src
# tar -zxvf zaptel-1.2.11.tar.gz
# cd zaptel-1.2.11/
# mv ztdummy.c ztdummy.c.orig
# sed "s/if 0/if 1/" < ztdummy.c.orig > ztdummy.c
# cd /usr/src/kernels/2.6.9-34.EL-i686/include/linux/
# mv spinlock.h spinlock.h.orig
#  sed "s/rw_lock_t/rwlock_t/" < spinlock.h.orig > spinlock.h
# cd /usr/src/zaptel-1.2.11/
# make linux26
# make install
# make config
# modprobe zaptel
# modprobe ztdummy
```

Ahora, el proceso de instalación de los módulos para el funcionamiento de canales PRI sería:

```
# cd /usr/src
# tar -zxvf libpri-1.2.4.tar.gz
# cd libpri-1.2.4
# make clean
# make install
```

La instalación propiamente de *Asterisk* se lleva a cabo gracias a la siguiente secuencia de comandos:

```
# cd /usr/src
# tar -zxvf asterisk-1.2.13.tar.gz
# cd asterisk-1.2.13
#  make clean
# make install
# make config
```

La instalación de los sonidos por defecto consiste en lo siguiente:

```
# cd /usr/src
# tar -zxvf asterisk-sounds-1.2.1.tar.gz
# make install
```

Finalmente, la instalación de paquetes suplementarios a través de **asterisk-addons** se realiza de la siguiente forma:

```
# tar -zxvf asterisk-addons-1.2.5.tar.gz
# cd asterisk-addons-1.2.5
#  make instal
```

Se asume que todos estos paquetes fueron descargados dentro del *path* `/usr/src`.
Finalmente se reinicia el equipo y el demonio de *Asterisk* ya se habrá iniciado. Esto se puede comprobar, accediendo a la consola de la aplicación mediante el comando **asterisk -r** y el resultado debería ser el que se muestra.

```
[root@gateway ~]# asterisk -r
Asterisk 1.2.13, Copyright (C) 1999 - 2006 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'show warranty' for
details.
This is free software, with components licensed under the GNU General
Public
License version 2 and other licenses; you are welcome to redistribute it
under
certain conditions. Type 'show license' for details.
=========================================================================
Connected to Asterisk 1.2.13 currently running on gateway (pid = 31911)
Verbosity is at least 3
gateway*CLI>
```

Configuración de *Asterisk* para Recepción de Fax por una interfaz Zaptel

Una vez que el sistema IP PBX de *Asterisk* está funcionando se le puede sacar partido en tareas comunes, como por ejemplo recibir un fax, aprovechando la conexión del servidor con la PSTN. Esto llega a ser de mucha utilidad si no se dispone de una máquina de fax para recibir documentos que otras empresas únicamente envían mediante ese medio.

El proceso que se indica a continuación permitirá que en una de las extensiones creadas en el servidor *Asterisk* se pueda dar tono de fax a algún cliente o proveedor que requiera enviar determinada información mediante este canal.
El fax llegará en forma de archivo adjunto a un correo electrónico y podría descargarse e imprimirse si fuese necesario.

Instalación y Configuración

Es necesario instalar las siguientes aplicaciones:
- hylafax
- iaxmodem

Y las siguientes dependencias:
- ghostscript
- sharutils

De todos estos paquetes, las dos dependencias pueden instalarse mediante la herramienta yum:

```
# yum install ghostscript sharutils
```

Ahora hay que descargarse **hylafax** mediante el siguiente comando:

```
# wget -c ftp://ftp.hylafax.org/binary/linux/redhat/RPMS/i386/hylafax-4.3.2-3rhel4.i386.rpm
```

Una vez descargado, en el mismo directorio donde se descargó, se ejecuta:

```
# rpm –ihv hylafax-4.3.2-3rhel4.i386.rpm
```

La extensión que recibirá en fax utilizará el protocolo IAX, de manera que esta extensión se configurará en el archivo `/etc/asterisk/iax.conf` de la siguiente manera:

```
[300]
username=300
type=friend
secret=300
qualify=yes
notransfer=yes
host=dynamic
context=from-internal
callerid="Fax" <300>
allow=all
```

Asimismo, la interfaz de conexión con la PSTN deberá detectar la llegada de un fax y esto se configura en el archivo correspondiente al módulo zaptel, ubicado en `/etc/asterisk/zaptel.conf`:

```
faxdetect=incoming
```

La extensión deberá formar parte del *dialplan* en el archivo `/etc/asterisk/extensions.conf`:

```
exten => 300,1,Dial(IAX2/300)
```

Ahora se procede a la configuración de la aplicación que manejará la recepción de fax a través del protocolo IAX.
Primero se descarga el paquete utilizando el siguiente comando:

```
# wget –c http://ufpr.dl.sourceforge.net/sourceforge/iaxmodem/iaxmodem-0.2.1.tar.gz
```

Se copia el paquete dentro del directorio `/usr/src` y se lo desempaqueta:

```
# tar -zvxf iaxmodem-0.2.1.tar.gz
```

Esto genera un nuevo archivo llamado `/usr/src/iaxmodem-0.2.1`. Dentro de ese archivo se procede a instalar el paquete, a través de la siguiente secuencia de comandos:

```
# cd lib/libiax2
# ./configure
# make && make install
# cd ../spandsp
# ./configure
# make && make install
# cd ../../
# ./configure
# make
```

Dentro del mismo directorio `/usr/src/iaxmodem-0.2.1` se copia el archivo binario iaxmodem hacia la carpeta `/usr/bin`:

```
# cp iaxmodem /usr/bin
```

Dentro de /etc/iaxmodem se crea otro directorio llamado ttyIAX que guardará la configuración del módem IAX de la siguiente manera:

```
device          /dev/ttyIAX
port            45699
refresh         300
server          localhost
peername        300
secret          300
cidname         Fax
cidnumber       300
codec           alinear
```

Ahora se verifica si el módem se registra con el *Asterisk*:

```
# iaxmodem ttyIAX
[2007-05-10 23:56:51] Modem started
[2007-05-10 23:56:51] Setting device = '/dev/ttyIAX'
[2007-05-10 23:56:51] Setting port = 45699
[2007-05-10 23:56:51] Setting refresh = 300
[2007-05-10 23:56:51] Setting server = 'localhost'
[2007-05-10 23:56:51] Setting peername = '300'
[2007-05-10 23:56:51] Setting secret = '300'
[2007-05-10 23:56:51] Setting cidname = 'Fax'
[2007-05-10 23:56:51] Setting cidnumber = '300'
[2007-05-10 23:56:51] Setting codec = slinear
[2007-05-10 23:56:51] Error: group unspecified, using root instead
[2007-05-10 23:56:51] Error:  user not found in passwd file, using root
instead
[2007-05-10 23:56:51] Error: invalid mode string () ? Leaving default
modes on /dev/ttyIAX
[2007-05-10 23:56:51] Opened pty, slave device: /dev/pts/1
[2007-05-10 23:56:51] Created /dev/ttyIAX symbolic link
[2007-05-10 23:56:51] Error: mode is 0, leaving default permissions
Restart 0
[2007-05-10 23:56:51] Registration completed successfully.
```

Para determinar si la extensión 300 se ha registrado en el *Asterisk,* hay que acceder a la consola y preguntarle:

```
gateway*CLI> iax2 show peers
Name/Username    Host                 Mask             Port         Status
300/300          127.0.0.1       (D)  255.255.255.255  45699        OK (2 ms)
```

Para que el MODEM esté disponible de forma permanente, se puede agregar la siguiente línea al final del archivo `/etc/inittab`.

```
iax:2345:respawn:/usr/bin/iaxmodem ttyIAX &> /var/log/asterisk/iaxmodem-
ttyIAX
```

Y se ejecuta el siguiente comando para que este archivo sea leído popr el kernel nuevamente:

```
# telinit q
```

Ahora es necesario configurar la aplicación **`hylafax`** para que acepte los faxes:

```
# faxsetup
```

Luego de ejecutar este comando, se iniciará un proceso que irá requiriendo información cuyos valores pueden quedar como los mostrados por defecto, a excepción de aquellos correspondientes al código de área, número de fax y permisos de los archivos.

Dentro del *path* `/usr/src/iaxmodem-0.2.1` existe una carpeta existe un archivo llamado `config.ttyIAX` que debe copiarse al directorio `/var/spool/hylafax/etc` .

```
# cp config.ttyIAX /var/spool/hylafax/etc/
```

Este archivo debe modificarse tal como se muestra a continuación:

```
CountryCode:            593
AreaCode:               2
FAXNumber:              +593-2-223-0096
LongDistancePrefix:     0
InternationalPrefix:    00
DialStringRules:        etc/dialrules
ServerTracing:          0xFFF
SessionTracing:         0xFFF
RecvFileMode:           0664
LogFileMode:            0664
DeviceMode:             0666
RingsBeforeAnswer:      1
SpeakerVolume:          off
GettyArgs:              "-h %l dx_%s"
LocalIdentifier:        "MachangaraSoft"
TagLineFont:            etc/lutRS18.pcf
TagLineFormat:          "From %%l|%c|Page %%P of %%T"
MaxRecvPages:           150
```

El resto de parámetros puede quedar como estaba.

Para que la recepción de fax esté permanentemente activa, al igual que en el caso de iaxmodem, se agrega la siguiente línea:

```
fax:2345:respawn:/usr/sbin/faxgetty ttyIAX
```

Y se ejecuta el comando:

```
# telinit q
```

Finalmente, para configurar la dirección de correo electrónico a la que el fax llegará digitalizado como archivo adjunto, se editará el archivo `/var/spool/hylafax/etc/FaxDispatch:`

```
SENDTO=MachangaraSoft;
FILETYPE=pdf;

case "$DEVICE" in
        ttyIAX)           SENDTO=jestrada@machangarasoft.com;;
esac
```

Si se alcanza o marca la extensión 300, se obtendrá el tono de fax correspondiente.

ANEXO C

Especificaciones de Hardware Telefónico

digium data sheet

X100P

Scalable and Effective SOHO Solution

The Wildcard X100P provides a single FXO interface for connecting the Open Source Asterisk PBX server with an incoming line. The device allows Asterisk to answer calls from a service provider's standard analog line or to receive calls from another PBX over TDM without the use of T1 hardware. The X100P is ideal for Interactive Voice Response (IVR) and Voicemail applications. The X100P supports all standard enhanced call features including Caller ID, Call Conferencing, and Call Waiting/Caller ID.

By using the X100P in conjunction with the Open Source Asterisk PBX, the X100P also provides an economical way to implement powerful and flexible call services, such as multilayered IVR, small VoIP gateways, directory services, and business class voicemail.

Target Applications:

- SOHO (Small Office Home Office) applications
- Packet Voice Gateways and Switches
- Conferencing
- One Number Services
- Message Services
- Voicemail Server
- Customized and Web Telephony

Services and Features:

- CallerID and CallWaiting/CallerID
- Digital Gain Control (Transmit and Receive)
- PCI Half-Length Slot
- RJ-11 Connector

Environment Conditions:

- Operation Range: 0 to 50°C, 32° to 122° F
- Storage Range: -20° to 65°C, 4° to 149° F
- Humidity: 10-90% non-condensing

Hardware and Software Requirements:

- 500-Mhz Pentium III or better with 64MB RAM
- Available PCI Slot
- Linux 2.4 Kernel

digium

About Digium

Based in high-tech Huntsville, Alabama, Digium is the creator and primary developer of Asterisk, the industry's first Open Source PBX. Used in combination with Digium's PCI telephony interface cards, Asterisk offers a strategic, highly cost-effective approach to voice and data transport over TDM, switched, IP, and Ethernet architectures.

Digium solutions reduce the costs of traditional TDM and VoIP implementations through Open Source, standards-based software and innovative hardware solutions, including legacy PBX, IVR, Auto-attendant, and next-generation gateways, media servers, and application servers. Digium hardware supports traditional voice protocols, including PRI, RBS, FXS, FXO, E&M, Feature Group D, Groundstart, and Loopstart. Data protocols include PPP, Cisco HDLC, and Frame Relay. For packet voice, Asterisk supports IAX (Inter-Asterisk eXchange), SIP, MGCP, Skinny, and H.323 VoIP protocols.

Digium provides a highly refined selection of quality hardware and software products, developed and implemented using innovative engineering techniques (primarily Open Source development). A full range of professional services complement these product lines, including consulting, technical support, and custom software development services.

The Open Source Communications Revolution is here, and Digium is leading the way.

digium data sheet

TDM400P

Scalable and Effective SOHO Solution

The Wildcard TDM400P is a half-length PCI 2.2-compliant card that supports FXS and FXO station interfaces for connecting analog telephones and analog POTS lines through a PC. Using Digium's TDM hardware, Open Source Asterisk PBX software, and a standard PC, users can create a Small Office Home Office (SOHO) telephony environment which includes all the sophisticated features of a high-end PBX/Voicemail platform.

The TDM400P takes the place of an expensive channel bank and brings the system price point to the lowest in the industry. The FXO and FXS modules are interchangeable to create various combinations of interfaces. To scale this solution, just add additional TDM400P cards. This revolutionary solution has an unprecedented price point in the industry!

Target Applications

- Small Office Home Office (SOHO) applications
- Gateway Termination to Analog Telephones
- Add Inexpensive Analog Phones to Existing PBXs
- Wireless Point-to-Point Applications between Asterisk Servers

Services and Features

- Caller ID and Call Waiting Caller ID
- ADSI Telephones
- PCI Half-length Slot
- RJ-11C Connector

Environment Conditions

- Operation Range: 0° to 50°C, 32° to 122° F
- Storage Range: -20° to 65°C, 4° to 149° F
- Humidity: 10-90% non-condensing

Standard Configurations

- TDM10B: 1-port FXS bundle
- TDM40B: 4-port FXS bundle
- TDM01B: 1-port FXO bundle
- TDM04B: 4-port FXO bundle
- TDM11B: 1-port FXS &1-port FXO bundle
- TDM22B: 2-port FXS &2-port FXO bundle
- TDM31B: 3-port FXS &1-port FXO bundle
- *Other configurations available on request

Hardware and Software Requirements

- 500-Mhz Pentium III or better with 64MB RAM
- Available PCI Slot

digium

About Digium

Based in high-tech Huntsville, Alabama, Digium is the creator and primary developer of Asterisk, the industry's first Open Source PBX. Used in combination with Digium's PCI telephony interface cards, Asterisk offers a strategic, highly cost-effective approach to voice and data transport over TDM, switched, IP, and Ethernet architectures.

Digium solutions reduce the costs of traditional TDM and VoIP implementations through Open Source, standards-based software and innovative hardware solutions, including legacy PBX, IVR, Auto-attendant, and next-generation gateways, media servers, and application servers. Digium hardware supports traditional voice protocols, including PRI, RBS, FXS, FXO, E&M, Feature Group D, Groundstart, and Loopstart. Data protocols include PPP, Cisco HDLC, and Frame Relay. For packet voice, Asterisk supports IAX (Inter-Asterisk eXchange), SIP, MGCP, Skinny, and H.323 VoIP protocols.

Digium provides a highly refined selection of quality hardware and software products, developed and implemented using innovative engineering techniques (primarily Open Source development). A full range of professional services complement these product lines, including consulting, technical support, and custom software development services.

The Open Source communications revolution is here, and Digium is leading the way.

Digium, Inc. www.digium.com · 877.linux.me! · (877.546.8963)

digium data sheet

TDM2400P

Scalable and Effective SME Solution

The Wildcard TDM2400P is a full-length 32-bit 33MHz PCI 2.2-compliant card that supports quad FXS station and quad FXO office interfaces (for a total of 24 maximum channels on a single PCI slot) for connecting analog telephones and lines through a PC. The TDM2400P can be used to connect standard analog telephone lines to a PC and Asterisk.

Using Digium's TDM hardware, Asterisk PBX software, and standard PCs, users can create an SME or SOHO telephony environment that includes all the sophisticated features of a high-end PBX/Voicemail platform.

Using an industry-standard PCI chip, the TDM2400P replaces the requirement for separate channel bank and T1 interface cards at the industry leading price. The quad FXO and quad FXS modules are interchangeable, allowing the creation of any combination of interfaces. The available hardware echo cancellation module provides 256 taps of echo cancellation for superior echo cancellation on both FXO and FXS interfaces on the card itself. Scaling of this solution is accomplished by adding additional TDM2400P cards.

For more information on the TDM2400P, please visit the *Products* section of digium.com.

Target Applications

- Channel Bank Replacement / Alternative
- Small Office Home Office (SOHO) applications
- Small and Medium Enterprise (SME) applications
- Gateway Termination to analog telephones and lines

Services and Features

- Caller ID and Call Waiting Caller ID
- ADSI Telephones
- PCI Full-length Slot
- RJ-21X Connector

Environment Conditions

- Operation Range: 0° to 50°C, 32° to 122° F
- Storage Range: -20° to 70°C, -4° to 158° F
- Humidity: 10-90% non-condensing

Standard Configurations

- Up to six slots available for 4-port FXS or FXO modules, or combination
- Hardware Echo cancellation available for each module

Modular Parts

- X400M: Quad FXS Module, up to 6 per TDM2400P
- X400M: Quad FXO Module, up to 6 per TDM2400P
- VPM100M: 32ms Tail / 256 Taps G.168 Echo Can. Module, 1 per TDM2400P

Hardware and Software Requirements

- 1.4 GHz Pentium 4 or better with 64MB RAM
- Available Full-length PCI Slot

digium

About Digium

Based in high-tech Huntsville, Alabama, Digium is the creator and primary developer of Asterisk, the industry's first Open Source PBX. Used in combination with Digium's PCI telephony interface cards, Asterisk offers a strategic, highly cost-effective approach to voice and data transport over TDM, switched, IP, and Ethernet architectures.

Digium solutions reduce the costs of traditional TDM and VoIP implementations through Open Source, standards-based software and innovative hardware solutions, including legacy PBX, IVR, Auto-attendant, and next-generation gateways, media servers, and application servers. Digium hardware supports traditional voice protocols, including PRI, RBS, FXS, FXO, E&M, Feature Group D, Groundstart, and Loopstart. Data protocols include PPP, Cisco HDLC, and Frame Relay. For packet voice, Asterisk supports IAX (Inter-Asterisk eXchange), SIP, MGCP, Skinny, and H.323 VoIP protocols.

Digium provides a highly refined selection of quality hardware and software products, developed and implemented using innovative engineering techniques (primarily Open Source development). A full range of professional services complement these product lines, including consulting, technical support, and custom software development services.

The Open Source communications revolution is here, and Digium is leading the way.

LINKSYS®
A Division of Cisco Systems, Inc.

Feature-rich telephone service through your Internet connection!

The Linksys Phone Adapter enables high-quality feature-rich telephone service through your cable or DSL Internet connection. Just plug it into your home Router or Gateway and use the two standard telephone jacks to connect your existing phones or fax machines. Each phone jack operates independently, with separate phone service and phone numbers—like having two phone lines. With an appropriate Internet telephone service provider, you'll get clear telephone reception and reliable fax connections, even while using the Internet at the same time for normal data operations.

With Internet telephony, along with low domestic and international phone rates, an impressive array of special phone features are available. Choose your preferred free local dialing US area code, regardless of where you live. Or add a virtual phone number in any area code, forwarded to your Internet phone. You can even add a toll-free number. The Linksys Phone Adapter is compatible with these and all of the other special telephone features that are available from your telephone service provider, such as Caller ID, Call Waiting, Voicemail, Call Forwarding, Distinctive Ring, etc.

Let the Linksys Phone Adapter turn your existing Internet connection into a high-quality high-value telephone service.

Enables feature-rich telephone service over your cable or DSL Internet connection	Two standard telephone jacks for your phones or fax machines, with independent phone numbers	High quality, clear sounding voice service simultaneous with Internet use	Compatible with all common telephone features: Caller ID, Call Waiting, Voicemail, etc.

Phone Adapter

WIRED

with 2 Ports for Voice-over-IP

Product Data

Model No. PAP2

Cisco Systems

Ç

Phone Adapter
with 2 Ports for Voice-over-IP

Features

- Two voice ports (RJ-11) for analog phones or fax machines with two independent telephone numbers
- One RJ-45 port for 10Base-T Ethernet connection
- Supports Dynamic Host Configuration Protocol (DHCP)
- Supports Session Initiation Protocol (SIP)
- Supports multiple voice compression G.711, G.726, G.729, and G.723.1
- Web-based configuration through a built-in web server
- Telephone key pad configuration via voice prompts
- Supports DTMF tone detection and generation
- Supports FSK Caller ID, DTMF Caller ID and FSK VMWI
- Support echo cancellation and Voice Activity Detection (VAD)
- Password protected access and configuration
- Supports auto-provisioning with remote firmware upgrade

Specifications

Model	PAP2
Standards	IEEE 802.3 (10BaseT), IEEE 802.3u (100BaseTX)
Ports	One 10/100 RJ-45 Network Port, Two Standard Phone Ports, One Power Port
Cabling Type	RJ-45 Ethernet Category 5, RJ-11 Standard Phone Cable
LEDs	Power, Ethernet, Phone1, Phone2
Voice Protocol	Session Initiation Protocol (SIP v2)
Voice Codecs	G.711 a-law, G.711μ-law, G.726, G.729 A, G.723.1
Ringer Equivalence Number (REN)	5 REN per RJ-11 port
Ring Frequency	10 Hz - 40 Hz
FXS Port Impedance	Eight Configurable Setting Including North America 600 ohms, European CTR21
Ring Voltage	60 - 90 Vrms Configurable
Security Features	Password-Protected Administration

Environmental

Dimensions	101 mm x 101 mm x 15 mm (3.98" x 3.98" x 0.59")
Unit Weight	4.80 oz. (0.14 kg)
Power Input	5V DC 2.0A
Certifications	FCC, cUL, CE
Operating Temp.	41°F to 113°F (5°C to 45°C)
Storage Temp.	-13°F to 185°F (-25°C to 85°C)
Operating Humidity	10% to 90%, Non-Condensing
Storage Humidity	5% to 90%, Non-Condensing

Linksys
A Division of Cisco Systems, Inc.
18512 Teller Avenue
Irvine, CA 92612 USA

E-mail: sales@linksys.com
support@linksys.com

Web: http://www.linksys.com

Linksys products are available in more than 50 countries, supported by 12 Linksys Regional Offices throughout the world. For a complete list of local Linksys Sales and Technical Support contacts, visit our Worldwide Web Site at www.linksys.com.

Minimum Requirements

- High-speed Internet connection (cable/DSL/other)
- Broadband Router or Gateway to share Internet connection
- Regular analog touch-tone telephone or fax machine
- CD-ROM drive

Package Contents

- Phone Adapter
- Power Adapter
- User Guide on CD-ROM
- Network Cable
- Quick Installation
- Registration Card
- Vonage Service Materials

PAP2-DS-40810BJL

Product Data

Model No. PAP2

AT-530 IP PHONE

Enterprise, small office and residential applications

KEY FEATURES

- Support two sip servers running at the same time.
- Redundancy sip server capable.
- NAT, Firewall.
- DHCP client and server.
- Support PPPoE, (used for ADSL, cable modem connecting).
- Support major G7.xxx CODEC.
- VAD,CNG.
- G.168compliant 32ms echo cancellation
- E.164 dial plan and customized dial rules
- Hotline.
- Call Forward, Call Transfer, 3-way conference calls
- Call ID display
- DND(Do Not Disturb),Black List,Limit List

DATA FEATURES

- Static/Dynamic WAN-IP-Addressing
- PPPoE

MANAGEMENT

- Web, telnet and keypad management.
- Adjustable user password and super password
- Upgrade firmware through HTTP, FTP or TFTP.
- Telnet remote management.
- Upload/download setting file
- Auto-provision.
- Safe mode provide reliability
- Phone book, maximum 100 entries.

INTERFACES

- Two RJ45 ports, one for WAN, one LAN.
- Power port

SUPPORTED SPECIFICATION AND APPLICATIONS

DATA NETWORKING

- MAC Address
- TCP:Transmission Control Protocol
- DHCP:Dynamic Host Configuration Protocol
- PPPoE:PPP Protocol over Ethernet
- SNTP, Simple Network Time Protocol
- STUN - Simple Traversal of User Datagram ...
- MD5 Message-Digest Algorithm
- DNS：Domain Name Server
- RTP: Real-time Transport Protocol
- RTCP:Real-time Control Protocol
- Telnet:Internet's remote login protocol
- HTTP:Hyper Text Transfer protocol
- FTP:File Transfer protocol
- TFTP:Trivial File Transfer Protocol

CALL CONTROL /VOIP FEATURES

- SIP RFC3261,RFC 2543
- Tone generation and Local DTMF re-generation according with ITU-T
- G.711(A-law or u-law)
- G.723.1(6.3kbps,5.3 kbps)
- G729
- AGC(Auto Gain Control)
- G.168/165 compliant 16ms echo cancellation
- AEC(Auto Echo Cancellation)
- VAD (Voice Activity Detection)
- CNG(Comfort Noise Generation

ENVIROMENTAL

ELECTRIC REQUIREMENTS

- Voltage: 9V ~ 24V
- Power adapter: output DC 12V/450 mA

OPERATING REQUIREMENT

- Operation temperature: 0 to 40° C (32° to 104° F)
- Storage temperature: -30° to 65° C (-22° to 149° F)
- Humidity: 10 to 90% no dew

REGULATORY COMPLIANCE

- CE,FCC part 15,

ANEXO D

Complementos – Prototipo de Firewall para el MachángaraSoft

Componentes para la implementación de Reglas de Tráfico con `iptables`

Tablas

Existen, dentro de iptables, tres tablas que ya están incorporadas, cada una de las cuales contiene ciertas cadenas predefinidas. Se pueden crear tablas, a través de módulos de extensión. Además el administrador tiene la potestad de crear y eliminar cadenas definidas por usuarios dentro de cualquier tabla. Inicialmente todas las cadenas están vacías y tienen una política de destino que permite que pasen todos los paquetes sin ser bloqueados o alterados.

Tabla de Filtros (filter table).- Todos los paquetes pasan a través de esta tabla, que se encarga de bloquear o permitir que estos paquetes continúen su camino (filtrarlos). Esta tabla posee las siguientes cadenas predefinidas:

Cadena INPUT (o Cadena de Entrada). Va dirigida a todos los paquetes cuyo destino es el propio sistema de firewall.
Cadena OUPUT (o Cadena de Salida). Va dirigida a los paquetes generados en el sistema y que salen de él.
Cadena FORWARD (o Cadena de Redirección o Reenvío). Dirigida a los paquetes que atraviesan el sistema, para enrutarse, sin que el firewall sea su origen o destino.

Tabla NAT.- Se encarga de configurar las reglas de re-escritura de direcciones o puertos en los paquetes. El paquete inicial de cualquier conexión pasa a través de esta tabla. Algunas de las cadenas predefinidas en esta tabla son:
Cadena PREROUTING (o Cadena de Preruteo). Los paquetes entrantes pasan a través de esta cadena antes de que se consulte la tabla de enrutamiento local, principalmente para DNAT (*destination*-NAT o traducción de direcciones de red de destino)
Cadena POSTROUTING (o Cadena de Postruteo). Los paquetes salientes pasan por esta cadena luego de tomarse la decisión de enrutamiento, principalmente para SNAT (*source*-NAT o traducción de direcciones de red de origen)
Cadena OUTPUT (o Cadena de Salida). Esta cadena permite realizar un DNAT limitado en paquetes salientes, generados de forma local.

Tabla de destrozo (mangle table).- Es la encargada de ajustar algunas de las opciones de los paquetes, la calidad de servicio, por ejemplo. Esta tabla es atravesada por todos los paquetes y por el hecho de estar diseñada para efectos avanzados, contiene todas las cadenas predefinidas posibles:
Cadena PREROUTING (o Cadena de Preruteo). Es atravesada por los paquetes que logran entrar en el sistema de firewall, antes que el enrutamiento decida si el paquete debe ser reenviado (cadena de REENVIO) o si tiene destino local (cadena de ENTRADA).

Cadena INPUT (o Cadena de Entrada). Es atravesada por los paquetes cuyo destino es el sistema local.
Cadena FORWARD (o Cadena de Reenvío). Es atravesada por todos los paquetes que simplemente pasan por este sistema.
Cadena OUTPUT (o Cadena de Salida). Atravesada por todos los paquetes originados en este sistema.
Cadena POSTROUTING (o Cadena de Postruteo). Es atravesada por todos los paquetes que abandonan el sistema, luego de que se haya tomado la decisión de enrutamiento.

Cada cadena incorpora una lista de reglas y cuando un paquete se envía a una de estas cadenas, se compara, en orden, con cada regla en la cadena. La regla especifica las propiedades que debe tener el paquete para que la regla se cumpla, ya sea el número de puerto o la dirección IP. Si la regla no se cumple, el proceso continúa con la regla siguiente, pero si la regla se cumple, se siguen las instrucciones de destino de la regla (y entonces cualquier otro proceso de la cadena normalmente se aborta). Ciertas propiedades de los paquetes pueden únicamente examinarse en determinadas cadenas (por ejemplo, la interfaz de red de salida no es válida en la cadena INPUT). Asimismo, algunos destinos sólo pueden usarse en ciertas cadenas y/o ciertas tablas, por ejemplo, el destino SNAT solamente puede utilizarse en la cadena de POSTROUTING de la tabla de traducción de direcciones de red.

Destino de Reglas

El destino de una regla puede definirlo el usuario a través de un nombre o mediante uno de los destinos ya incorporados: ACCEPT, DROP, QUEUE o RETURN (*aceptar, descartar, encolar o retornar*, respectivamente). Cuando un destino es el nombre de una cadena definida por el usuario, el paquete es dirigido hacia esa cadena para que sea procesado. Si el paquete logra atravesar la cadena definida por el usuario sin que ninguna de las reglas de esa cadena actúe sobre él, el procesamiento del paquete continúa desde donde había quedado en la cadena actual. El nivel de anidación de llamados entre cadenas puede elevarse hasta donde se desee.
Los destinos existentes ya incorporados son:
ACCEPT (aceptar). Este destino permite que el paquete sea aceptado. El significado de esto depende de la cadena que ha realizado la aceptación. Por ejemplo, a un paquete que es aceptado por la cadena INPUT se le permite ser recibido por el sistema, aun paquete aceptado en la cadena OUTPUT se le permite abandonar el sistema y aun paquete aceptado en la cadena FORWARD se le permite ser enrutado a través del sistema.
DROP (descartar). Este destino permite que el paquete sea descartado sin procesamiento posterior alguno. El paquete solo desaparece sin que se indique que se descartó. Sin embargo, esto se refleja al terminal origen como un *communication timeout* (máximo tiempo de espera en la comunicación), lo que podría causar cierta confusión.
QUEUE (encolar). En este destino, el paquete es enviado a una cola. Si no hay ninguna aplicación que lea la cola, el destino es equivalente a DROP.

RETURN (retorno). Este destino permite que el paquete deje de circular por la cadena en cuya regla se ejecutó el destino RETURN. Si esta cadena es una subcadena de otra, el paquete continuará por el paquete superior como si nada hubiera pasado. Si, en cambio, se trata de una cadena principal (como la cadena INPUT), al paquete se le aplicará la política por defecto de la cadena en cuestión (ACCEPT, DROP o cualquiera de ellas).

REJECT (rechazo). Este destino maneja el mismo efecto que DROP, pero envía un paquete de error a quien envió el paquete originalmente. Es usado principalmente en las cadenas INPUT y FORWARD de la tabla de filtrado. Se puede controlar el tipo de paquete de error que se envía de regreso con el parámetro *'reject-with'*. Este paquete de rechazo puede indicar explícitamente que la conexión ha sido filtrada, aunque es más común que se prefiera que el paquete indique simplemente que la computadora no acepta este tipo de conexión. Si el parámetro '-- reject with' no se especifica, el paquete de rechazo por defecto es siempre *icmp-port-unreachable.*

LOG (bitácora). Este destino permite llevar el paquete a una bitácora o *log* y puede utilizarse en cualquier cadena en cualquier tabla, comúnmente con miras a depuración.

DNAT. Este destino hace que la dirección (y opcionalmente el puerto) de destino del paquete se reescriban para realizar NAT. A través de la opción '--to-destination' se debe indicar el destino que se usará. Es válido únicamente en las cadenas de OUTPUT y PREROUTING dentro de la tabla de NAT. Se debe resaltar que esta decisión se recuerda para todos los paquetes futuros que pertenecen a la misma conexión de modo que las respuestas tengan su dirección y puerto de origen cambiados al original.

SNAT. Este destino hace que la dirección (y opcionalmente el puerto) de origen del paquete se reescriban para realizar NAT. Gracias a la opción '--to-source' se debe indicar el origen que se utilizará. Esto resulta únicamente válido dentro de la cadena POSTROUTING dentro de la tabla de NAT y, al igual que en DNAT, se recuerda para todos los paquetes que pertenecen a la misma conexión.

MASQUERADE. Es una forma especial de destino, restringida de SNAT, restringida para direcciones IP dinámicas, como las provistas por muchos de los ISPs, para módems o DSL. En lugar de que se cambie la regla de SNAT cada vez que vez que su dirección IP cambia, se calcula la dirección IP de origen a la cual se hará NAT, fijándose en la dirección IP de la interfaz de salida cuando un paquete cumple la regla. Además, permite recordar las conexiones que usan MASQUERADE y si la dirección de la interfaz cambia, se olvidan todas las conexiones que hacen NAT a la dirección vieja.

Un esquema que puede ayudarnos a comprender de mejor manera la forma como los paquetes recorren las cadenas y tablas de netfilter:

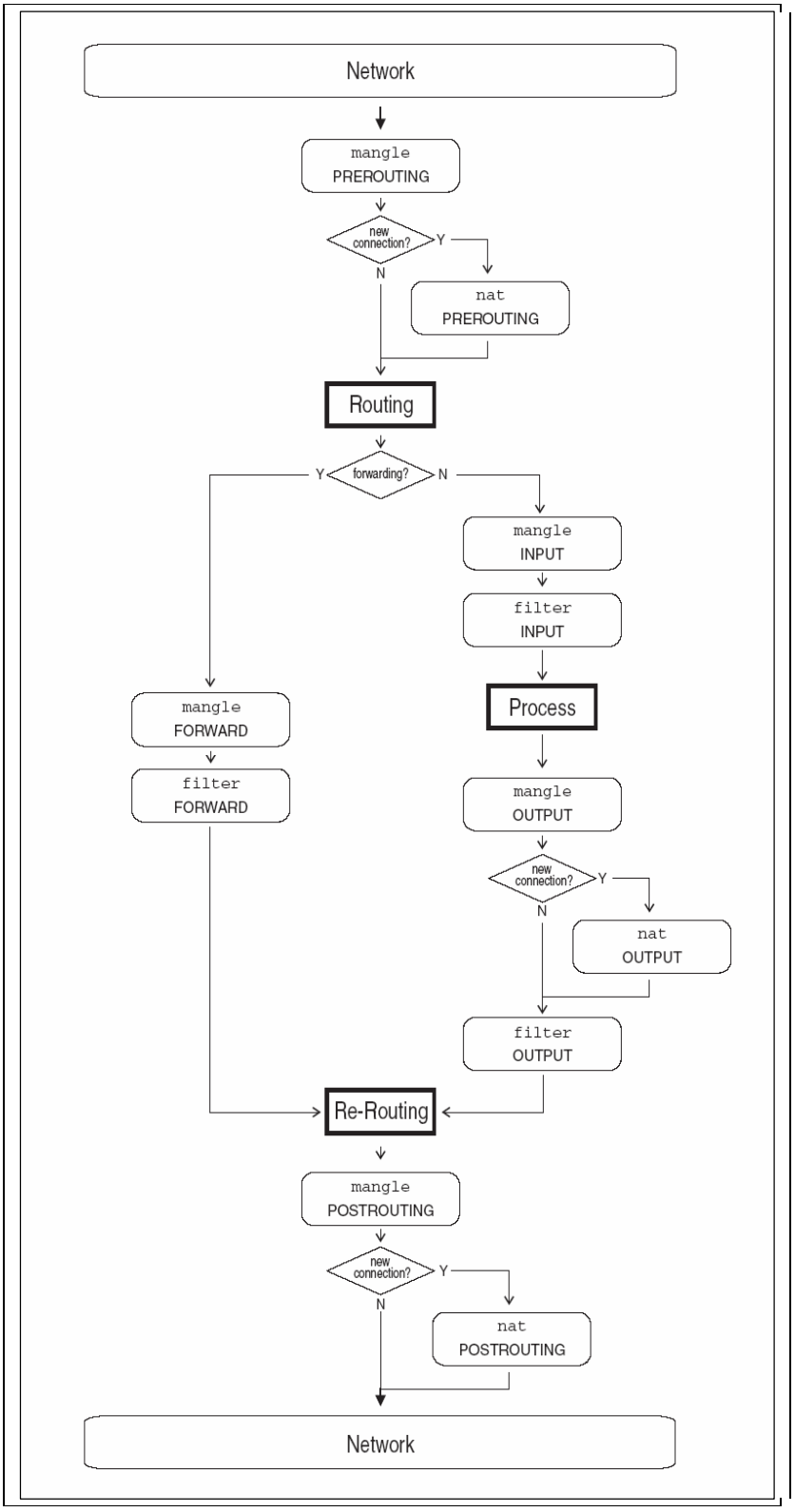


Figura. Esquema de travesía de paquetes por tablas y cadenas.

Seguimiento de Conexiones

Como se indicó, una importante característica de la estructura de netfilter/iptables es el seguimiento de conexiones o *connection tracking*. Este seguimiento le permite al núcleo de linux monitorear todas las conexiones o sesiones lógicas de red y así relacionar los paquetes que pueden llegar a ser parte de esa conexión. NAT depende de esa información para traducir los paquetes relacionados de la misma forma.

Este seguimiento de conexiones permite la clasificación de los paquetes en uno de los estados siguientes:

NEW (nuevo). Un paquete que está intentando crear una conexión nueva.

ESTABLISHED (establecido). Un paquete que es parte de una conexión ya existente.

RELATED (relacionado). Un paquete relacionado, aunque no realmente parte de la conexión existente.

INVALID (inválido). Un paquete que no es parte de una conexión existente e incapaz de crear una conexión nueva.

Un panorama común podría ser que el primer paquete de una conexión y que vea el cortafuegos se clasifique como NEW, la respuesta como ESTABLISHED y un error ICMP será RELATED. Asimismo, un paquete de error ICMP que no se corresponda con una conexión conocida será INVALID.

La capacidad de seguimiento de conexiones de iptables permite usar la información que genera para hacer reglas de filtrado más potentes y fáciles de manejar. La extensión de “estado” permite que las reglas de iptables examinen la clasificación seguimiento de conexión para un paquete. Por ejemplo, una regla puede permitir el paso solamente a paquetes NEW desde dentro del firewall hacia el mundo exterior, permitiendo al mismo tiempo el paso de paquetes RELATED y ESTABLISHED en ambas direcciones, lo que permitiría el paso de paquetes de respuesta normales desde el exterior (ESTABLISHED), evitando que lleguen nuevas conexiones desde el Internet.

ESPECIFICACIÓN DE REGLAS

Dado que iptables requiere privilegios elevados para operar, el único que puede ejecutarlo es el súper-usuario (root). Su sintaxis en la mayoría de los sistemas linux está detallada en su página de manual (man) y se accede a ella escribiendo “man iptables” desde la línea de comandos.

Algunas Opciones Comunes

Las opciones más comunes dentro de las formas de invocación de `iptables` son:

`-t` *`tabla`* Hace que el comando se aplique a la *tabla* especificada. Si se omite esta opción, el comando se aplica a la tabla *filter* por defecto.

-n Genera una salida numérica (es decir el despliegue de números de puerto en lugar de nombres de servicio y de direcciones IP en vez de nombres de dominio).

-v Genera una salida con detalles (*verbose* en inglés)

--line-numbers Agrega números de línea al comienzo de cada regla, cuando se listan reglas. Estos números corresponden a la posición de la regla en la cadena.

Especificación de Reglas

Casi la totalidad de formas de comandos de iptables necesitan que se les indique una especificación de reglas, usada para corresponderse con un subconjunto particular de tráfico de paquetes de red procesados por una cadena. Dentro de esta especificación se incluye también un destino que indica qué hacer con los paquetes que cumplen con la regla. Algunas de las opciones que se utilizan para crear estas especificaciones de las reglas son:

-j *destino*
--jump *destino*

Que especifica el destino de una regla. Como ya se indicó, el destino es el nombre de una cadena definida por el usuario (que se ha creado con la opción –N, uno de los destinos incorporados ACCEPT, DROP, QUEUE o RETURN, o un destino de extensión como REJECT, LOG, DNAT o SNAT).

Si esta opción no se especifica en una regla, la correspondencia de la regla no tendrá efecto en el destino de un paquete, aunque los contadores en la regla se incrementarán.

-i [!] *in-interface*
--in-interface [!] *in-interface*

Indica el nombre de la interfaz por la que un paquete va a ser recibido (válida únicamente para paquetes entrando en las cadenas INPUT, FORWARD y PREROUTING). Si el argumento '!' se utiliza antes del nombre de la interfaz, el significado se invierte. Por otro lado, si el nombre de la interfaz termina con '+', cualquier interfaz que comience con este nombre cumplirá con la regla. Si la opción se omite, se hará corresponder todo el nombre de la interfaz.

-o [!] *out-interface*
--out-interface [!] *out-interface*

Indica el nombre de una interfaz a través de la cual un paquete va a ser enviado (saliente), para paquetes entrando en las cadenas FORWARD, OUTPUT y PREROUTING. Las opciones '!' y '+', tienen el mismo significado que en la especificación anterior.

-p [!] *protocol*

--protocol [!] *protocol*

Como su nomenclatura lo indica, hace corresponder los paquetes del nombre de protocolo especificado. Si '!' precede el nombre del protocolo, la especificación se invierte y no se hace corresponder el protocolo especificado. Algunos nombres de protocolos válidos son ICMP, UDP, TCP, etc.

-s [!] *origen*[/prefijo]

--source [!] *origen*[/prefijo]

Hace corresponder los paquetes IP que provienen de la dirección de origen especificada. Esta dirección de origen puede ser una dirección IP, una dirección IP con prefijo de red asociado, o un nombre de host (hostname). De igual manera, si '!' precede al origen, hacen corresponder todos los paquetes que no provienen del origen especificado.

-d [!] *destino*[/prefijo]

--destination [!] *destino*[/prefijo]

Hace corresponder a todos los paquetes IP que se dirigen a la dirección IP destino especificada. Esta dirección puede especificarse de la misma forma que se explica en el párrafo anterior.

--destination-port [!] [*puerto*[[:*puerto*]]]

--dport [!][*puerto*[[:*puerto*]]]

Esta especificación hace corresponder los paquetes TCP o UDP (dependiendo del argumento de la opción -p) destinados a los puertos o rangos de puerto (si se usa la forma *puerto:puerto*) especificados. Si '!' precede la especificación de los puertos, se hacen corresponder todos los paquetes TCP o UDP que no provienen de los puertos o rango especificados.

--source-port [!] [*puerto*[[:*puerto*]]

--sport [¡][*puerto*[[:*puerto*]]

Hace match de todos los paquetes TCP o UDP (dependiendo del argumento de la opción -p) que vienen de los puertos o rango de puertos especificados. El parámetro '!' tiene el efecto correspondiente ya analizado.

--tcp-flags [!] *mask comp*

Hace corresponder los paquetes TCP en los que se han marcado o desmarcado determinadas banderas del protocolo TCP. El argumento inicial especifica las banderas que se van a examinar en cada paquete, escritas en una lista separada por comas (sin espacios). El argumento siguiente es otra lista separada por comas de banderas que deben estar marcadas dentro de las que se deben examinar. Estas banderas son SYN, ACK, FIN, RST, URG, PSH, ALL y NONE. Por ejemplo, la opción "–tcp-flags SYN,ACK,FIN,RST SYN" solamente hará corresponder los paquetes con la bandera SYN marcada y las banderas ACK, FIN y RST desmarcadas.

[!] –syn

Hace corresponder los paquetes TCP que poseen la bandera SYN marcada y las banderas ACK, FIN y RST desmarcadas. Estos paquetes se usan para iniciar conexiones TCP, por lo que si se bloquean en la cadena de INPUT, se evitan las conexiones TCP entrantes, pero las conexiones TCP salientes no serán afectadas. Se puede notar que esta opción puede combinarse con otras como –source, para bloquear o permitir conexiones TCP entrantes solamente de ciertas terminales o redes.

Invocación

De las formas de invocación que se describirán, puede aclararse el significado de algunos de los símbolos:

Los ítems entre llaves {…|…|…} presentan opciones requeridas, pero puede indicarse uno de los ítems separados por '|'.

Los ítems entre corchetes, […], son opcionales.

iptables { -A | --append | -D | --delete } *cadena especificación-de-regla* [*opciones*]

Esta forma de invocar del comando agrega (a través de -A o --append) o elimina (-D o –delete) una regla de la cadena especificada, por ejemplo, para agregar una regla a la cadena INPUT en la tabla *filter* (tabla por defecto cuando la opción -t no se especifica) que descarte todos los paquetes UDP, usamos el comando:

```
iptables –A INPUT –p udp –j DROP
```

Para borrar la regla que se agregó, usaremos:

```
iptables –D INPUT –p udp –j DROP
```

Se debe aclarar que si existen varias reglas idénticas en la cadena, se borrará únicamente la primera regla que corresponda.

iptables { -R | --replace | -I | --insert } *cadena numregla especificación-de-regla* [*opciones*]

Esta invocación de iptables reemplaza (`-R o --replace`) una regla existente o inserta (`-I o --insert`) una regla nueva en la cadena especificada. Por ejemplo, si deseáramos reemplazar la sexta regla en la cadena de INPUT por una regla que descarte todos los paquetes ICMP, deberíamos usar:

```
iptables –R INPUT 6 –p icmp –j DROP
```

Si, en cambio, quisiéramos insertar una regla nueva en el segundo lugar de la cadena OUTPUT que descarte todo el tráfico TCP dirigido al puerto 80 en cualquier host, se usaría:

```
iptables –I OUTPUT 2 –p tcp --dport 80 –j DROP
```

iptables { -D | --delete } *cadena numregla* [*opciones*]

Esta forma de invocación de iptables elimina una regla del índice numérico en la cadena especificada. Las reglas se numeran comenzando en 1. Si quisiéramos eliminar la cuarta regla de la cadena FORWARD, se usa el comando:

```
iptables –D FORWARD 3
```

iptables { -L | --list | -F | --flush | -Z | --zero } [*cadena*] [*opciones*]

Esta forma de invocación de iptables nos permite listar las reglas existentes en una cadena (`-L` o `--list`), eliminar todas las reglas de una cadena (`-F` o `--flush`), o para poner en cero el byte y los contadores de paquetes de una cadena (`-Z` o `--zero`). Si no se especifica ninguna cadena, la operación se realiza para todas las cadenas. Si necesitáramos listar las reglas en la cadena INPUT, usaríamos:

```
iptables –L INPUT
```

Ahora si deseáramos eliminar todas las reglas, deberíamos usar el comando:

```
iptables –F
```

iptables { -N | --nueva-cadena } *cadena*
iptables { -X | --borrar-cadena } [cadena]

Esta forma de invocación de iptables se utiliza para crear (`N` o `--new-chain`) una cadena definida por el usuario o para eliminar (`-X` o `--delete-chain`) una cadena que ha sido definida por el usuario existente. Como antes, si no se especifica ninguna cadena con las opciones `–X` o `--delete-chain`, se eliminarán todas las cadenas definidas por el usuario. No se pueden eliminar las cadenas predefinidas como OUTPUT o INPUT.

iptables { -P | --policy } *cadena destino*

Esta forma de invocación del comando permite especificar la **política de destino** para una cadena. Por ejemplo, para especificar la política de destino para la cadena INPUT en DROP, tendríamos que usar el comando siguiente:

```
iptables –P INPUT DROP
```

iptables { -E | --rename-chain } *nombre-de-cadena-viejo nombre-de-cadena-nueva*

Esta forma de invocación permite renombrar una cadena definida por el usuario.

SERVIDOR DHCP en Linux

DHCP *(Dynamic Host Configuration Protocol) e*s un protocolo de red que permite a cada nodo en una red adquirir sus parámetros de configuración de conexión de forma automática. Este protocolo se basa en un modelo cliente/servidoren el que el servidor dispone esencialmente de un conjunto de direcciones IP dentro de un rango determinado que va distribuyendo de forma dinámica a los miembros de la red a la que pertenece el mencionado servidor. Este protocolo permite llevar un control del tiempo de asignación de los parámetros y de las máquinas objetos de asignación.

Este protocolo facilita la administración de equipos dentro de una red de computadores pues evita que quien esté encargado del direccionamiento tenga que configurar manualmente estos parámetros de red, lo cual tomaría una gran cantidad de tiempo si se trata de una red de grandes dimensiones.

Además es posible ejercer un pequeño control respecto de quién puede ser sujeto de asignación. A partir de la dirección física de cada máquina se puede designar una dirección IP correspondiente de modo que se evite la conexión de clientes no identificados.

Parámetros Configurables

Entre los parámetros que se pueden configurarse a través de este servicio están:

- Dirección IP del servidor DNS
- Nombre DNS
- Dirección IP de la puerta de enlace predeterminada *(Default Gateway)*
- Dirección IP de *Broadcast*
- Máscara de Subred
- Tiempo máximo de espera del ARP *(Adress Resolution Protocol)*
- *MTU* – Unidad de Transferencia Máxima
- Dirección IP de los servidores NIS *(Network Information Service)*
- Dirección IP de los servidores NTP *(Network Time Protocol)*
- Dirección IP del servidor SMTP *(Simple Mail Transfer Protocol)*
- Dirección IP del servidor TFTP *(Trivial File Transfer Protocol)*
- Nombre del servidor WINS

Configuración de DHCP

La configuración del programa servidor de DHCP en la plataforma operativa de Linux es muy sencilla. Su base está fundamentalmente en el archivo de configuración ubicado en el *path* `/etc/dhcpd.conf`. La configuración es bastante intuitiva.
Antes que nada hay que asegurar que el paquete correspondiente al servicio DHCP esté instalado. Esto puede realizarse mediante la siguiente línea de comando:

```
# rpm -qa dhcp*
dhcp-3.0.1-58.EL4
dhcpv6-0.10-14_EL4
dhcpv6_client-0.10-14_EL4
```

A continuación se detalla un ejemplo de este archivo de configuración:

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#
max-lease-time 3600;
ddns-update-style none;
```

```
default-lease-time 600;
subnet 10.0.0.0 netmask 255.255.255.0 {
option routers 10.0.0.1;
option domain-name-servers 10.0.0.1;
option subnet-mask 255.255.255.0;
range 10.0.0.100 10.0.0.200;
host cpazmino {
        option host-name "cpazmino.machangarasoft.com";
        hardware ethernet 00:09:6B:8D:6A:86;
        fixed-address 10.0.0.101;
}
host ppullas {
        option host-name "ppullas.machangarasoft.com";
        hardware ethernet 00:0D:60:2E:23:D5;
        fixed-address 10.0.0.102;
}
host criptex {
        option host-name "criptex.machangarasoft.com";
        hardware ethernet 00:16:17:49:BB:98;
        fixed-address 10.0.0.103;
}
}
```

Ahora se analizan una a una las opciones detalladas en el archivo de configuración y que tiene mayor importancia:

```
max-lease-time 3600;
```

Este parámetro indica que los clientes podrán mantener los datos de configuración que les ha asignado durante 3600 segundos hasta que una nueva petición se haga. En el instante en que una nueva petición se ejecute, este tiempo se asignará nuevamente.

```
ddns-update-style none;
```

Especifica el método de actualizaciones dinámicas del servidor DNS.

```
default-lease-time 600;
```

Es el tiempo por defecto que los parámetros de configuración se asignarán a los clientes de la *intranet*.

```
subnet 10.0.0.0 netmask 255.255.255.0
```

Especifica la dirección correspondiente a la red a la que se asignarán los parámetros de configuración de red, y la máscara correspondiente de red.

```
option routers 10.0.0.1;
```

Indica la dirección IP del *gateway(*s) por defecto que se asignará a los equipos para que puedan tener acceso a Internet desde la red interna a través del *router*.

```
option domain-name-servers 10.0.0.1;
```

Esta opción detalla la(s) dirección(es) IP del (los) servidor(es) DNS de la red interna, los que se encargarán de atender las peticiones de resolución de nombres de dominio, o de redireccionar estas peticiones a servidores públicos en Internet.

```
option subnet-mask 255.255.255.0;
```

Esta opción indica la máscara de red con que se configurarán automáticamente los clientes de la *intranet.*

```
range 10.0.0.100 10.0.0.200;
```

Indica el rango de direcciones IP dentro de nuestra red dentro del cual están todas las direcciones IP que se asignarán de forma dinámica a los clientes.

```
host cpazmino {
        option host-name "cpazmino.machangarasoft.com";
        hardware ethernet 00:09:6B:8D:6A:86;
        fixed-address 10.0.0.101;
}
```

Este párrafo permite configurar una máquina con una dirección IP estática, reconociéndola a través de su dirección física. Se debe especificarel nombre del *host*, la dirección física de la tarjeta de red y la dirección IP física que siempre se le asignará.

Es de esencial importancia la configuración de los parámetros que específican la puerte de enlace predeterminada y la dirección de los DNS pues si uno de estos no es asignado correctamente, los usuarios internos no podrán acceder a Internet.

Activación del servicio DHCP

Luego de configurar correctamente el archivo dhcpd.conf, es necesario iniciar o reiniciar el servicio de modo que la nueva configuración tenga efecto. Esto se logra mediante el siguiente comando:

```
# service dhcpd restart
```

DNS *(Domain Name System)*
Este protocolo permite mapear un nombre de dominio a su correspondiente dirección IP, entre otra información como la dirección IP del servidor de correo o el de FTP. Esto resulta de bastante utilidad puesto que generalmente es mucho más sencillo recordar un nombre que una dirección IP, sin tomar en cuenta que es mucho más fiable dado que es más común que cambie la dirección IP que el nombre.
El momento en que se registra un dominio, a ese registro se hace correspondiente la dirección IP de los servidores que se encargarán de resolver el dominio. Esta información se guarda en una base de datos del "Internet Network Information Center", pero debido a que es imposible que un solo servidor se encargue de atender las peticiones de cientos de millones de usuarios, existe un mecanismo de propagación de estos registros que hace posible que varios servidores de DNS atiendan este tipo de peticiones.

Introducción
El Parque Tecnológico MachángaraSoft está conformado por empresas jóvenes y por tanto, la totalidad de ellas hospedan sus sitios web y servicios de correo en servidores contratados para ese efecto y que se encuentran en algún lugar del Internet. Esto implica también que los servicios de resolución de nombres (DNS) para la *intranet* se encuentren también alojados en el Internet. De hecho, las direcciones IP de los servidores DNS son asignadas por parte del ISP (SATNet) y corresponden a servidores propios de esta empresa.
Debido a que la corporación no cuenta aún con servidores públicos en su *intranet*, en realidad no es necesario que exista un servidor DNS dentro de ella, aún cuando puede ser útil asignar nombres de *host* a cada máquina dentro del parque y mapearlos con sus correspondientes direcciones IP. La corporación MachángaraSoft tiene su propio dominio: *machangarasoft.com* y dentro de www.machangarasoft.com tiene alojado su sitio web.

Las PYMES (Pequeñas y Medianas Industrias) inicialmente no tienen la necesidad de involucrar servicios sino internos de DNS. Esto es quizá crear un dominio no registrado de manera que los usuarios de la LAN puedan resolver adecuadamente los *hosts* dentro de la empresa.

Configuración de servicio DNS

Como se mencionó anteriormente, el alcance de la configuración del servidor que se plantea es de *redireccionar* las peticiones DNS que no se puedan resolver internamente.
Para iniciar, el servidor DHCP deberá asignar como servidor de Dominio a la máquina que configuraremos como DNS, en este caso asumiremos que la dirección IP de esta PC será 10.0.0.1/24.
Ahora, cualquier petición de resolución correspondiente a un host en Internet se dirijirá efectivamente a nuestro servidor DNS pero este simplemente reenviará la petición a servidores externos que se conozca y que siempre estén activos. Las peticiones de resolución del dominio interno irán también a la interfaz del servidor DNS pero como se trata de un dominio privado, el servidor dispondrá de su mapeo correspondiente y resolverá la petición.

Nota.- La configuración se orientará al manejo de la herramienta de administración gráfica *Webmin,* ya que simplifica enormemente el proceso, especialmente en cuanto a lo referente a DNS por lo complejo de los registros.

Se descarga el paquete *rpm* de Webmin mediante el siguiente comando:

```
# wget -c http://prdownloads.sourceforge.net/webadmin/webmin-1.330-1.noarch.rpm
```

Se instala el paquete:

```
# rpm -ihv webmin-1.330-1.noarch.rpm
```

Se hace *login* a través del *browser*, apuntando en éste la siguiente dirección:

http://127.0.0.1:10000

Se coloca el nombre de usuario y la contraseña del administrador y se accede al panel de administración del sistema operativo.

En la lista de íconos superior se da click en **Servers** y dentro de la nueva ventana que aprarece click en **BIND DNS Server** para configurarlas zonas respectivas.

Dentro de la nueva ventana que aparece, en la parte inferior se ubica la sección **Existing DNS Zones** y se crea una nueva Zona Maestra dando click en **Create MasterZone**. La nueva ventana qque aparece es similar a la *Figura 1.*

En esta ventana, en el primer campo de **Domain Name/Network** se ubica el nombre del dominio que vamos a resolver, por ejemplo **refundation.com**.

El siguiente campo de importancia es **Master Server** en el que se registra la dirección IP o el nombre del host que será el servidor DNS de ese dominio, generalmente es el nombre de la máquina en la que instalamos el Webmin a menos que estemos configurando el servidor de manera remota. En el ejemplo, nuestro servidor es **santox.refundation.com**.

Por último, sería conveniente registrar una dirección de correo electrónico a la que llegue cualquier tipo de advertencia.

Los parámetros de temporización pueden dejarse por defecto y funcionarán sin problema alguno.

Se da click en **Create** para crear la Zona Maestra.

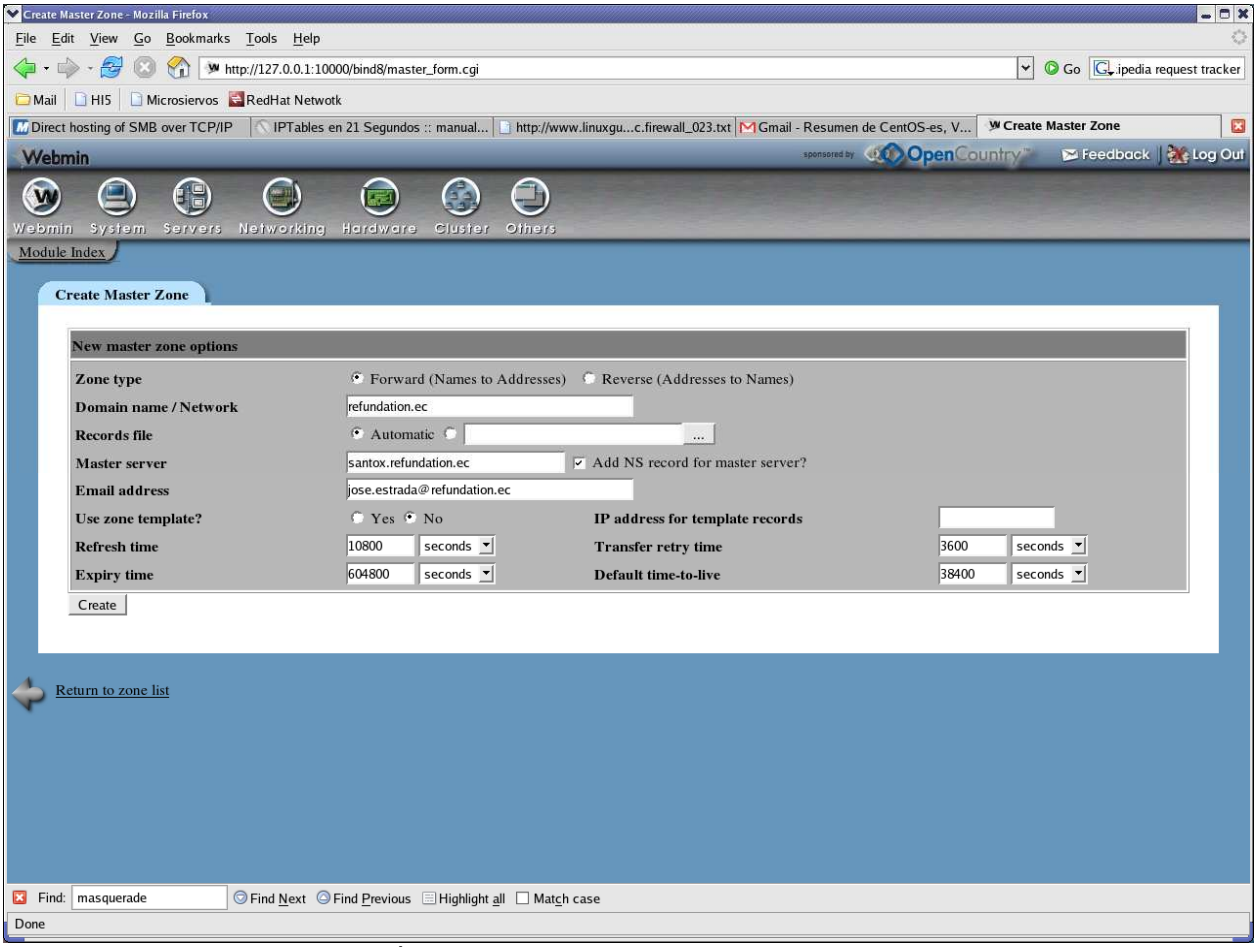


Figura 1. Ventana de Creación de Zona Maestra DNS

Se regresa a la Sección **Existing DNS Zones** y ahora es visible la zona que se creó. Se da click sobre ella para configurar algunos de sus registros. Entonces aparece una nueva ventana que incluye, entre otros, registros **Address** y registros **NS.** Por el momento es importante solamente el registro **Address** que contendrá el mapeo de resolución de nombre de host a dirección IP.

Se da click en el registro A (de Address) y aparece la ventana correspondiente a **Address Records.** En esa ventana se llena el campo **Name** con el nombre del host y el campo **Address** con la dirección IP a la que queremos ligar el nombre de host.
En la *Figura 2* se observa la forma como quedaría la zona **refundation.ec** con tres registros A.

Es necesario configurar también la zona reversa corresponiente a la Zona Maestra que se creó. Esta zona reversa registra el mapeo de dirección IP a nombre de host y permite resolver un nombre de host dada su dirección IP.

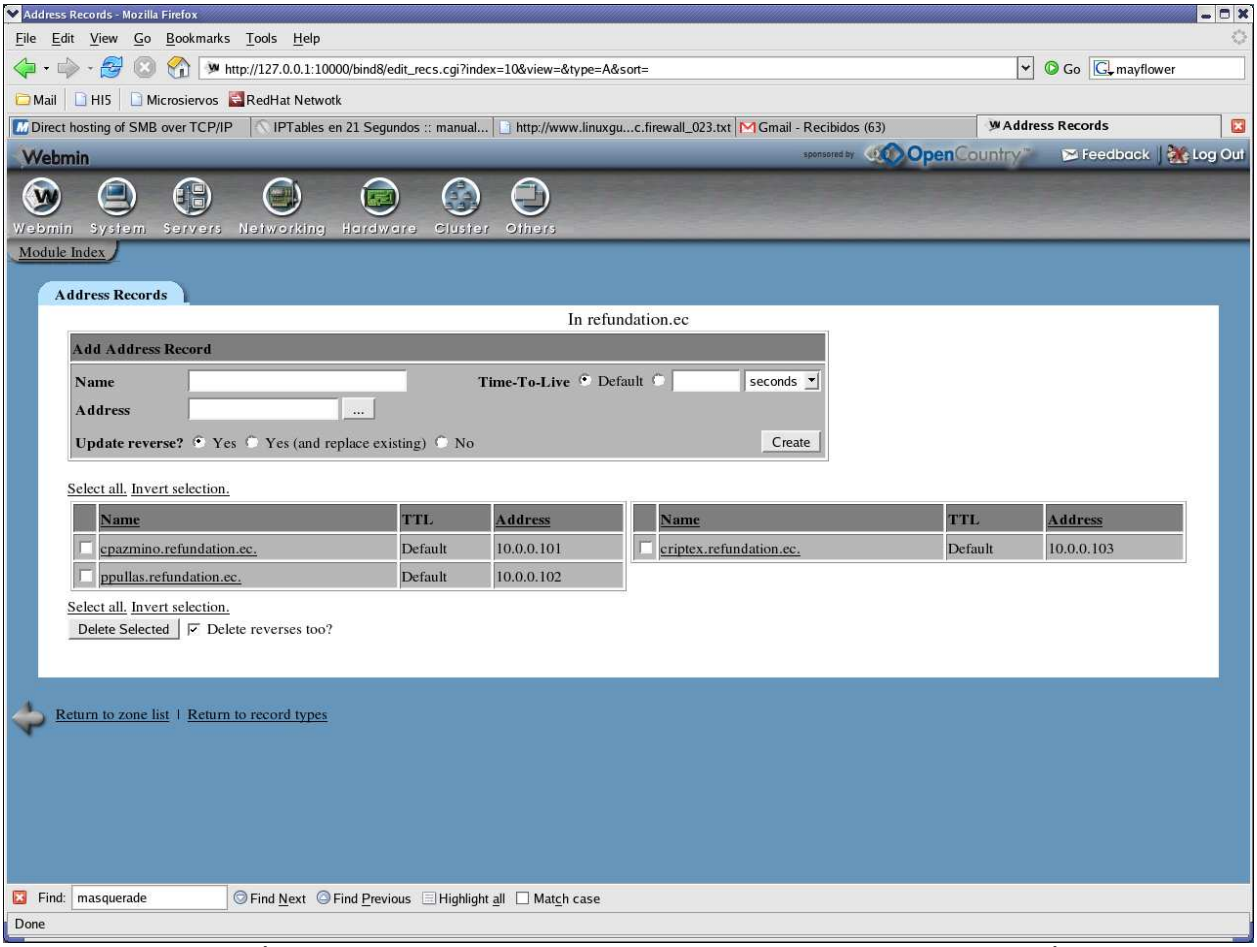


Figura 2. Creación de registros *Address en la Zona Maestra* (Nombre de Host a Dirección IP)

El proceso de creación de Zonas reversas es similar al de creación de una Zona Maestra. En la ventana del Servidor **BIND DNS** nuevamente en la sección **Existing DNS Zones**, se da click en **Create a MasterZone** como se realizó anteriormente.
En la ventana que aparece ahora hay que tener cuidado en seleccionar el **Tipo de Zona** en **Reverse.**

Los campos importantes son los mismos que para la zona maestra. Solamente hay que tener cuidado en el campo **Domain Name/Network** en el que se registrará la dirección IP de la red interna a la que se dará el servicio de resolución reversa. En el ejemplo, la dirección correspondiente de la red sería **10.0.0,** el resto de campos y los parámetros de temporización pueden quedar igual que en el caso anterior.

Se crea la zona Reversa dando click en el botón inferior **Create** y aparece una nueva ventana en la que se ubican algunos registros como **NS** y **PT**. En este último se ubicará el mapeo perteneciente a la resolución Dirección IP a nombre de host.

Al dar click en el botón correspondiente al Registro PT se obtiene una ventana en la que se llenan los campos correspondientes a la dirección IP del host y su nombre respectivo.
La *Figura 3* muestra algunos registros creados, correspondientes a la Zona maestra del ejemplo al que se refiere la presente guía.

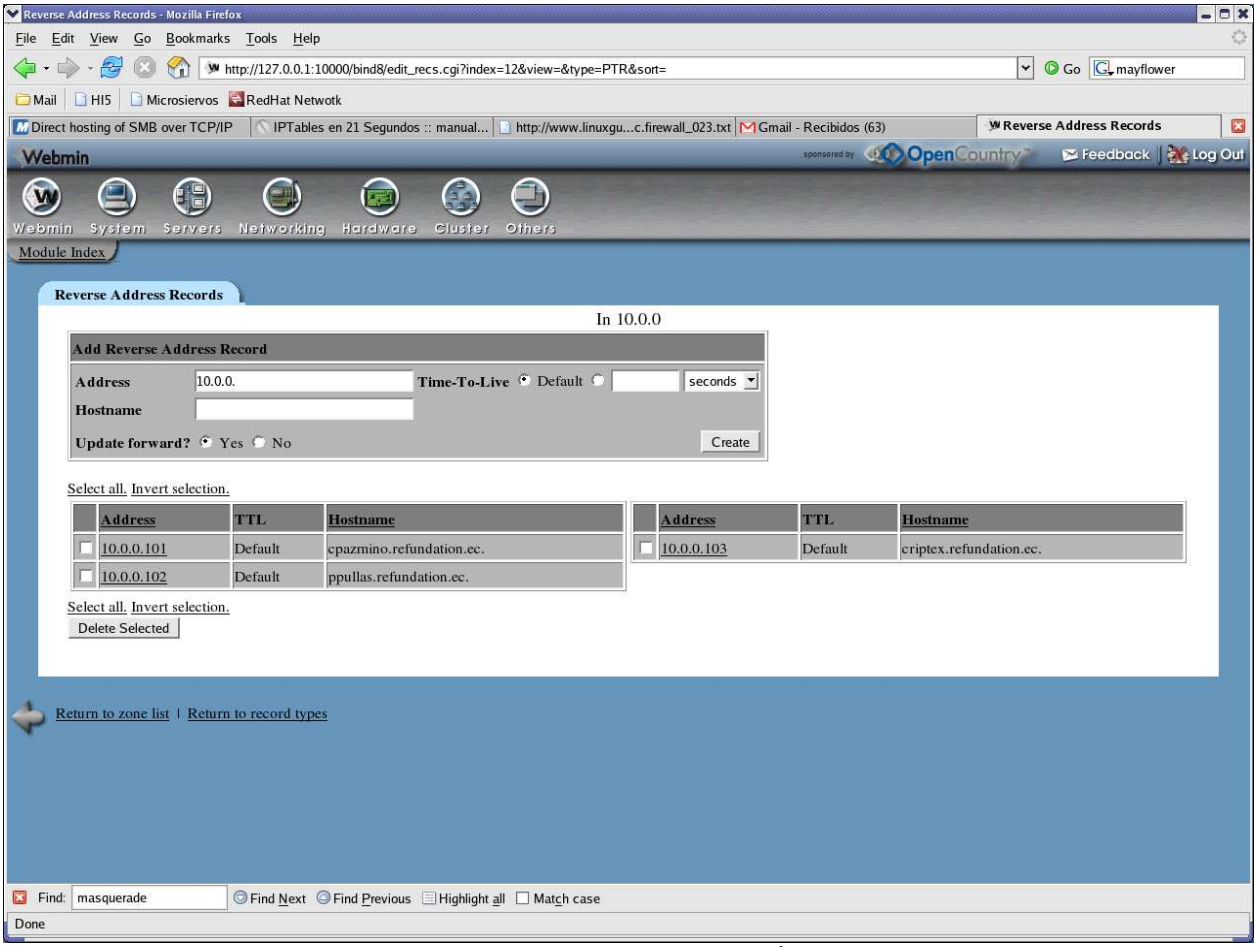


Figura 3. Ingreso de Registros PT en la Zona Reversa (Dirección Ip a nombre de host)

Luego de configurar nuestra zona reversa, en la ventana principal del Servidor BIND DNS de Webmin es posible Iniciar el servicio. Se puede dar click allí o simplmente en la línea de comandos escribir:

```
# service named start
```

Ahora, para comprar la resolución de nuestro servidor interno DNS se utiliza por ejemplo el comando **host.** Si estamos en la máquina cuyo nombre es criptex.refundation.com y se desea comprobar la dirección IP de la máquina cuyo dueño es Christian Pazmiño podría escribir en la línea de comandos:

```
[root@criptex ~]# host cpazmino.refundation.ec
cpazmino.refundation.ec has address 10.0.0.101
```

Se observa que el servidor responde que cpamino.refundation.com posee la dirección IP 10.0.0.101, lo que es correcto.
Se puede comprobar la resolución reversa con el mismo comando:

```
[root@criptex ~]# host 10.0.0.101
101.0.0.10.in-addr.arpa domain name pointer cpazmino.refundation.ec.
```

El servidor responde que el nombre de host correspondiente a la dirección IP 10.0.0.101 es cpazmino.refundation.ec, que es lo que se esperaba.

Importante: Es importante recordar que para que las peticiones DNS se dirijan hacia el servidor que se ha seteado, es necesario configurar los clientes para que direccionen sus peticiones adecuadamente hacia el servidor que interesa.
En plataformas operativas Linux y Windows, esto puede configurarse automáticamente vía DHCP o de manera estática.

REDIRECCIÓN DE PETICIONES
Tal y como se mencionó, el servidor, por el momento será capaz de resolver únicamente las peticiones DNS relacionadas con los registros anotados anteriormente y que corresponden a una *intranet.* Es necesario, sin embargo, resolver también peticiones de nombres de host ubicados en Internet, de modo que los usuarios puedan navegar adecuadamente. Esto se logrará en este caso redireccionando estas peticiones hacia servidores DNS conocidos en Internet, a través de una pequeña modificación en un archivo de configiración.
El archivo de configuración es **/etc/named.conf.**

Se abre el archivo con el comando:

```
# vi /etc/named.conf
```

En la sección **Options** se agregan las líneas siguientes:

```
forward only;
        forwarders { 200.63.212.110; 200.25.144.1; };
```

Lo que indica que las peticiones que no puedan resolverse a través de los registros existentes en el servidor DNS, serán reenviadas a los servidores cuyas direcciones IP se especifican.

Luego de cambiar este archivo, será necesario reiniciar el servicio *named* con el comando:

```
# service named restart
```

Para comprobar el reenvío de peticiones hacia Internet bastará con utilizar el comando **host,** como se hizo anteriormente, para resolver cualquier dominio conocido.

```
[root@criptex ~]# host www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 72.14.209.104
www.l.google.com has address 72.14.209.99
```

Servidor Proxy en Linux

Un servidor *proxy* es un dispositivo o un programa que permite la conexión de toda una red privada a Internet a través de una única conexión (dirección IP) disponible de salida.
El servicio de proxificación se encarga de aceptar peticiones de conexión generalmente en el puerto 8080 de la capa de transporte de la pila TCP/IP y de reenviar esas peticiones hacia el Internet. Asimismo, redirecciona los paquetes de respuesta desde Internet hacia la red interna.
No es necesario que el servidor de proxificación esté configurado para enrutar paquetes y esto implica un mayor nivel de seguridad en el sentido de que se establecen conexiones independientes desde la intranet hacia el servidor proxy, a través de la interfaz interna, y desde Internet al proxy a través de la intefaz pública externa. El servicio de *proxy* más común sobre plataformas operativas basadas en Linux es *Squid*.

Instalación y Configuración del Servicio de Proxy Squid en Linux
El paquete de *Squid* se encuentra por defecto en la mayoría de distribuciones de Linux.
Para la instalación se puede ejecutar el siguiente comando, una vez que este haya sido localizado.

```
rpm -ihv squid-2.5.STABLE6-3.4E.12.rpm
```
(Válido para la versión 4.3 de CentOS)

El archivo de configuración para el servicio de proxificación en el sistema operativo Linux es `/etc/squid/squid.conf`.
Un ejemplo funcional de configuración del archivo mencionado podría ser el siguiente:

```
http_port 10.0.0.1:8080
icp_port 0
cache_mem 32 MB
cache_swap_low 90
cache_swap_high 95
maximum_object_size 8192
ipcache_size 2048
ipcache_low 90
ipcache_high 95
fqdncache_size 2048
cache_store_log /var/log/squid/store.log
#cache_dir /var/spool/squid
cache_dir ufs /var/spool/squid 700 16 256
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
pid_filename /var/run/squid.pid
acl all src 0.0.0.0/0.0.0.0
acl allowed_hosts src 10.0.0.0/255.255.255.0
http_access allow allowed_hosts
http_access deny all
icp_access deny all
miss_access allow all
cache_effective_user squid
cache_effective_group squid
snmp_port 0
```

```
http_port
```
En la primera línea, indica la direeción IP de la interfaz en la que se escucharán las peticiones dirigidas y el puerto en el que el servidor lo hará. En este caso se indica una interfaz con dirección IP 10.0.0.1 y un puerto de recepción de peticiones como el 8080.

```
icp_port 0
```

Se indica el número de puerto en el que Squid envía y recibe consultas ICP (*Internet Cache Protocol*). El valor por defecto es 3130 y para deshabilitarlo se usa el valor de 0.

```
cache_mem 32MB
```
Este parámetro no especifica la cantidad máxima de memoria asignada para los procesos de Squid sino que simplemente establece un límite en referencia a cuánta memoria adicional usará Squid como caché de objetos.

```
cache_swap_low 90
```
Este parámetro señala el nivel en porcentaje de capacidad mínima aceptada por Squid, es decir, los objetos se mantendrán en el cache hasta que se cope el límite mínimo.

```
cache_swap_high 95
```
Parámetro que especifica en porcentaje el límite máximo que utiliza Squid para mantener objetos en el cache. Si el valor asignado es del 95%, squid comenzará a eliminar los objetos del caché cuando se tope el 95% de la capacidad asignada a squid.

```
maximum_object_size 8192
```
Este parámetro, especificado en KB, indica el tamaño máximo que se almacena en el cache. Por defecto se utiliza 4MB.

```
ipcache_size 2048
```
Identifica a la tabla de direcciones IP que Squid ha utilizado, tanto las exitosas (encontradas, con datos) como las negativas (no encontradas).
Cada dirección IP ocupa unos pocos bytes, de manera que aumentar o reducir esta tabla tiene poco impacto en el *pool* total.

```
ipcache_low 90
```
Especifica la marca de agua inferior de cacheo de dirreciones IP.

```
ipcache_high 95
```
Especifica la marca de agua superior de cacheo de direcciones IP.

```
fqdncache_size 2048
```
Especifica el número máximo de entradas FQDN en caché.

```
cache_store_log /var/log/squid/store.log
```
Este parámetro especifica la ubicación del archivo de registro de objetos sacados del caché. No es necesario activarlo. Es mejor desactivarlo para ahorrar espacio en disco.

```
cache_dir ufs /var/spool/squid 700 16 256
```
Especifica el directorio de ubicación del cache, por defecto /usr/local/squid/cache. Este parámetro incluye tres parámetros numéricos adicionales. El primero incluye el **número de MB que se utilizarán en este directorio para el cache, por defecto 100MB**, el segundo el número de directorios a utilizar en el primer nivel (16 por defecto) y el tercero el número de subdirectorios en el segundo nivel (256 por defecto).

```
cache_access_log /var/log/squid/access.log
```
Especifica en que directorio se realizará el registro de accesos al squid. Este parámetro es importante para definir posteriormente en el sistema de análisis de estadísticas SARG, la ubicación del registro de accesos.

```
cache_log /var/log/squid/cache.log
```
Define en donde se almacenan los mensajes del sistema.

```
pid_filename /var/run/squid.pid
```
Define la ubicación del archivo squid.pid, se utiliza el valor por defecto `/usr/local/squid/logs/squid.pid.`

```
acl all src 0.0.0.0/0.0.0.0
```
Lista de control de acceso que permite las peticiones de proxificación desde todas las fuentes.

```
acl allowed_hosts src 10.0.0.0/255.255.255.0
```
Lista de control de acceso que especifica los host origen válidos para las peticiones a Squid.

```
http_access allow allowed_hosts
```
Indica que se acepten las peticiones para los hosts permitidos.

```
http_access deny all
```
Línea de acceso que deniega cualquier petición restante.

```
icp_access deny all
```
Descarta cualquier petición *ICP* (*Internet Cache Protocol*)

```
cache_effective_user squid
```
Definen qué usuario (*user*) ejecuta Squid. La versión de RPM utiliza el usuario squid y grupo squid. La versión de las fuentes utiliza el usuario nobody y grupo nogroup.

```
cache_effective_group squid
```
Define qué usuario (*user*) y grupo (*group*) ejecuta squid. La versión de RPM utiliza el usuario squid y grupo squid. La versión de las fuentes utiliza el usuario nobody y grupo nogroup

```
snmp_port 0
```
Permite obtener información de estadísticas y estado mediante SNMP. Un valor de 0 deshabilita el soporte SNMP. Si se desea utilizarlo, se debe setear a 3401 para usar el soporte normal de este protocolo.

Todos los anteriores parámetros son básicamente los más importantes aún cuando se pueden configurar muchas opciones más.

En el caso de que se desee configurar un sistema de proxy transparente, es decir que los usuarios de la red tengan acceso al servicio sin tener que configurar sus navegadores, se deben agregar las opciones siguientes, además por supuesto de las reglas correspondientes de redirección de puertos en el firewall.

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Esta configuración es válida si lo que se requiere es redireccionar las peticiones de navegación al puerto 80 (puerto en el que normalmente se hacen las peticiones HTTP) hacia el puerto 8080 (de Squid), de modo que sean administradas por nuestro servidor de acceso.

Utilización de Listas de Control de Acceso

Se podría juzgar a Squid como un sistema de *firewall* de capa de aplicación pues, a diferencia de *iptables*, permite filtrar el acceso HTTP en base al URL, archivos o apicaciones, extensiones, contenido, etc. Esto significa un alto nivel de control de acceso

que incluso puede extenderse al filtrado en función de horarios y períodos de tiempo determinados lo que reflejaría un significativo ahorro de ancho de banda.

La sintaxis para establecer estas reglas de control a nivel de aplicación se resumen en: la *definición de la regla* y la *aplicación de la misma*.

Definición

```
acl [nombre de la lista] src [lo que compone a la lista]
```

Por ejemplo si se quiere definir una regla que impida el acceso a Internet a determinados usuarios, definiríamos la regla más o menos así:

```
acl no_permitidos src "/etc/squid/no_permitidos"
```

Esta línea nos indica que se crea una lista de control de acceso llamada `no_permitidos` y que tomará la información contenida en el archivo `/etc/squid/no_permitidos`.
El fichero mencionado podría contener por ejemplo:

```
192.168.1.100
192.168.1.101
192.168.1.102
```

Si se deseara referirse como no_permitidos a estas direcciones.

Si, por ejemplo es necesario crear una lista de control de acceso que abarque toda una red, se podría escribir lo siguiente:

```
acl no_permitidos src 192.168.0.0/255.255.255.0
```

Aplicación
Luego de definir la lista, hay que aplicarla, es decir determinar las acciones a tomarse: denegar (`deny`) o aceptar (`accept`) una conexión y, en este sentido, se puede utilizar como criterios de decisión los siguientes:

- La dirección o conjunto de direcciones de las que salen las solicitudes de acceso web
- La dirección o conjunto de direcciones de las que **no** salen las solicitudes.
- La extensión de los archivos accedidos.
- La URL (*Unified Resource Locator*) a la que acceden los usuarios.
- Mediante autenticación a través de un fichero de texto simple con claves de acceso creadas.
- El horario de acceso.

Se puede observar que el nivel de control es sumamente granular, detallado y tiene muchas posibilidades.
La sintaxis correspondiente a la aplicación de las listas sería la siguiente:

```
http_access [deny o allow] [lista de control de acceso]
```

Un ejemplo de aplicación de la definición de lista de acceso mostrada anteriormente sería:

```
http_access deny no_permitidos
```

Esta lista indica que se denegará el acceso a Internet de todos los *hosts* cuya dirección IP están contenidas en la lista `no_permitidos`.

Luego de cambiar los parámetros indicados, será necesario reiniciar el *demonio* de proxy mediante el comando siguiente:

```
[root@gateway]# service squid restart
```

Como se mencionó anteriormente, existen varios criterios en función de los cuales es posible controlar el acceso a páginas en Internet, y estas posibilidades se multiplican con la capacidad que tiene Squid de combinar estos criterios.

Monitoreo de Acceso Web mediante SARG y Control de Acceso con Squid

SARG proviene de *Squid Analysis Report Generator* (Generador de Reportes de Análisis de Squid). Es una herramienta que permite monitorear el acceso a páginas web en Internet. Es capaz de desplegar reportes o estadísticas de acceso a páginas web segmentando la información por usuario (según la dirección IP), fecha, hora y por URL.

Esta herramienta despliega los logs generados en el archivo /var/log/squid/access.log generados por el proxy *Squid* de forma sumamente amigable y fácil de interpretar a través del servicio web *httpd* de Linux.
Entre las opciones de clasificación de reportes están:

- Detalle de los sitios más visitados
- Detalle de sitios visitados por usuario
- Registro de descargas
- Registro de intentos de acceso denegados

Esta herramienta resulta de mucha utilidad si se desea determinar políticas de acceso a Internet de la empresa, a partir de las conclusiones que se puedan obtener del análisis de los reportes obtenidos.

Instalación
Como se mencionó, SARG es un intérprete de los *logs* de acceso generados por *Squid* y su instalación se basará en algunos parámetros relacionados con este proxy, así como de otros servicios como el servidor web para el despliegue de los reportes.
La instalación de esta aplicación, con un sistema de repositorios en CentOS actualizado no representará ninguna dificultad al hacerlo con la herramienta **yum**. Para ello se ejecutará el siguiente comando:

```
# yum install sarg
```

Una vez que se ha instalado, hay que asegurar en el archivo `/etc/sarg/sarg.conf` la existencia de los siguientes parámetros y sus valores:

```
language Spanish                          #Configura el Idioma
access_log /var/log/squid/access.log      #Archivo de logs del Squid
temporary_dir /tmp                        #Directorio temporal
output_dir /var/www/html/squid-reports    #Archivo de acceso a la interfaz web de
reportes
```

Se debe crear por supuesto el directorio `/var/www/html/squid-reports`.
Ahora se debe editar el archivo /etc/httpd/conf/httpd.conf de modo que se pueda acceder a través del servicio web al archivo que se creó, agregando estas líneas al final del archivo.

```
<Directory                                  "/var/www/html/squid-reports">
AuthName                          "Squid                          Reports"
AuthType                                                            Basic
AuthUserFile                                  /etc/.htpasswdsquidreports
require                                                       valid-user
Options                       Indexes                    FollowSymLinks
AllowOverride                                                       None
Order                                                         allow,deny
Allow                               from                             all
</Directory>
```

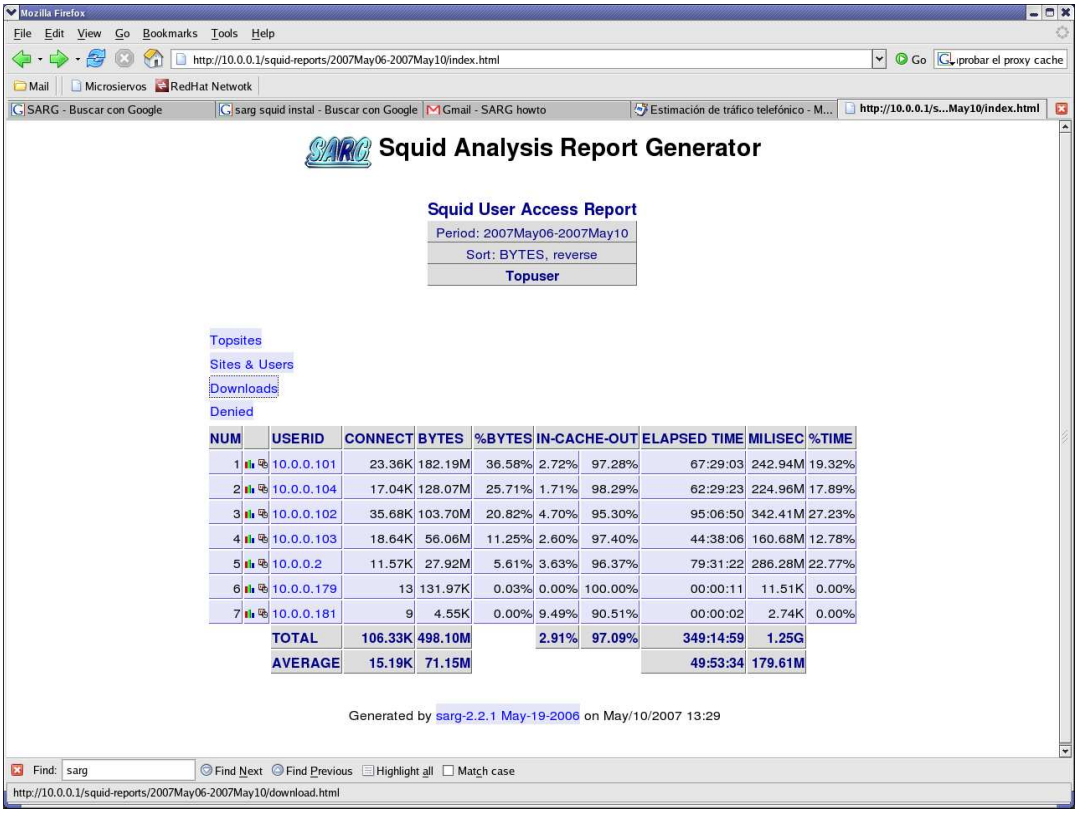
Una vez hecho esto sólo resta reiniciar el servicio web y el correspondiente al *Squid*, utilizando los comandos:

```
#service httpd restart
#service squid restart
```

Cada vez que se requiera un reporte se debe ejecutar el comando **sarg**, luego de lo que se podrá tener acceso a éste apuntando en un *browser* mediante la siguiente URL:

http://localhost/squid-reports

Si se desea tener acceso de manera remota sólo habrá que cambiar localhost por la dirección IP del servidor proxy o su nombre de host, y se desplegará una página parecida a la de la figura.



Dando click en cada dirección IP se obtendrá los accesos a páginas web realizados a través del proxy.

Una muy buena política de seguridad, aunque no la única, en relación al control de acceso web, y que favorece indirectamente al control del ancho de banda es la generación de reglas de acceso a Internet, ejecutando un control por cliente y por horario en función de la jerarquía de los clientes del Squid y las potestades estrictas que esta jerarquía puede adjudicar.
Esto significa, por ejemplo, que no tiene sentido que la estación perteneciente a una empleada que trabaja únicamente durante las mañanas, tenga acceso a Internet durante todo el día. Esta posibilidad supone que alguien ajeno a ese puesto de trabajo podría conectarse en horasde la tarde y consumir recursos que no deberían estar disponibles, evitando que se pueda aprovechar adecuadamente la capacidad de ancho de banda de la organización.

De acuerdo a las indicaciones que se especifican anteriormente en el proceso de configuración de políticas de acceso para *Squid* es posible definir listas de control de acceso (en el archivo `/etc/squid/squid.conf`) en base a horas y días de la semana para los horarios de la tarde, la mañana, horario completo y algunos horarios especiales.

```
acl horario time MTWHFAS 01:00-24:00
acl horario_tarde time MTWHF 08:00-14:00
acl horario_manana time MTWHF 13:00-19:00
acl horario_especial time MTWHF 07:00-22:00
acl dialaboral time 08:30-19:30
```

Asimismo es consecuente definir listas de acceso de acuerdo a la jerarquía y los niveles en la organización, en base a la lectura de ciertos archivos que contendrán las direcciones IP de las estaciones pertenecientes a los miembros de determinado grupo.

```
acl permitidos src "/etc/squid/permitidos"
acl gerencia src "/etc/squid/gerencia"
acl permitidos_manana src "/etc/squid/permitidos_manana"
acl permitidos_tarde src "/etc/squid/permitidos_tarde"
acl permitidos_especial src "/etc/squid/permitidos_especial"
```

Por ejemplo, la dirección IP de la empleada que se mencionaba anteriormente, estará incuida en el archivo `permitidos_manana` y en ningún otro, de manera que esa estació no tenga acceso al proxy en ningún otro período que no sea ese.

Finalmente se procederá a la aplicación de estas reglas.

```
http_access deny permitidos_manana horario_manana
http_access deny permitidos_tarde horario_tarde
http_access deny permitidos_especial horario_especial
```

Los cambios en el archivo se harán efetivos al reiniciar el servicio squid.

```
#service squid restart
```

VPN en Linux

VPN (*Virtual Private Network* – Red Privada Virtual)

La Red Privada Virtual es una tecnología de red que permite comunicar dos o más redes privadas o locales distantes en el espacio, como si estuvieran dentro del mismo segmento de red, es decir, virtualmente. Esta interconexión utiliza como vínculo a Internet, pero debido al hecho de que éste es un ambiente hostil se requiere implementar mecanismos de seguridad y encriptación de la información de tal manera que se garantice la integridad de los datos.

Esta infraestructura permite ahorrar gastos pues utiliza el Internet como medio de conexión y transporte, a diferencia de enlaces dedicados entre sucursales cuyos costos son sumamente elevados.
Fundamentalmente, esta tecnología de conectividad tiene 3 requerimientos básicos:

- *Autenticación de usuarios*
- *Encriptación de información*
- *Soporte a múltiples protocolos*

Existe una amplia gama de modos de conexión: Host a Host, Host a Red y Red a Red. Este último modelo de configuración es el más complejo puesto que involucra a los dos primeros.

Estructura
La **Red Privada Virtual *Red a Red*** permite conectar redes (sucursales) remotas de una organización con una sede central y, desde luego, su red interna. La estructura involucra un servidor central de VPN con una conexión permanente a Internet mediante un proveedor cualquiera que recibe peticiones de clientes VPN correspondientes a las estaciones de *intranet* de otras sucursales.
Toda la información que se transmite a nivel de Internet está generalmente sin cifrar y esto constituye un riesgo enorme en el sentido de que cualquier entidad con acceso a nuestra red podrá leer estos datos.
Una técnica que permite asegurar esta transferencia de información es la de *tunneling*. Esta técnica establece conexiones entre dos máquinas a través de un protocolo seguro (se activa el túnel) y a través de ella se envía la información.
Las VPNs utilizan este mecanismo de *tunneling* para proteger la información y dotarle de confidencialidad.
Existen soluciones de hardware y software y aún cuando las primeras brindan mayor confiabilidad y rendimiento, las últimas gozan de una flexibilidad y adaptabilidad envidiable.

En la mayoría de los casos, los paquetes no se originan en los *routers* o puntos de acceso a Internet sino más bien en las redes privadas que están detrás de ellos. Estas redes se pueden alcanzar entre sí a través del *túnel* que se establece previamente entre los puntos de acceso correspondientes a cada una de ellas.

Escenario de ejemplo
Se puede imaginar una empresa con dos sucursales, una principal y una secundaria. Cada una de estas sucursales goza de una conexión a Internet de banda ancha a través de un proveedor cualquiera. Además, en cada una de las dos oficinas existe un *firewall-box* o *router* en base a una estación con plataforma operativa Linux, a través de la cual las redes internas (correspondientes a cada sucursal) obtienen servicio de *firewall* y *proxy* fundamentalmente. Se supone además que se tiene acceso a la configuración de las reglas del *firewall* pues será necesario modificar algunos aspectos del enrutamiento desde y hacia Internet.
De manera gráfica, esto luciría más o menos así:

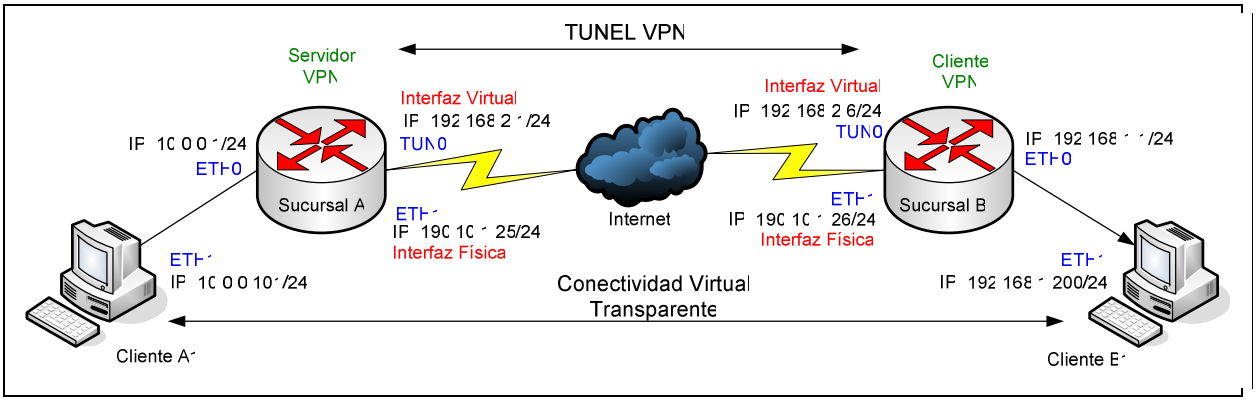


Figura. Esquema de conexión VPN entre dos sucursales

El objetivo que se ilustra es permitir que los clientes de la Sucursal A puedan comunicarse directamente con las estaciones de la Sucursal B, más claramente, que el cliente B1 pueda, por ejemplo hacer *ping* al Cliente A1 sin que se configure directamente un mecanismo común de enrutamiento en los *routers* ni que la información viaje sin cifrar.
Este sistema puede ser muy útil para el administrador de redes que se encuentra en la sucursal principal (A) y quiere acceder remotamente a la *intranet* en la sucursal B sin que tenga un acceso directo a los *routers* en cada una de las oficinas. Podrá, por tanto conectarse vía **ssh** desde el cliente A1 hasta el cliente B1 como si estuvieran en la misma red, de manera muy segura, y además económica, utilizando la infraestructura pública de Internet.
El esquema no nos limita, pues existe la posibilidad de interconectar más sucursales a través del mismo mecanismo, agregando clientes VPN que enviarán las peticiones de conexión hacia el servidor.
Tal como se puede observar en la figura, esta estructura involucra la creación de una interfaz virtual en cada uno de los puntos de red que conformarán el túnel.

Instalación y configuración de una VPN en Linux
Para la configuración de la VPN en el esquema que se mostró con anterioridad, se utilizará una herramienta para plataformas operativas Linux (y también para Windows) llamada OpenVPN.
OpenVPN es una solución que implementa conexiones de capa 2 o 3 y utiliza los estándares SSL/TLS para cifrar la información. Será necesario instalar esta herramienta en el punto de acceso (*router-firewall-proxy*) a Internet de cada una de las sucursales. La principal será configurada como servidor VPN y las restantes se configurarán como clientes.
Se debe tomar en consideración también el tráfico a través de los firewall en cada sucursal, pues si no se modifica adecuadamente el mecanismo de filtrado se tendrá muchos inconvenientes para que la VPN pueda establecerse.

La instalación en el sistema operativo CentOS-RedHat es muy sencilla. Se puede descargar el paquete correspondiente e instalarlo o hacerlo simultáneamente mediante el comando:

```
# yum install openvpn
```

Este último resolverá los problemas de dependencias, instalando todos los paquetes necesarios.
Una vez instalado el software, se puede proceder a la creación de las claves de encriptación tanto en el servidor como en el cliente.

Para *desactivar* y *activar* el servicio de **openvpn** se puede ejecutar lo siguiente:

```
# service openvpn stop
# service openvpn start
# chkconfig openvpn on
```

El último comando permitirá que el servicio se active junto con los otros servicios de inicio al arrancar el sistema operativo.

Luego de que se realicen cambios y si se desea que el servicio lea nuevamente la configuración se puede reiniciar el servicio mediante:

```
# service openvpn restart
```

Para iniciar el proceso de configuración, haya que aclarar que es **necesario** que *los segmentos físicos de cada una de las intranets deben poseer un direccionamiento distinto*.
En la figura de ejemplo se muestra que los segmentos de red son los siguientes:
Sucursal A: 10.0.0.0/25
Sucursal B: 192.168.1.0/25

Una vez que se ha instalado el software mencionado, se creará el *path* `/usr/share/doc/openvpn-2*/easy-rsa` que contiene una serie de *scripts* que serán de mucha ayuda para crear una serie de claves y certificados iniciales que permitan autenticar y encriptar la información entre el servidor y los clientes.
Primeramente se copia este directorio dentro del *path* `/etc/openvpn/` e ingresamos en él:

```
cp -a /usr/share/doc/openvpn-2*/easy-rsa /etc/openvpn
cd /etc/openvpn/easy-rsa
```

Dentro de ese directorio se ejecutan los siguientes comandos:

```
. vars
sh clean-all
sh build-ca
```

Los comandos precedentes permiten inicializar algunas variables de ambiente necesarias para la creación de los certificados, borrar posibles configuraciones anteriores y finalmente generar el Certificado de Autoridad, todo esto en el dispositivo servidor de VPN.
Alguna información solicitada lucirá como sigue:

```
Generating a 1024 bit RSA private key
.....................................................................
..................
.................................++++++.....................++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:EC
State or Province Name (full name) [NA]:Pichincha
Locality Name (eg, city) [BISHKEK]:Quito
Organization Name (eg, company) [OpenVPN-TEST]:MachangaraSoft
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:santox
Email Address [me@myhost.mydomain]:jose.estrada@refundation.com
```

El parámetro que no debe dejarse en blanco es *Common Name*, que representará al Certificado de Autoridad. En este caso, el nombre del *host*: santox.

Ahora se debe generar el certificado y la clave de encriptación para el servidor, a través del siguiente comando:

```
# sh build-key-server server
```

La salida del comando anterior requerirá del ingreso de información similar a la anteriormente agregada acerca del certificado del servidor. El parámetro *Common Name* debe ser diferente al que se especificó anteriormente, en este caso *server:*

```
Generating a 1024 bit RSA private key
.....................++++++
.........................++++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:EC
State or Province Name (full name) [NA]:Pichincha
Locality Name (eg, city) [BISHKEK]:Quito
Organization Name (eg, company) [OpenVPN-TEST]:MachangaraSoft
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:server
Emaase enter the following 'extra' attributes to be sent with your
certificate request A challenge password []
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature Signature ok The Subject's
Distinguished Name is as follows
countryName :PRINTABLE:'EC'
stateOrProvinceName :PRINTABLE:'Pichincha'
localityName :PRINTABLE:'Quito'
organizationName :PRINTABLE:'MachangaraSoft'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'server'
emailAddress :IA5STRING:'jose.estrada@refundation.com'
The stateOrProvinceName field needed to be the same in the CA
certificateil Address
```

Asimismo, deben generarse certificados y claves de seguridad para cada cliente que vaya a acceder a la VPN a través del servidor. Esto se logra mediante el comando:

```
# sh build-key client1
```

El comando debe ejecutarse dentro de */etc/openvpn/easy-rsa* y de él se obtendrá la siguiente salida:

```
Generating a 1024 bit RSA private key
.....................................................................
.....................++++++
......++++++
writing new private key to 'client1.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KG]:EC
State or Province Name (full name) [NA]:Pichincha
Locality Name (eg, city) [BISHKEK]:Quito
Organization Name (eg, company) [OpenVPN-TEST]:EcuaLinux
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:client1
Email Address [me@myhost.mydomain]:info@ecualinux.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'EC'
stateOrProvinceName :PRINTABLE:'Pichincha'
localityName :PRINTABLE:'Quito'
organizationName :PRINTABLE:'MachangaraSoft'
organizationalUnitName:PRINTABLE:'IT'
commonName :PRINTABLE:'client1'
emailAddress :IA5STRING:'jose.estrada@refundation.com'
Certificate is to be certified until Nov 24 05:25:40 2016 GMT (3650 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Esto generará un certificado y una clave para *client1*.
Si es necesario agregar más clientes, se utilizará el comando más o menos así:

```
# sh build-key client2
# sh build-key client3
```

Se debe tener cuidado en especificar el *Common Name* que es el mismo que se indica en el comando para generar los certificados de cada cliente *(client1, client2, client3* en este ejemplo*).*

En el dispositivo servidor se debe generar también un parámetro llamada de ***Diffie-Hellman***:

```
# sh build-dh
```

```
Generating DH parameters, 1024 bit long safe prime, generator 2
```

```
This is going to take a long time

.....................+..............................+..........
```

Todos estos comandos deben ejecutarse en el *path* que se indicó, y los archivos necesarios y correspondientes a cada comando se generarán en */etc/openvpn/easy-rsa/keys.*
Para **el servidor VPN** se deben copiar hacia el directorio `/etc/openvpn` los siguientes archivos:

- ca.crt
- ca.key
- server.key
- server.crt
- dh1024.pem

En cada **cliente VPN** se deberán copiar en el directorio `/etc/openvpn` los siguientes archivos:

- ca.crt
- client**X**.crt
- client**X**.key

Nota: La transferencia de estos archivos debe hacerse de manera segura, utilizando cifrado de la información o algún medio magnético pues se trata de las claves de encriptación de los clientes.

CONFIGURACIÓN
SERVIDOR
La configuración del Servidor VPN se realiza en el archivo `/etc/openvpn/server.conf` al que se debe agregar la siguiente información:

```
#Puerto que utilizara OpenVPN
port 1194
#Protocolo de capa de transporte que utilizara
proto udp
#Dsipositivo de tunel que se crea
dev tun
#Certificado de autoridad
ca ca.crt
#Certificado del servidor
cert server.crt
#Clave de cifrado del servidor
key server.key
dh dh1024.pem
#Direcciones que se asignaran a los
#clientes, el server es .1
server 192.168.2.0 255.255.255.0

ifconfig-pool-persist ipp.txt

#Ruta para que los clientes alcancen la red local del server (56.0/24)
push "route 10.0.0.0 255.255.255.0"

#Directorio donde el servidor buscara configuraciones especiales segun el
cliente
client-config-dir ccd
#Ruta para que el servidor alcance los clientes
route 192.168.1.0 255.255.255.0
client-to-client
push "route 192.168.1.0 255.255.255.0"

keepalive 10 120
comp-lzo
```

```
user nobody
group nobody
persist-key
persist-tun
status openvpn-status.log
verb 4
```

Ahora se debe crear en el servidor, el directorio `/etc/openvpn/ccd` con el comando:

```
# mkdir /etc/openvpn/ccd
```

Dentro de este directorio se crea el archivo *`client1`* que corresponde al cliente *`client1`* y se coloca la información de la ruta estática hacia la *intranet* del cliente 1.

```
iroute 192.168.1.0 255.255.255.0
```

Como en el ejemplo, la *intranet* correspondiente al cliente1 (Sucursal B) tiene un segmento de red 192.168.1.0, especificamos esa ruta para que el túnel pueda llegar desde la *intranet* del servidor hacia la *intranet* del cliente. Si se tuvieran más clientes (sucursales), se debería agregar un archivo para cada uno de ellos en el directorio mencionado.

CLIENTE

Se crea el archivo *`client1.conf`* en el directorio */etc/openvpn* dentro del dispositivo cliente VPN y se edita de la siguiente forma:

```
client
#Dispositivo de tunel
dev tun
proto udp
#Direccion IP:puerto del servidor VPN
remote 190.10.1.25 1194
resolv-retry infinite
nobind
#Las dos siguientes opciones no van en clientes windows
user nobody
group nobody

persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
comp-lzo
verb 4
```

Los parámetros importes son:

`client:` Indica que algunas configuraciones se tomarán del servidor.

`nobind:` establece que el dispositivo solo actúe como cliente

Se debe tomar en cuenta que los certificados (`client1.crt`) y las claves (`client1.key`) deben ser únicos para cada cliente y de esa forma deben generarse en el servidor.

Una vez que se haya configurado tanto servidor como cliente se debe reiniciar el servicio de OpenVPN como se indicó en un inicio, tanto en el servidor como en los clientes.

Al hacerlo, se genera una interfaz virtual de nombre *tun0* en el servidor con la dirección IP 192.168.2.1 como se indicó en el archivo de configuración e igualmente en el cliente pero

con una dirección dinámicamente asignada, en este caso 192.168.2.6. Si se procedió correctamente deberá los *routers* deberán poder alcanzarse entre sí a través de estas interfaces, es decir, si hacemos *ping* desde el servidor a la dirección 192.168.2.6, éste tendrá éxito.

Aún así pueden surgir varios inconvenientes referentes a las reglas de filtrado especificadas en el *firewall* de modo que permitan el tráfico a través del puerto configurado (1194), a través de la nueva interfaz virtual y desde cada segmento de red de las sucursales.

Consideraciones de *Firewall*

Dependiendo del tipo de *firewall* que se haya establecido en cada uno de los dispositivos de acceso en las sucursales, se deberá considerar varias posibilidades como las que se mencionan a continuación:

- Como se trata de interfaces bastante seguras, es posible especificar reglas que acepten el tráfico de entrada hacia la interfaz virtual, en este caso tun0, asi como todo el tráfico UDP de entrada cuyo puerto destino sea el 1194 perteneciente al servicio OpenVPN.

```
$IPTABLES -A INPUT -i tun0 -j ACCEPT
$IPTABLES -A INPUT -p udp --dport 1194 -j ACCEPT
```

- Asimismo se deben establecer reglas que permitan el tráfico de reenvío a través de los *routers* Linux y su *firewall-box*. Esto significa permitir el tráfico hacia y desde las sucursales que salgan e ingresen a través de la interfaz virtual.

```
$IPTABLES -A FORWARD -i $INTERNAL_INT -o tun+ -j ACCEPT
$IPTABLES -A FORWARD -i tun+ -o $INTERNAL_INT -j ACCEPT
```

- Dependiendo de las reglas de filtrado en cada sucursal será necesario permitir el tráfico de salida a través de la interfaz virtual y de regreso hacia los clientes con el puerto de origen del servicio OpenVPN.

```
$IPTABLES -A OUTPUT -o tun+ -j ACCEPT
$IPTABLES -A OUTPUT -o tap+ -j ACCEPT
$IPTABLES -A OUTPUT -p udp --sport 1194 -j ACCEPT
```

Si existen aún inconvenientes en cuanto a la conexión se deberá depurar los errores que se presenten a través de los registros del kernel en el archivo `/var/log/messages` o la herramienta *tcpdump*, *iptraf* o *nmap* para el monitoreo de paquetes en redes.

Al tomar en cuenta todas las consideraciones que se han especificado en este documento, debería ser posible la conexión entre los segmentos de red de las sucursales y de manera segura.

Configuración del Servicio de Correo Electrónico de manera segura utilizando *Sendmail* en Linux

El proceso de instalación y configuración del servicio de correo electrónico de manera segura, utilizando para ello el Agente de Transporte de Correo *sendmail*, involucra las siguientes etapas:

- Permitir a los usuarios internos enviar correo electrónico a usuarios de Internet
- No permitir que el servidor sea utilizado como *mail-relay*, evitando que el servidor de correo sea utilizado para enviar correo a otros usuarios de Internet. Esto es generalmente hecho por *spammers* para usurpar identidad.
- Bloquear mensajes que sean demasiado largos
- Bloquear *spam* entrante.
- Bloquear mensajes con virus.
- Bloquear mensajes con archivos adjuntos peligrosos.

Sendmail es una aplicación muy tradicional a nivel de correo electrónico y aún cuando se le acusa de tener un amplio historial de *bugs* de seguridad, sigue siendo utilizada a pesar de su antigüedad, muy seguramente por la evolución efectiva de que le han dotado y muy especialmente por la facilidad de su configuración.

Instalación
Los paquetes que deben estar instalados son:

- sendmail
- sendmail-cf

Si el servidor está conectado a Internet el proceso de instalación únicamente requerirá de la ejecución de los comandos:

```
#yum install sendmail
#yum install sendmail-cf
```

Configuración

El proceso de configuración hará referencia a un servidor de correo ubicado en el borde de la red interna conectado directamente a Internet.

Inicialmente es necesario configurar al servidor como un *mail gateway,* lo que quiere decir que todas las peticiones de envío de correo electrónico que no puedan ser realizadas por el *Server*, serán reenviadas a través de él a otros Agentes que si puedan alcanzar el destino.

Sendmail, por defecto está configurado para no realizar *relay* (reenvío) de correo hacia ningún dominio, de manera que para que Sendmail pueda enviar correos desde y hacia un dominio mayor, este dominio debe agregarse en el archivo `/etc/mail/access`, con la etiqueta `RELAY` junto a él.

La secuencia de comandos específica sería la siguiente:

```
# cd /etc/mail
# vi access
Se agrega: machangarasoft.com RELAY
# make
```

Otro paso importante es des-comentar la línea que contiene la opción `DaemonPortOptions` en el archivo `/etc/mail/sendmail.cf`. Al hacerlo se evita que el servidor escuche peticiones únicamente en la interfaz de *loopback*. Resulta mucho más sencillo editar el archivo sendmail.mc y generar el sendmail.cf mediante el comando **m4**. el proceso sería el siguiente:

```
#vi sendmail.mc
La línea quedaría:
dnl DAEMON_OPTIONS('Port=smtp,Addr=127.0.0.1,Name=MTA')
#m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

Y se procede a reiniciar el servicio de sendmail:

```
# service sendmail restart
```

Ahora resta el proceso de configuración propiamente del servicio de correo electrónico mediante la edición del archivo ya mencionado `/etc/mail/sendmail.mc`.

Si se trata del propio dominio, el servidor debe saber que debe entregar correo para nuestro dominio de forma local, en lugar de tratar de pasarlo a otro servidor. Esto se logra agregando nuestro dominio a la clase Cw en el archivo `/etc/mail/local-host-names`, quedando así:

```
Cwmachangarasoft.com
```

Se debe también configurar el nombre de dominio para el correo saliente como nuestro dominio. Esta opción permitirá que cuando alguien que haya recibido un correo con nuestro dominio y haga un *Reply,* se use para ello la dirección de correo electrónico

correcta. Para ello se utiliza la opción `MASQUERADE_DOMAIN` en el archivo `sendmail.mc`.
Los registros MX deben estar correctamente configurados y apuntando a nuestros servidores de correo.

El **tamaño** máximo de los mensajes de correo puede limitarse a través de una opción existente en el archivo **sendmail.mc.**

```
define('confMAX_MESSAGE_SIZE','5000')
```

No hay que olvidar el commando para generar el archive sendmail.cf a partir del sendmail.mc.

El bloqueo de spam se puede hacer a nivel de sendmail directamente al editar el archivo **/etc/mail/access**, utilizando una de las siguientes etiquetas para un dominio específico:

- REJECT obliga a que un mensaje de correo se devuelto a su origen con el mensaje de que no se puede entregar.
- OK Permite aceptar el mensaje aún cuando otras reglas lo rechacen.
- DISCARD Muy útil para protección anti spam. Descarta el mensaje pero sin nada de regreso a quien lo envió, el que pensará que el mensaje fue entregado.
- ### Mensaje de error Permite definir mensajes de error propios, especificando el número y el mensaje a ser usado.

Para la protección más directa contra el *spam* es posible utilizar una herramienta llamada *Spamassassin* o una más completa como *Mailscanner.*
La protección antivirus puede ejecutarse mediante la instalación de una herramienta llamada Clamav.

Configuración de Servidor FTP en Linux

Introducción
FTP *(File Transfer Protocol)* o Protocolo de Transferencia de Archivos es uno de los protocolos estándar más utilizados en Internet, siendo quizá el más adecuado para la transferencia de grandes bloques de datos a través de redes de datos que soportan TCP/IP. Esta característica se ser estándar permite que este disponible y funcione independiente de la plataforma operativa, permitiendo que, con un cliente FTP, prácticamente cualquier sistema operativo pueda tener acceso a este servicio.
El servicio utiliza los puertos 20 y 21, exclusivamente sobre TCP. El puerto 21 se utiliza para el envío de órdenes del cliente hacia el servidor, mientras que el puerto 20 es utilizado por el servidor para crear el canal de datos.

El servicio en Linux
En el sistema operativo Linux, Vsftpd *(Very Secure FTP Daemon)* es un equipamiento lógico utilizado para implementar servidores de archivos a través del protocolo FTP. La ventaja de su utilización es referente a sus valores predeterminados, siendo estos muy seguros y sencillos de configurar, en comparación con otras alternativas como *ProFTPD* y *Wu-ftpd*.
FTP está diseñado para una transferencia de datos muy rápida pero no para hacerla segura, dado que toda la comunicación se realiza en texto plano, de modo que los nombres de usuario y contraseñas podrían ser fácilmente interceptados por cualquier atacante.

Instalación
El único paquete necesario para poner en marcha el servicio FTP es el **vsftpd**. Para instalarlo en una distribución como CentOS 4.3 y con los repositorios adecuadamente configurados, basta con ejecutar en la línea de comandos lo siguiente:

```
# yum install vsftpd
```

Configuración
Para el proceso de configuración del servicio se emplean básicamente 2 archivos:
- `/etc/vsftpd/vsftpd.conf`
- `/etc/vsftpd/vsftpd.user_list`

Dentro del fichero `/etc/vsftpd/vsftpd.conf` se modificarán los siguientes parámetros:

```
anonymous_enable=YES
```
La activación de este parámetro permite el acceso anónimo al servidor.

```
local_enable=YES
```
Establece si se van a permitir los accesos autenticados de los usuarios locales del sistema.

```
write_enable=YES
```
Establece si se permite el mandato **write** (escritura) en el servidor.

```
ftpd_banner=Bienvenido al servidor FTP de MachangaraSoft
```
Este parámetro sirve para establecer el mensaje de bienvenida que será mostrado cada vez que un usuario acceda al servidor. Puede establecerse cualquier frase breve que se considere conveniente.

De manera predeterminada, los usuarios del sistema tienen acceso a otros directorios además de su directorio personal. Si se requiere que solamente puedan utilizar los directorios personales, se debe setear el parámetro **chroot_local_user** que habilitará la función de **chroot()** y los parámetros chroot_list_enable y chroot_list_file para establecer el fichero con la lista de usuarios que quedarán excluidos de la función **chroot()**.
A continuación las líneas de configuración correspondientes:

```
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
```

Es importante que se cree el archivo `vsftpd.chroot_list` pues de otro modo el servicio vsftpd no arrancará.

Se puede reazlizar un control del ancho de banda utilizado por los usuarios del servicio a través de los siguientes parámetros:

```
anon_max_rate=5120
```
Permite limitar la tasa de transferencia de los usuarios anónimos. Su valor se expresa en bytes por segundo.

```
local_max_rate=5120
```
Limita la tasa de transferencia máxima para los usuarios locales del sistema.

```
max_clients=10
```

Establece el número máximo de clientes simultáneos que podrán acceder al servidro FTP.

```
max_per_ip=5
```

Determina el número máximo de conexiones que se pueden establecer hacia el servidor desde una misma dirección IP.

Y como con todos los servicios de Linux, para que tenga efecto los cambios de configuración debe simplemente reiniciar el servicio.

```
# service vsftpd restart
```

ANEXO E

Script de Firewall – Prototipo de Firewall para el MachángaraSoft

Script Firewall MachángaraSoft - prototipo

Ruta: cualquiera

```
###############################################################################################################
################
#--------------------------------------------------------------------------------------------
#     Firewall IPTABLES: script firewall -> de fuego
#--------------------------------------------------------------------------------------------
-----------------
# Realizado por Jose Antonio Estrada Jimenez
# jose.antonio.estrada@gmail.com
#--------------------------------------------------------------------------------------------
-----------------
# Forma de utilizar:
# Ubicarse en el directorio donde esta almacenado el script y ejecutar:
#
# ./firewall {start|stop|status} [Interfaz Externa] [Interfaz Interna]
#--------------------------------------------------------------------------------------------
-----------------
# LOS PARAMETROS:
#  start  -> Inicia la asignacion y verificaciÃ³n de las interfaces y carga las reglas de
#  filtrado
#  stop   -> Borra y Resetea Todas las reglas aplicadas dejando todos los valores por
#  defecto
#  status -> Despliega toda la configuraciÃ³n de reglas de trafico e informacion de las
#  interfaces
#
#  -Interfaz Externa  ->   Corresponde al nombre de la interfaz que tiene conexion de
#  Internet y a traves
#  de la cual toda la red interna se conecta a ala red de redes.
#  -Interfaz Interna  ->  Corresponde al nombre de la Interfaz atraves de la que los
#  usuarios de
#  la red interna ingresan al firewall y luego son enrutados al exterior.
#
#  Ejemplo de utilizacion:
#  [root@gateway]# ./fire start eth1 eth0
#
#  El ejemplo describe la configuracion de la estacion como firewall y especifica que la
```

```
#  interfaz WAN
#  es la eth1 mientras que la interfaz LAN es la eth0.
#
#    * Nota: Si no se especifican los nombres de las interfaces interna y externa, el
#      comando tomara los
#      valores por defecto que son: INTERNAL_INT:eth0 y EXTERNAL_INT: eth1
#
################################################################################
#!/bin/sh

# Ubicacion en el SO del comando iptables
IPTABLES="/sbin/iptables"

# Validacion de parametros del comando
case "$1" in  # Parametro stop -> Flush completo de reglas + polÃticas por defecto ACCEPT
   stop)
      echo "Desactivando el Firewall..."
      $IPTABLES -F
      $IPTABLES -F -t mangle
      $IPTABLES -F -t nat
      $IPTABLES -X
      $IPTABLES -X -t mangle
      $IPTABLES -X -t nat
      #Politica Restrictiva por defecto (Descartar todos los paquetes no explicitamente
      #permitidos)
      $IPTABLES -P INPUT ACCEPT
      $IPTABLES -P OUTPUT ACCEPT
      $IPTABLES -P FORWARD ACCEPT
      echo "...Desactivado"
      ;;
   status) # Parametro status -> Lista contenido de tablas filter, nat y mangle
      echo $"Tabla: filter"
      iptables --list
      echo $"Tabla: nat"
      iptables -t nat --list
      echo $"Tabla: mangle"
      iptables -t mangle --list
      ;;
   restart|reload) # Parametro restart|reload -> Permite reiniciar la carga de reglas
      $0 stop
      $0 start
      ;;
   start)  # Parametro start -> inicio de configuracion y activacion del firewall
    echo "Activando el Firewall..."
    echo ""

################################################################################
##-------------------------Configuracion Inicial------------------------------##
################################################################################

#---------------------------------#
#---- Interfaces por Defecto -----#
#---------------------------------#
## Interfaz Externa por Defecto (se usa si EXTERNAL_INT no se especifica como parametro del
##comando)
DEFAULT_EXTERNAL_INT="eth1"
## Interfaz Interna por Defecto (se usa si INTERNAL_INT no se especifica como parametro del
##comando)
DEFAULT_INTERNAL_INT="eth0"

#-----------------------------#
#--- Variables Especiales ----#
#-----------------------------#
# Mascara IP que identifica a todas las direcciones IP
ALL="0.0.0.0/0"
# Especificacion de puertos no privilegiados
HIGHPORTS="1024:65535"
# Especificacion de puertos para el sistema X Window.
XWINPORTS="6000:6063"

#--------------------------------#
#--- Variables de InundaciÃ³n ----#
#--------------------------------#
# Estas variables representan limites de flujo de paquetes
# Limite para deteccion de Inundacion de paquetes TCP-SYN
```

```
TCPSYNLIMIT="5/s"
# Limite de desborde para deteccion de Inundacion de paquetes TCP-SYN
TCPSYNLIMITBURST="10"
# Limite de Logging en las Cadenas de Log
LOGGINGLIMIT="2/s"
# Limite de desborde para las Cadenas de Log
LOGGINGLIMITBURST="10"
# Limite para la Deteccion de Inundacion de Ping
PINGLIMIT="5/s"
# Limite de Desborde para la deteccion de Inundacion de Ping
PINGLIMITBURST="10"

#-------------------------------------------------------------#
#--- Deteccion Automatica de Informacion de las Interfaces ------#
#-------------------------------------------------------------#
### INTERFAZ EXTERNA:
### ------------------
## Obtencion de informacion de interfaz externa via linea de comandos
## Si no se especifica interfaz en el comando se configura $DEFAULT_EXTERNAL_INT como
EXTERNAL_INT
if [ "x$2" != "x" ]; then
   EXTERNAL_INT=$2
else
   EXTERNAL_INT=$DEFAULT_EXTERNAL_INT
fi
echo Interfaz Externa: $EXTERNAL_INT

## Determinar direccion IP de la Interfaz Externa
EXTERNAL_IP="`ifconfig $EXTERNAL_INT | grep inet | cut -d : -f 2 | cut -d \  -f 1`"
  if [ "$EXTERNAL_IP" = '' ]; then
     echo "Aborting: Unable to determine the IP-address of $EXTERNAL_INT !"
     exit 1
  fi
echo IP Externa: $EXTERNAL_IP

## Determinar direccion IP del gateway externo
EXTERNAL_GW=`route -n | grep -A 4 UG | awk '{ print $2}'`
echo Default Gateway: $EXTERNAL_GW

echo " --- "

### INTERFAZ INTERNA:
### ------------------
## Obtencion de informacion de interfaz interna via linea de comandos
## Si no se especifica interfaz en el comando se configura $DEFAULT_INTERNAL_INT como
INTERNAL_INT

if [ "x$3" != "x" ]; then
   INTERNAL_INT=$3
else
   INTERNAL_INT=$DEFAULT_INTERNAL_INT
fi
echo Interfaz Interna: $INTERNAL_INT

## Determinar direccion IP de la Interfaz Interna
INTERNAL_IP="`ifconfig $INTERNAL_INT | grep inet | cut -d : -f 2 | cut -d \  -f 1`"
  if [ "$INTERNAL_IP" = '' ]; then
     echo "Aborting: Unable to determine the IP-address of $INTERNAL_INT !"
     exit 1
  fi
echo IP Interna: $INTERNAL_IP

## Determinar la mascara de red Interna
INTERNAL_MASK="`ifconfig $INTERNAL_INT | grep Mask | cut -d : -f 4`"
echo Mascara de Red Interna: $INTERNAL_MASK

## Determinar la direccion IP de la red interna
INTERNAL_LAN=$INTERNAL_IP'/'$INTERNAL_MASK
echo LAN Interna: $INTERNAL_LAN

echo ""

#---------------------------------#
#----Carga de Modulos Iptables-----#
#---------------------------------#
```

```
echo "Cargando modulos para IPTABLES"

dmesg -n 1 #Evita que se despliegue un mensaje de CopyRight al cargar los modulos
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp
dmesg -n 6
echo " ..... "

#---------------------------------------------#
#----Limpiar y Resetear todas las cadenas-----#
#---------------------------------------------#

#Borrar todo para partir de CERO
$IPTABLES -F
$IPTABLES -F -t mangle
$IPTABLES -F -t nat
$IPTABLES -X
$IPTABLES -X -t mangle
$IPTABLES -X -t nat
#Configurar las Politicas por defecto en DROP
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

#---------------------------------------------#
#----Configurar opciones de red del kernel-----#
#---------------------------------------------#

echo "Configurando opciones de kernel"

# Habilitar enrutamiento en el kernel
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward

# Deshabilitar IP-Spoofing en todas las interfaces
echo 2 > /proc/sys/net/ipv4/conf/all/rp_filter

# No Responder a Pings Broadcast (Proteccion Smurf-Amplifier)
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

#Bloqueo de source routing
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

# Eliminar marcas de tiempo (timestamps)
echo 0 > /proc/sys/net/ipv4/tcp_timestamps

# Habilitar SYN Cookies
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# Eliminar redirects
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects

# Habilitar proteccion contra mensajes bad error
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Registrar martians (paquetes con direcciones IP imposibles)
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians

#Set out local port range
echo "32768 61000" > /proc/sys/net/ipv4/ip_local_port_range

# Reducir la posibilidad de ataques DoS reduciendo los timeouts
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 2400 > /proc/sys/net/ipv4/tcp_keepalive_time
echo 0 > /proc/sys/net/ipv4/tcp_window_scaling
echo 0 > /proc/sys/net/ipv4/tcp_sack

echo " ..... "

echo "Creando Cadenas de Usuario"

#---------------------------------#
#--- Crear Cadenas de Logging ----#
#---------------------------------#

# Son cadenas de registro de eventos (logging). Poseen un limite de entradas por segundo
# para prevenir
# la inundacion de logs

# Paquetes invalidos (no ESTABLISHED,RELATED o NEW)
```

```
        $IPTABLES -N LINVALID
 $IPTABLES -A LINVALID -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST –j LOG --log-prefix "Detect FW: INVALID:1 DROP"
 $IPTABLES -A LINVALID -j DROP

# Paquetes TCP con banderas erroneas
        $IPTABLES -N LBADFLAG
 $IPTABLES -A LBADFLAG -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: BADFLAG:1 DROP"
 $IPTABLES -A LBADFLAG -j DROP

# Logging de intentos de conexion en puertos especiales (Escaneo de puertos troyanos, servicios especiales, etc.)
        $IPTABLES -N LSPECIALPORT
 $IPTABLES -A LSPECIALPORT -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: SPECIALPORT:1 DROP"
 $IPTABLES -A LSPECIALPORT -j DROP

#Logging de posibles Inundaciones SYN
        $IPTABLES -N LSYNFLOOD
 $IPTABLES -A LSYNFLOOD -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: SYNFLOOD:1 DROP"
 $IPTABLES -A LSYNFLOOD -j DROP

#Logging de Posibles Inundaciones Ping
        $IPTABLES -N LPINGFLOOD
 $IPTABLES -A LPINGFLOOD -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: PINGFLOOD:1 DROP"
 $IPTABLES -A LPINGFLOOD -j DROP


# Paquetes Restantes Descartados
        $IPTABLES -N LDROP
 $IPTABLES -A LDROP -p tcp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: TCP:1 DROP"
 #$IPTABLES -A LDROP -p udp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: UDP:2 DROP"
 $IPTABLES -A LDROP -p icmp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: ICMP:3 DROP"
 $IPTABLES -A LDROP -f -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: FRAGMENT:4 DROP"
 $IPTABLES -A LDROP -j DROP

# Paquetes Restantes Rechazados
        $IPTABLES -N LREJECT
 $IPTABLES -A LREJECT -p tcp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: TCP:1 REJECT"
 #$IPTABLES -A LREJECT -p udp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: UDP:2 REJECT"
 $IPTABLES -A LREJECT -p icmp -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: ICMP:3 REJECT"
 $IPTABLES -A LREJECT -f -m limit --limit $LOGGINGLIMIT --limit-burst $LOGGINGLIMITBURST -j LOG --log-prefix "Detect FW: FRAGMENT:4 REJECT"
 $IPTABLES -A LREJECT -p tcp -j REJECT --reject-with tcp-reset
        $IPTABLES -A LREJECT -p udp -j REJECT --reject-with icmp-port-unreachable
        $IPTABLES -A LREJECT -j REJECT
#-------------------------------#
#----Creando Cadenas ACCEPT-----#
#-------------------------------#

#TCPACCEPT - Revisa posible inundacion de paquetes SYN antes de permitir el paso de paquetes TCP
```

```
       $IPTABLES -N TCPACCEPT
        $IPTABLES -A TCPACCEPT -p tcp --syn -m limit --limit $TCPSYNLIMIT --limit-burst
$TCPSYNLIMITBURST -j ACCEPT
        $IPTABLES -A TCPACCEPT -p tcp --syn -j LSYNFLOOD
        $IPTABLES -A TCPACCEPT -p tcp ! --syn -j ACCEPT
#-----------------------------------------------#
#----Creando Cadenas de Usuario Especiales-----#
#-----------------------------------------------#


#CHECKBADFLAG - Descarta cualquier paquete TCP entrante o saliente con combinaciones
imposibles de banderas (Utilizadas por escaneres de #puertos)
#------------
        $IPTABLES -N CHECKBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL FIN,URG,PSH -j LBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j LBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL ALL -j LBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags ALL NONE -j LBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags SYN,RST SYN,RST -j LBADFLAG
        $IPTABLES -A CHECKBADFLAG -p tcp --tcp-flags SYN,FIN SYN,FIN -j LBADFLAG



#FILTRADO DE PUERTOS ESPECIALES - Puertos propensos a ataques backdoor
#------------------------------


        #Entrante/Saliente

                #SMB-Traffic
                $IPTABLES -N SMB

                $IPTABLES -A SMB -p tcp --dport 137 -j DROP
                $IPTABLES -A SMB -p tcp --dport 138 -j DROP
                $IPTABLES -A SMB -p tcp --dport 139 -j DROP
                $IPTABLES -A SMB -p tcp --dport 445 -j DROP
                $IPTABLES -A SMB -p udp --dport 137 -j DROP
                $IPTABLES -A SMB -p udp --dport 138 -j DROP
                $IPTABLES -A SMB -p udp --dport 139 -j DROP
                $IPTABLES -A SMB -p udp --dport 445 -j DROP

                $IPTABLES -A SMB -p tcp --sport 137 -j DROP
                $IPTABLES -A SMB -p tcp --sport 138 -j DROP
                $IPTABLES -A SMB -p tcp --sport 139 -j DROP
                $IPTABLES -A SMB -p tcp --sport 445 -j DROP
                $IPTABLES -A SMB -p udp --sport 137 -j DROP
                $IPTABLES -A SMB -p udp --sport 138 -j DROP
                $IPTABLES -A SMB -p udp --sport 139 -j DROP
                $IPTABLES -A SMB -p udp --sport 445 -j DROP

       #Puertos Especiales de Entrada

                $IPTABLES -N SPECIALPORTS

                #Escaneo Deepthroat
                $IPTABLES -A SPECIALPORTS -p  tcp --dport 6670 -j LSPECIALPORT

                #Escaneo Subseven
                $IPTABLES -A SPECIALPORTS -p tcp --dport 1243 -j LSPECIALPORT
                $IPTABLES -A SPECIALPORTS -p udp --dport 1243 -j LSPECIALPORT
                $IPTABLES -A SPECIALPORTS -p tcp --dport 27374 -j LSPECIALPORT
                $IPTABLES -A SPECIALPORTS -p udp --dport 27374 -j LSPECIALPORT
                $IPTABLES -A SPECIALPORTS -p tcp --dport 6711:6713 -j LSPECIALPORT

                #Escaneo Netbus
                $IPTABLES -A SPECIALPORTS -p tcp --dport 12345:12346 -j LSPECIALPORT
                $IPTABLES -A SPECIALPORTS -p tcp --dport 20034 -j LSPECIALPORT

                #Escaneo Back Orifice
                $IPTABLES -A SPECIALPORTS -p udp --dport 31337:31338 -j LSPECIALPORT

                #X-Windows
                $IPTABLES -A SPECIALPORTS -p tcp --dport $XWINPORTS  -j LSPECIALPORT

                #Hack'a'Tack 2000
```

```
		$IPTABLES -A SPECIALPORTS -p udp --dport 28431 -j LSPECIALPORT

#FILTRADO ICMP/TRACEROUTE
#------------------------

	 #ICMP/Traceroute Entrante
	#------------------------
		$IPTABLES -N ICMPINBOUND

		#Proteccion contra Inundacion Ping - Se acepta un limite de peticiones/segundo y el resto se descartan
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type echo-request -m limit --limit $PINGLIMIT --limit-burst $PINGLIMITBURST -j ACCEPT
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type echo-request -j LPINGFLOOD

		#Bloquear ICMP-Redirects - Aun cuando ya deberian filtrarse gracias a las opciones de kernel
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type redirect -j LDROP

		#Bloquear paquetes ICMP-Timestamp - deberian ya filtrarse gracias a las opciones de kernel
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type timestamp-request -j LDROP
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type timestamp-reply -j LDROP

		#Bloquear mensajes ICMP-address-mask (puede evitar problemas de OS-fingerprinting)
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type address-mask-request -j LDROP
		$IPTABLES -A ICMPINBOUND -p icmp --icmp-type address-mask-reply -j LDROP

		#Permitir todos los paquetes ICMP restantes que ingresan
		$IPTABLES -A ICMPINBOUND -p icmp -j ACCEPT

	#ICMP/Traceroute de Salida
	#-------------------------
		$IPTABLES -N ICMPOUTBOUND

		# Bloquear paquetes ICMP-Redirect (ya deberian descartarse mediante las opciones de kernel)
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type redirect -j LDROP

		# Bloquear paquetes ICMP-TTL-Expired
		#Traceroute MS (MS usa ICMP en lugar de UDP para hacer tracert)
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type ttl-zero-during-transit -j LDROP
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type ttl-zero-during-reassembly -j LDROP

		#Bloquear paquetes ICMP-Parameter-Problem
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type parameter-problem -j LDROP

		#Bloquear paquetes ICMP-Timestamp (ya deberian descartarse mediante las opciones de kernel)
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type timestamp-request -j LDROP
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type timestamp-reply -j LDROP

		#Bloquear paquetes ICMP-address-mask (permite prevenir OS-fingerprinting)
		$IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type address-mask-request -j LDROP
```

```
                $IPTABLES -A ICMPOUTBOUND -p icmp --icmp-type address-mask-reply -j LDROP

                #Aceptar el resto de paquetes ICMP salientes
                $IPTABLES -A ICMPOUTBOUND -p icmp -j ACCEPT


#----Fin de las Cadenas de Usuario-----#
#*************************************#
echo " --- "

################################################
#---- INICIO DEL SET DE REGLAS DE FIREWALL ----#
################################################
echo "Implementando Reglas de Firewall..."


##################
## Cadena INPUT ## (Se refiere a todos los paquetes que van dirigidos al firewall-box)
##################
# Aceptar paquetes pertenecientes a conexiones establecidas o relacionadas
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  #----------------
  # Filtrado GENERAL
  #-----------------
  # Descartar paquetes invalidos (que no son del tipo ESTABLISHED, RELATED o NEW)
  $IPTABLES -A INPUT -m state --state INVALID -j LINVALID

  # Descartar paquetes con banderas erroneas
  $IPTABLES -A INPUT -p tcp -j CHECKBADFLAG

  #-----------------------------------------
  # Paquetes que se originan en el firewall-box
  #-----------------------------------------
    #Interfaz de loopback
    $IPTABLES -A INPUT -i lo -j ACCEPT

    #Descartar conexiones hacia la interfaz de loopback desde Internet (Deberia ya
    #bloquearse con las opciones de kernel)
    $IPTABLES -A INPUT -d 127.0.0.0/8 -j LREJECT

  #-----------------------------------------
  #Paquetes que se ORIGINAN en la RED INTERNA
  #-----------------------------------------

    ##Filtrado ICMP y tracerouting

      #Filtrar ICMP
      $IPTABLES -A INPUT -i $INTERNAL_INT -p icmp -j ICMPINBOUND
      #Bloquear traceroute UDP
      $IPTABLES -A INPUT -p udp --dport 33434:33523 -j LDROP

    ##Permitir el ingreso de peticiones DNS
   $IPTABLES -A INPUT -i $INTERNAL_INT -p udp -s $INTERNAL_LAN -d $ALL --dport 53 -j ACCEPT
   $IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN -d $ALL --dport 53 -j ACCEPT
    ##Permitir el ingreso de peticiones al servicio de PROXY SQUID
    $IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS --dport
8080 -j TCPACCEPT
    ##SSH desde INTRAnet
    $IPTABLES -A INPUT -i $INTERNAL_INT -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS --dport
22 -j TCPACCEPT
    ##OpenVPN desde la INTRAnet e INTERNET - Imprescindible
    $IPTABLES -A INPUT -i tun+ -j ACCEPT
    $IPTABLES -A INPUT -i tap+ -j ACCEPT
    $IPTABLES -A INPUT -p udp --dport 1194 -j ACCEPT

    ##Descartar todo lo que provenga de fuera y que pretenda venir de la red interna
    (Spoofing--> ya deberia descartarse mediante las opciones ##de kernel)
    $IPTABLES -A INPUT -s $INTERNAL_LAN -j LREJECT

  #---------------------------------------
  #Paquetes provenientes de la RED EXTERNA
  #---------------------------------------

  ##Filtrado ICMP y tracerouting
```

```
    #Filtrar ICMP
    $IPTABLES -A INPUT -i $EXTERNAL_INT -p icmp -j ICMPINBOUND
    #Bloquear traceroute UDP
    $IPTABLES -A INPUT -p udp --dport 33434:33523 -j LDROP

  ##Drops/Rejects

    #Descartar todo el trafico SMB
    $IPTABLES -A INPUT -i $EXTERNAL_INT -j SMB
    #Rechazar Ident de forma silenciosa (No se descarta ident, se pueden generar retardos
cuando se establecen conexiones salientes)
    $IPTABLES -A INPUT -i $EXTERNAL_INT -p tcp --dport 113 -j REJECT --reject-with tcp-
reset
  ##Permitir Acceso a Webmin
  $IPTABLES -A INPUT -i $EXTERNAL_INT -p tcp --dport 10000 -j ACCEPT
  ##SSH desde INTERnet
  $IPTABLES -A INPUT -i $EXTERNAL_INT -p tcp --dport 22 -j TCPACCEPT
  ##Registro de intentos de conexion o escaneo de puertos especiales
  $IPTABLES -A INPUT -i $EXTERNAL_INT -j SPECIALPORTS
  ##Descartar el resto de paquetes entrantes
  $IPTABLES -A INPUT -j LDROP



###################
## Cadena OUTPUT ## (todo lo que sale directamente del Firewall-Box)
###################
#Aceptar trafico establecido en la interfaz de salida
$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

  #----------------------------------------
  #Paquetes que van dirigidos al firewall-box
  #----------------------------------------
    #Interfaz de loopback
    $IPTABLES -A OUTPUT -o lo -j ACCEPT

  #-------------------------------------
  #Paquetes que van hacia la red INTERNA
  #-------------------------------------
    #Permitir trafico ilimitado hacia la red interna
    $IPTABLES -A OUTPUT -o $INTERNAL_INT -d $INTERNAL_LAN -j ACCEPT

  #-------------------------------------
  #Paquetes que van hacia la red EXTERNA
  #-------------------------------------
     #OpenVPN
     $IPTABLES -A OUTPUT -o tun+ -j ACCEPT
     $IPTABLES -A OUTPUT -o tap+ -j ACCEPT
     $IPTABLES -A OUTPUT -p udp --sport 1194 -j ACCEPT
     #ICMP y Traceroute
     $IPTABLES -A OUTPUT -o $EXTERNAL_INT -p icmp -j ICMPOUTBOUND
     #SMB
     $IPTABLES -A OUTPUT -o $EXTERNAL_INT -j SMB
     #Ident
     $IPTABLES -A OUTPUT -o $EXTERNAL_INT -p tcp --sport 113 -j REJECT --reject-with tcp-
reset
     #Aceptar todo el trafico tcp/udp saliente en puertos no privilegiados
     $IPTABLES -A OUTPUT -o $EXTERNAL_INT -s $EXTERNAL_IP -p tcp --sport $HIGHPORTS -j
ACCEPT
     $IPTABLES -A OUTPUT -o $EXTERNAL_INT -s $EXTERNAL_IP -p udp --sport $HIGHPORTS -j
ACCEPT

#Descartar el resto de paquetes salientes
$IPTABLES -A OUTPUT -j LDROP

######################
## Cadena FORWARD  ## (Todos los paquetes que atraviesen el firewall)
######################
#Hacer FORWARD de TODOS los paquetes provenientes de la RED INTERNA
#$IPTABLES -A FORWARD -i $INTERNAL_INT -s $INTERNAL_LAN -j ACCEPT
#Aceptar paquetes pertenecientes a conexiones establecidas o relacionadas
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
  #----------------
  # Filtrado GENERAL
  #-----------------
  #Descartar paquetes invalidos
  $IPTABLES -A FORWARD -m state --state INVALID -j LINVALID

  # Descartar paquetes con banderas erroneas
  $IPTABLES -A FORWARD -p tcp -j CHECKBADFLAG

  #-------------------------------------
  #Paquetes que van hacia la Red Externa
  #-------------------------------------

     #OpenVPN - Indispensables para trafico OpenVPN
     $IPTABLES -A FORWARD -i $INTERNAL_INT -o tun+ -j ACCEPT
     $IPTABLES -A FORWARD -i tun+ -o $INTERNAL_INT -j ACCEPT

  #Reenvio SIP ASTERISK
  $IPTABLES -A FORWARD -i $INTERNAL_INT -s $INTERNAL_LAN -m udp -p udp --dport 5060 -j
ACCEPT
  $IPTABLES -A FORWARD -i $INTERNAL_INT -s $INTERNAL_LAN -m udp -p udp --dport 10000:10100
-j ACCEPT
  #$IPTABLES -A FORWARD -i $EXTERNAL_INT -o $INTERNAL_INT -s $INTERNAL_LAN -p udp --dport
5060 -j ACCEPT
  #$IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p udp --dport
10000:10100 -j ACCEPT
  #$IPTABLES -A FORWARD -i $EXTERNAL_INT -o $INTERNAL_INT -s $INTERNAL_LAN -p udp --dport
10000:10100 -j ACCEPT


  #Reenvio ICMP
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p icmp -j ACCEPT

  #Trafico SMB
  $IPTABLES -A FORWARD -o $EXTERNAL_INT -j SMB

  #Reenvio DNS
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p udp --sport
$HIGHPORTS -d $ALL --dport 53 -j ACCEPT
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p tcp --sport
$HIGHPORTS -d $ALL --dport 53 -j ACCEPT

  #Reenvio SSH
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p tcp --sport
$HIGHPORTS -d $ALL --dport 22 -j ACCEPT

  #Reenvio FTP
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p tcp --sport
$HIGHPORTS -d $ALL --dport 20 -j ACCEPT
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p tcp --sport
$HIGHPORTS -d $ALL --dport 21 -j ACCEPT
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p udp --sport
$HIGHPORTS -d $ALL --dport 20 -j ACCEPT
  $IPTABLES -A FORWARD -i $INTERNAL_INT -o $EXTERNAL_INT -s $INTERNAL_LAN -p udp --sport
$HIGHPORTS -d $ALL --dport 21 -j ACCEPT

  #Reenvio HTTPS
  $IPTABLES -A FORWARD -m state --state NEW -p tcp --dport 443 -j ACCEPT

  #Reenvio MESSENGER
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL --dport 1863 -j
ACCEPT

  #Reenvio LIMEWIRE
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL --dport 6346 -j
ACCEPT

  #Reenvio POP/IMAP-SMTP
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL --dport 110 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL --dport 109 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS -d $ALL --dport 25 -j
ACCEPT
```

```
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 25 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 143 -j
ACCEPT
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 143 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 220 -j
ACCEPT
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 220 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 993 -j
ACCEPT
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 993 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 995 -j
ACCEPT
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 995 -j
ACCEPT
  $IPTABLES -A FORWARD -p tcp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 106 -j
ACCEPT
  $IPTABLES -A FORWARD -p udp -s $INTERNAL_LAN --sport $HIGHPORTS  -d $ALL --dport 106 -j
ACCEPT

  # Reenvio hacia el servidor Web
  $IPTABLES -A FORWARD -o $EXTERNAL_INT -s $HTTP_IP -p tcp --sport 80 -j ACCEPT
  $IPTABLES -A FORWARD -i $EXTERNAL_INT -p tcp -d $HTTP_IP --dport 80 -j ACCEPT

  # Reenvio servidor de correo
   $IPTABLES -A FORWARD -o $EXTERNAL_INT -s $SMTP_IP -p tcp --sport 20 -j ACCEPT
   $IPTABLES -A FORWARD -o $EXTERNAL_INT -s $SMTP_IP -p tcp --sport 21 -j ACCEPT
   $IPTABLES -A FORWARD -i $EXTERNAL_INT -p tcp -d $SMTP_IP --dport 20 -j ACCEPT
   $IPTABLES -A FORWARD -i $EXTERNAL_INT -p tcp -d $SMTP_IP --dport 21 -j ACCEPT

  #-------------------------------------
  #Paquetes que van hacia la Red Interna
  #-------------------------------------
  #Trafico SMB
  $IPTABLES -A FORWARD -i $EXTERNAL_INT -j SMB

  #Descartar paquetes de FORWARDING restantes
  $IPTABLES -A FORWARD -j LDROP

############################
## Cadena de PREROUTING ##
############################
#----------------------
#Redireccion de Puertos
#----------------------
  #Redireccion de trafico HTTP hacia el servicio de Proxy
  $IPTABLES -t nat -A PREROUTING -p tcp -s $INTERNAL_LAN -d $ALL --dport 80 -j REDIRECT --
to-port 8080

  # Redireccion de peticion ftp hacia el servidor ftp
  $IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP --dport 20 -j DNAT
--to $FTP_IP
  $IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP --dport 21 -j DNAT
--to $FTP_IP

  # Redirección de peticiones web al servidor web (HTTP_IP)
  $IPTABLES -A PREROUTING -t nat -i $EXTERNAL_INT -p tcp -d $EXTERNAL_IP --dport 80 -j DNAT
--to $HTTP_IP

#############################
## Cadena de POSTROUTING  ##
#############################

  #Enmascaramiento de la Red Interna Hacia la Red Externa
  $IPTABLES -A POSTROUTING -t nat -o $EXTERNAL_INT -j MASQUERADE

#------ Fin ------#

echo "...hecho"
echo ""
```

```
echo "--> Firewall IPTABLES ACTIVADO <--"

##-------------------------------Fin de Especificacion de Reglas-------------------------
-------##

   ;;
   *)
      echo "Forma de Uso: fire (start|stop|restart|status) EXTERNAL_INT INTERNAL_INT"
      exit 1
esac

exit 0
```

ANEXO F

Pruebas de Escaneo (Nessus-Nmap)

Resultado de Prueba de Escaneo con *Nmap* en la interfaz externa del *firewall* desde Internet

```
Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2007-05-19 16:09 ECT
sendto in send_ip_packet: sendto(8, packet, 32, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(8, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
sendto in send_ip_packet: sendto(3, packet, 40, 0, 190.10.225.128, 16) => Operation not permitted
Interesting ports on 128.190-10-225.uio.satnet.net (190.10.225.128):
(The 65529 ports scanned but not shown below are in state: filtered)
PORT     STATE  SERVICE   VERSION
20/tcp   closed ftp-data
21/tcp   open   ftp       vsFTPd
22/tcp   open   ssh       OpenSSH 3.9p1 (protocol 1.99)
25/tcp   open   smtp      Sendmail 8.12.11.20060308/8.12.11
113/tcp  closed auth
8081/tcp open   http      Apache httpd
MAC Address: 00:11:95:FF:8E:A1 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-
submit.cgi).
TCP/IP fingerprint:
SInfo(V=3.70%P=i686-redhat-linux-gnu%D=5/19%Time=464F68CF%O=21%C=20)
TSeq(Class=RI%gcd=1%SI=9180C%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=9184E%IPID=Z%TS=U)
TSeq(Class=RI%gcd=1%SI=917F4%IPID=Z%TS=U)
T1(Resp=Y%DF=Y%W=16D0%ACK=S++%Flags=AS%Ops=M)
T2(Resp=N)
```

```
T3(Resp=N)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=N)
PU(Resp=N)


Nmap run completed -- 1 IP address (1 host up) scanned in 313.763 seconds
```

Resultado de Prueba de Escaneo con *Nessus* en la interfaz externa del *firewall* desde Internet

Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List	
Host(s)	**Possible Issue**
190.10.225.128	Security note(s) found

[return to top]

Analysis of Host		
Address of Host	**Port/Service**	**Issue regarding Port**
190.10.225.128	ssh (22/tcp)	Security notes found
190.10.225.128	ftp (21/tcp)	Security notes found
190.10.225.128	smtp (25/tcp)	Security notes found
190.10.225.128	sunproxyadmin (8081/tcp)	Security notes found
190.10.225.128	general/tcp	Security notes found

Security Issues and Fixes: 190.10.225.128		
Type	**Port**	**Issue and Fix**
Informational	ssh (22/tcp)	An ssh server is running on this port Nessus ID : 10330
Informational	ftp (21/tcp)	An FTP server is running on this port. Here is its banner : 220 Bienvenido al servidor FTP de la Corporacion MachangaraSoft Nessus ID : 10330
Informational	smtp (25/tcp)	An SMTP server is running on this port Here is its banner : 220 server.7evenideas.com ESMTP Sendmail 8.12.11.20060308/8.12.11; Sat, 19 May 2007 16:15:12 -0500 Nessus ID : 10330
Informational	sunproxyadmin (8081/tcp)	A web server is running on this port Nessus ID : 10330
Informational	sunproxyadmin (8081/tcp)	Synopsis : A web server is running on the remote host. Description : This plugin attempts to determine the type and the version of the remote web server. Risk factor : None

		Plugin output : The remote web server type is : Apache and the 'ServerTokens' directive is ProductOnly Apache does not permit to hide the server type. Nessus ID : 10107
Informational	sunproxyadmin (8081/tcp)	Synopsis : It is possible to enumerate web directories. Description : This plugin attempts to determine the presence of various common dirs on the remote web server. Risk factor : None Plugin output : The following directories were discovered: /cgi-bin, /error, /icons, /manual, /prueba While this is not, in and of itself, a bug, you should manually inspect these directories to ensure that they are in compliance with company security standards Other references : OWASP:OWASP-CM-006 Nessus ID : 11032
Informational	general/tcp	Information about this scan : Nessus version : 2.2.8 (Nessus 2.2.9 is available - consider upgrading) Plugin feed version : 200705190615 Type of plugin feed : Registered (7 days delay) Scanner IP : 190.10.228.28 Port scanner(s) : synscan nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 1 Report Verbosity : 1 Safe checks : yes Max hosts : 20 Max checks : 4 Scan duration : unknown (ping_host.nasl not launched?) Nessus ID : 19506

BIBLIOGRAFÍA

- FONG, Paul; KNIPP, Eric, "Configuring Cisco Voice Over IP". Syngress Media Publishing Inc. USA 2002, Segunda Edición.
- SPENCER, Marc, "Inter-Asterisk EXchange (IAX) Version 2", Diguim, Inc, Enero 2005. (http://www.cornfed.com/iax.pdf)
- VAN MEGGELEN, Jim; SMITH, Jared; MADSEN, Leif, "Asterisk, The Future of Telephony". O'Reilly Media Inc. USA, Septiembre 2005, Primera Edición
- NEGUS, Christopher, "Linux Bible 2005 edition". Whyle Publishing. Indianápolis, Indiana. 2005
- NEGUS, Christopher, "Red Hat Fedora and Enterprise Linux 4 Bible", Wiley Publishing, Inc. 2005
- TANENBAUN, Andrew, "Sistemas operativos modernos". 2ª Edición. 2003
- ARROYO, José. "Edición especial Linux máxima seguridad". 2000
- Jalercom S.A. *'Cómo hacer aplicaciones de VoIP Seguras'* México 2007. URL: http://www.jalercom.com
- Hersent O, Gurle D, Petit J. 2000. *'IP Telephony : Packed Based Communications Systems'.* Adison –Wesley
- Ruiz Andrade, Pablo Aníbal. *"Estudio y diseño de una red de datos corporativa basada en la tecnología Frame Relay, que permita el transporte de voz y video".* Octubre 2004
- SPENCER, Mark, *Asterisk Handbook,* Digium Inc., 2003
- ZWICKY, Elizabeth, *Building Internet Firewalls,* O'Reilly, 2da Edición, Junio 2000
- GARFINKEL, Simson, *Practical Unix and Internet Security*, O'Reilly 2da Edición
- HUNT, Craig, *TCP/IP Network Administration,* O'Reilly, 3ra Edición, Abril 2002

- *Firewalling, NAT and Packet Mangling for Linux,* URL http://www.netfilter.org,
- *Wiki guía de referencia para VoIP*, URL http://www.voip-info.org/ ,

-
- *Website de la distribución Linux Tribox con Asterisk y la interfaz gráfica FreePBX embebida*, URL http://www.trixbox.org,
- *Tutoriales para Asterisk PBX*, URL http://www.asteriskguru.com/
- *Repositorio de código fuente de software libre*, http://sourceforge.net/
- *Website de la empresa dueña del Softphone Xlite*, URL http://www.counterpath.com/
- *Sitio de desarrollo de Ekiga Softphone*, URL http://www.ekiga.org/
- *Sitio de desarrollo de Linphone Softphone*, URL http://www.linphone.org/
- *Foro de Asterisk en español*, URL http://www.asterisk-es.org
- Digium, The Asterisk Company. URL: http://www.digium.com
- Jacques O. 2006: *'SIP Project Page'*. URL http://sipp.sourceforge.net
- Comunidad de Usuarios de Linux en Ecuador. URL: http://www.ecualug.org
- Proyecto iLBCfreeware.org. 2003. URL http://www.ilbcfreeware.org.