UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE UN SISTEMA GRÁFICO PARA LA GESTIÓN DE UN FIREWALL DEL TIPO PACKET FILTER

T E S I S

PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A

HUMBERTO KEYMUR LANDEROS

DIRECTOR DE TESIS:
ING. RAFAEL SANDOVAL VÁZQUEZ

MÉXICO DF 2014

Agradecimientos

A mi esposa Daniela y mi familia

Al Ingeniero Rafael Sandoval, por guiarme, apoyarme y brindarme el conocimiento necesario para la realización de esta tesis.

A mis compañeros, amigos y colegas, Carmen, Mauricio, Lilia y Ángel por su apoyo, entusiasmo y compañerismo.

A mi esposa, por estar siempre a mi lado y apoyarme en todo momento.

A mis padres por el apoyo, cariño, consejo y esfuerzo durante toda la vida.

A la UNAM, por ser mí casa desde la secundaria y a la Facultad de Ingeniería, por las oportunidades que me ha brindado.

A la CUAED y al Instituto de Ingeniería por aceptarme y guiarme en la dirección correcta.

Finalmente a la Academia Mexicana de Ciencias, por el apoyo, confianza y oportunidades que me han brindado, sin los cuales este proyecto no sería una realidad.

Contenido

Objetivo

Desarrollar una herramienta gráfica vía Web que permita gestionar la construcción de políticas y reglas de filtrado de paquetes para un *Firewall* de modo transparente o NAT en el sistema operativo OpenBSD con “Packet Filter”.

Introducción

El *Firewall* se considera la primera medida de seguridad que debe tener cualquier infraestructura de red, ya que implementa reglas simples que indican qué se puede y qué no se puede hacer.

Existen una gran cantidad de *Firewalls*. Los comerciales son fáciles de utilizar, efectivos, cuentan con una interfaz administrativa (sin comandos), respaldos de configuración adaptados a múltiples idiomas, funcionan en forma Transparente o NAT.

Los *Firewalls* desarrollados en software libre son tan buenos como los comerciales, se utilizan en instituciones y escuelas que no cuentan con grandes recursos económicos, pero tienen un inconveniente, sólo pueden ser utilizadas por muy pocos. Para implementarlo se deben contar con conocimientos de sistemas operativos tipo Unix, línea de comandos, administración de redes, protocolos y conocimiento del idioma inglés.

La mayoría de los F*irewalls* que se han implementado funcionan de forma parecida a “Packet Filter” de OpenBSD. Es por ello importante aprender a utilizarlos e implementarlos de manera adecuada, eficiente y sobre todo segura.

La propuesta *Desarrollo de un sistema gráfico para la gestión de un Firewall del tipo "Packet Filter"* permitirá realizar las tareas de configuración, administración y mantenimiento de las reglas de filtrado utilizando una interfaz gráfica vía Web.

Capítulo 1

Fundamentos de Seguridad y Redes

En este capítulo se describirán algunos de los conceptos básicos relacionados con la seguridad y las redes, así como los modelos y servicios empleados. Con el fin de tener una idea general de la seguridad informática.

Desde la antigüedad el hombre se ha preocupado por la seguridad, debido a que le confiere certeza sobre sus acciones y las repercusiones que le podrían acarrear sus actos, tanto el hacerlos u omitirlos. Se preocupaba sobre cuestiones tales como, las fechas idóneas para cosechar, los conflictos territoriales con pueblos cercanos, entre otras.

Por otro lado, las redes ayudan a comunicarse de una mejor manera, permitiendo realizar caminos, reducir tiempos de traslado, hablar por medio de cables, transmitir mensajes de manera alámbrica e inalámbrica acercando a las personas de una manera no conocida hasta entonces.

Con el inicio de la computación se lograron reducir los tiempos de respuesta y realizar tareas rutinarias de manera automática, de esta manera el hombre dejo de realizar tareas manuales confiriéndoselas a las computadoras y dejando en ellas grandes responsabilidades.

Cuando se crearon las redes de computadoras permitieron compartir los recursos informáticos en una red pequeña, destinada a universidades e institutos que controlaban estos equipos. Comparada con la de nuestros días en la que conviven millones de equipos en redes, subredes y subredes dentro de las subredes esto es un gran problema.

La sociedad actual se sustenta en el uso de las redes, con ellas podemos saber el tiempo, localizar un lugar en el planeta, mirar al espacio, reducir el tráfico automovilístico, asistir a la escuela, realizar transacciones bancarias, ver televisión, leer un libro, hablar por teléfono, entre otras.

Las preocupaciones del hombre han cambiado, pero siempre querrá tener la seguridad sobre aquello que lo rodea.

1.1 Seguridad: Conceptos Generales

La seguridad es definida como un mecanismo que asegura algún buen funcionamiento, precaviendo que este falle, se frustre o se violente (Real Academia Española, 2014).

Otros autores (López Barrientos & Quezada Reyes, 2006) definen la seguridad como todo tipo de precauciones y protecciones que se llevan a cabo para evitar cualquier acción que comprometa a la información".

En este capítulo se analizan los conceptos fundamentales de la seguridad informática así como los términos y definiciones relacionadas con esta.

1.1.1 Seguridad Informática

Algunos autores definen la seguridad informática como:

- “No se podrá entender la seguridad informática como un concepto cerrado consecuencia de la aplicación mecánica de una serie de métodos, sino como el proceso que se puede ver comprometido en cualquier momento de la forma menos sospechada” (Carvajal, Introducción a la inseguridad de la información, 2008).

- “Podemos definir a la seguridad informática como cualquier medida que impida la ejecución de operaciones no autorizadas sobre un sistema o red informática, cuyos efectos pueden conllevar daños sobre la información, comprometer su confidencialidad, autenticidad o integridad, disminuir el rendimiento de los equipos o bloquear el acceso de usuarios autorizados al sistema” (Gómez Vieites, 2011).

- “Se refiere a la prevención y a la protección, a través de ciertos mecanismos, para evitar que ocurra de manera accidental o intencional la transferencia, modificación, fusión o destrucción no autorizada de la información” (López Barrientos & Quezada Reyes, 2006).

Al hacer, la analogía con la medicina que trata de preservar la salud en los hombres, la seguridad informática trata de preservar la información. En la medicina a una persona enferma se le trata de sanar. ¿Por qué en la informática no sucede lo mismo?

En las definiciones anteriores hace falta un punto importante. Una vez que el sistema o red informático no se encuentra seguro, ¿qué se debe hacer?

Por lo tanto, en este trabajo definiremos a la seguridad informática como el proceso de métodos y reglas preventivas, correctivas y de detección que se deben seguir para salvaguardar los sistemas y redes informáticas.

1.1.2 Elementos Intrínsecos en los sistemas informáticos

Todos los sistemas informáticos están formados por circuitos eléctricos, que a su vez incluyen una lógica programada para realizar tareas específicas. De la misma forma los sistemas operativos están formados por cientos o miles de programas para realizar tareas específicas y controlar los circuitos. Es por esta razón que todos los sistemas informáticos son susceptibles a fallos, errores, descomposturas, entre otras, ya que son un enorme conjunto donde intervienen un gran número de elementos tanto físicos como lógicos.

Los sistemas informáticos en la actualidad no se reducen a un solo equipo o una red, en este momento podemos encontrar equipos dispersos por todo el mundo, utilizando satélites y ondas electromagnéticas para comunicarse.

Estas variables en los sistemas dan como resultado, que ningún sistema, sea 100% seguro; siempre existirán eventualidades no planeadas y no programadas que harán que falle.

1.1.2.1 Riesgo

La probabilidad de que una vulnerabilidad sea utilizada, ya sea intencional o accidentalmente ocasionando algún daño o pérdida es conocida como riego (Cole, Krutz, & Conley, 2005), (Phoha, 2002), (López Barrientos & Quezada Reyes, 2006).

1.1.2.2 Vulnerabilidad

Una vulnerabilidad es una condición originada por un defecto o debilidad en el diseño o implementación de los mecanismos de control (López Barrientos & Quezada Reyes, 2006), (OWASP, 2014), (Cole, Krutz, & Conley, 2005).

1.1.2.3 Ataque

Es un acto intencional explotando una vulnerabilidad. Si este modifica un archivo o registro, se denomina activo; mientras que uno pasivo no realiza modificaciones (OWASP, 2014), (Phoha, 2002), (Gómez Vieites, 2011).

1.1.2.4 Activos

Los activos son los recursos más importantes en una organización que requieren protección debido a su valor monetario o el costo de reposición. Algunos ejemplos pueden ser el hardware, software, información, infraestructura, servicios, personal humano, reputación, entre otros (López Barrientos & Quezada Reyes, 2006), (Gómez Vieites, 2011).

1.1.3 Servicios de la seguridad Informática

Los servicios de la seguridad informática mejoran un sistema o flujo de información mediante mecanismos de seguridad que permiten minimizar los ataques (López Barrientos & Quezada Reyes, 2006). Existen varios servicios o mecanismos de seguridad, como los mostrados en la **figura 1-1**.

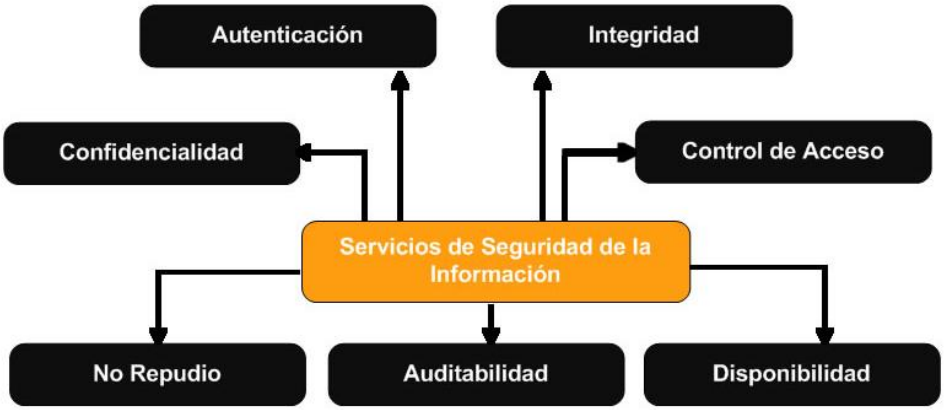


Figura 1-1 Servicios de seguridad de la información, adaptada de (Gómez Vieites, 2011).

- Confidencialidad

  Garantiza que únicamente las personas autorizadas tendrán acceso a cierta información, previniendo la divulgación de información no autorizada a terceros. (López Barrientos & Quezada Reyes, 2006), (Phoha, 2002).

- Autenticación

  Este servicio garantiza que la identidad es legítima mediante la verificación de credenciales de identificación y autorización (López Barrientos & Quezada Reyes, 2006), (ISECOM, 2010).

- Integridad

  La integridad garantiza que la información no ha sido modificada desde la fuente al destinatario, mediante firmas, sellos o cadenas de validación, entre otras (Gómez Vieites, 2011).

- No repudio

  Garantiza el envío y recepción de la información evitando la negación de las fuentes y destinos, de esta manera, el emisor puede comprobar que envió la información y el receptor no puede negar que la ha recibido (Gómez Vieites, 2011), (Phoha, 2002).

  Este es un servicio de vital importancia en las transacciones comerciales ya que permite proporcionar a los compradores y vendedores seguridad en los servicios bancarios.

- Disponibilidad

  Este servicio asegura que la información podrá ser consultada por los usuarios autorizados las veces que lo requieran cuando sea solicitada, previendo la recuperación de la información ante incidentes de seguridad o desastres naturales (Gómez Vieites, 2011), (López Barrientos & Quezada Reyes, 2006). Este es el servicio más importante, puesto que debemos tener en cuenta que los demás servicios, requieren que la información se encuentra disponible para que pueda ser utilizada.

- Autorización (control del acceso)

  Con el servicio de autorización se pretende limitar o controlar el acceso a la información una vez autenticado. Para ello, se definen unas Listas de control de Acceso (ACL) con la relación de usuarios y grupos de usuarios y sus distintos permisos de acceso (Gómez Vieites, 2011), (López Barrientos & Quezada Reyes, 2006).

- Auditabilidad

  El servicio de auditabilidad o trazabilidad permite registrar y monitorizar el acceso a la información. De este modo, es posible detectar situaciones o comportamientos anómalos llevando un control del rendimiento del sistema (Gómez Vieites, 2011).

- Reclamación de propiedad

  Este servicio garantiza que la información pertenece al titular de los derechos de autor.

En un sistema informático se puede recurrir a la implantación de diferentes técnicas y mecanismos de seguridad para ofrecer los servicios de seguridad descritos, como por ejemplo:

- Identificación de usuarios y política de contraseñas
- Copias de seguridad y respaldo
- Cifrado de las comunicaciones
- Huella digital
- Firma electrónica
- Protocolos criptográficos
- Análisis y filtrado del tráfico (cortafuegos)
- Servidores proxy
- Sistema de detección de intrusos (IDS) y Antivirus

1.1.4 Gestión de la Seguridad Informática

En la gestión de la seguridad de la información se contemplan mecanismos y procedimientos que permiten garantizar ciertos niveles de seguridad, plasmándolos en un documento que describe plenamente las responsabilidades y tareas de seguridad planificadas (Phoha, 2002), este documento es llamado: Sistema de Gestión de la Seguridad de la Información (SGSI) (Gómez Vieites, 2011) .

Tomando en cuenta que los riesgos no se pueden eliminar completamente, resulta imposible alcanzar una seguridad completa o del 100%; es por esta razón que se suele hablar de fiabilidad del sistema.

El SGSI comprende la política, la estructura, los recursos, los procedimientos y los procesos para implantar la gestión de la seguridad de la información en una organización (Gómez Vieites, 2011), tomando en cuenta normas reguladoras, procedimientos, reglas y buenas prácticas que determinan el modo en que todos los activos y recursos, incluyendo la información, son gestionados, protegidos y distribuidos.

Para establecer y gestionar un SGSI en base a ISO 27001, se utiliza el ciclo continuo PDCA, tradicional en los sistemas de gestión de la calidad (iso27000.es, 2013), como se muestra en la **figura 1-2**.

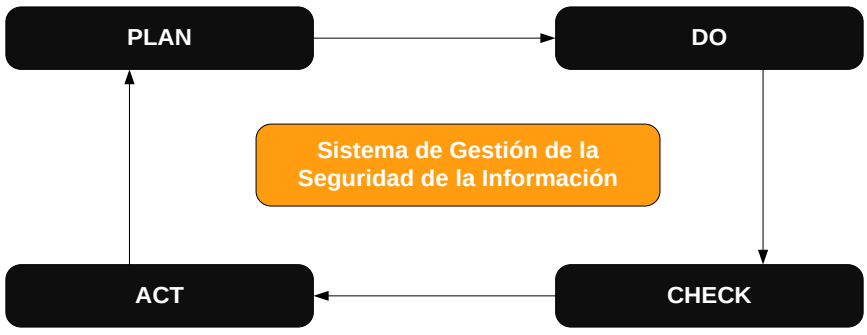


Figura 1-2 SGSI, adaptada de (Gómez Vieites, 2011).

- ***Plan*** (planificar): establecer el SGSI, selección y definición de medidas y procedimientos.

- ***Do*** (hacer): implementar y utilizar el SGSI, implantación de medidas y procedimientos de mejora.
- ***Check*** (verificar): monitorizar y revisar el SGSI, comprobación y verificación de las medidas implantadas.
- ***Act*** (actuar): mantener y mejorar el SGSI, actuación para corregir todas las deficiencias detectadas en el sistema.

En todo este proceso es necesario contemplar un modelo que tenga en cuenta los aspectos tecnológicos, organizativos, el cumplimiento del marco legal y la importancia del factor humano.

De igual forma resulta de vital importancia conseguir el apoyo completo por parte de la organización, ya que ésta debe proporcionar los medios y la autoridad suficiente para poder definir e implantar las políticas y procedimientos de seguridad, dotando además los recursos técnicos y humanos necesarios.

De hecho, en algunas organizaciones se ha definido la figura del Responsable de Gestión de Seguridad de la Información, conocido en inglés por sus siglas CISO (*Chief Information Security Officer*) (Gómez Vieites, 2011).

Existen varios modelos de madurez en la Gestión de la Seguridad de la Información (Chapin & Akridge, 2005) para una organización como se muestra en la **tabla 1-1**.

Tabla 1- 1 Modelos de madurez de seguridad publicados, adaptada de (Chapin & Akridge, 2005).

Modelo	Descripción	Comentarios
Modelo de Madurez de Seguridad TI de NIST - CSEAT	Cinco niveles de madurez progresiva: 1. Política 2. Procedimiento 3. Implantación 4. Prueba 5. Integración	Centrado en niveles de documentación
Modelo de Evaluación de la Seguridad de la Información de ***Citigroup*** (CITI-ISEM)	Cinco niveles de madurez progresiva: 1. Autocomplacencia 2. Reconocimiento 3. Integración 4. Prácticas comunes 5. Mejora continua	Centrado en concienciación y adopción por parte de la organización
Modelo de madurez de COBIT®	Cinco niveles de madurez progresiva: 1. Inicial / ad hoc 2. Repetible pero intuitivo 3. Proceso definido 4. Gestionado y medible 5. Optimizado	Centrado en procedimientos específicos de auditoría
Modelo SSE-CMM	Cinco niveles de madurez progresiva: 1. Realizado informalmente 2. Planificado y perseguido 3. Bien definido 4. Controlado cuantitativamente 5.Continuamente mejorado	Centrado en ingeniería de seguridad y diseño de software
Evaluación de la Capacidad de Seguridad de CERT/CSO	Cinco niveles de madurez progresiva: 1. Existente 2. Repetible 3. Persona designada 4. Documentado 5. Revisado y actualizado Mide usando cuatro niveles: 1. Inicial 2. En desarrollo 3. Establecido 4. Gestionado	Centrado en la medición de la calidad relativa a niveles de documentación

1.1.5 Perímetro de Seguridad

Durante los primeros años de las redes, los ataques a sistemas informáticos requerían pocos conocimientos técnicos. Los ataques realizados desde el interior de la red se basaban en la alteración de permisos para modificar la información del sistema. Mientras que los ataques externos se producían gracias al conocimiento de la contraseña para acceder a los sistemas.

Con el paso de los años se han desarrollados nuevos ataques cada vez más sofisticados para explotar vulnerabilidades tanto en el diseño de las redes como en la configuración y operación de los sistemas informáticos que conforman las redes conectadas a Internet.

Las técnicas de ataque se han ido automatizando día a día, por lo que en muchos casos sólo se necesitan conocimientos muy básicos para realizarlos. Actualmente, cualquier usuario con una conexión a Internet tiene acceso a numerosas herramientas, aplicaciones, manuales y tutoriales para realizar estos ataques.

Para identificar si un ataque es externo o interno se requiere implementar un perímetro de seguridad. El perímetro de seguridad es la frontera física o lógica que define un dominio o zona de seguridad que contienen los equipos y dispositivos acreditados para estar en ella y en la que se aplica una determinada política de seguridad o se ha implantado una determinada arquitectura de seguridad (Mañas, 20013), (Phoha, 2002).

1.1.5.1 Componentes del Perímetro de Seguridad

El perímetro de seguridad está conformado por 2 elementos como se muestra en la **figura 1-3**.

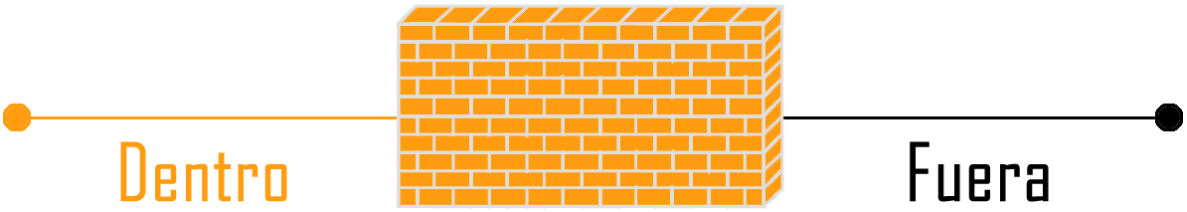


Figura 1-3 Perímetro de seguridad.

- Agente interno: Elementos con acceso al sistema dentro del perímetro de seguridad, es decir, con cierta autorización para el acceso.
- Agente externo: Elementos que acceden al sistema desde el exterior del perímetro de seguridad.

Para incrementar el nivel de seguridad se suele realizar seguridad en profundidad, agregando tantas delimitaciones como sea conveniente para separar las diferentes zonas de la red, como se muestra en la **figura 1-4**.

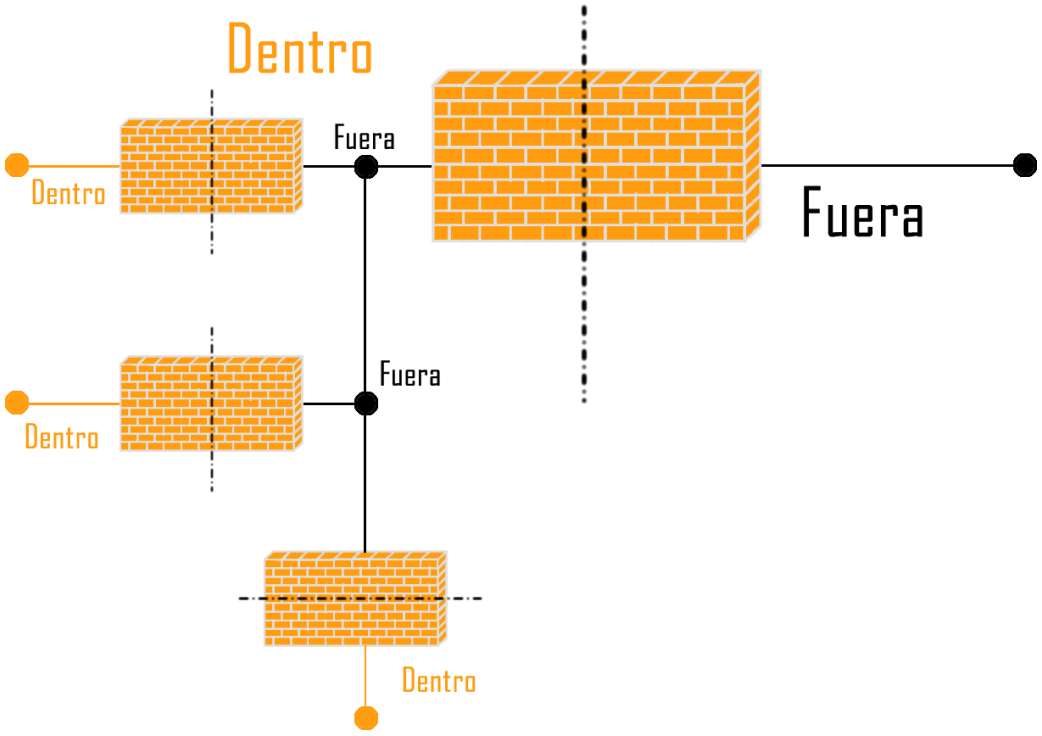


Figura 1-4 Seguridad en profundidad

1.1.6 Clasificación de amenazas

Una amenaza se representa a través de una persona, evento, circunstancia, fenómeno o idea maliciosa, que pretenden provocar daño cuando existe un acceso no autorizado, una violación de seguridad, destrucción, divulgación modificación de datos o denegación del servicio. (López Barrientos & Quezada Reyes, 2006), (Phoha, 2002). Las amenazas pueden clasificarse en:

- Naturales: inundación, incendio, tormenta, fallas eléctricas, explosión, etc.
- Externas: virus informáticos, ataques de una organización, sabotajes, disturbios y conflictos sociales, intrusiones en la red, robo de información, estafa, etc.
- Internas: empleados descuidados con una formación inadecuada o descontentos, errores en la utilización de las herramientas, recursos y el factor humano.

Adicionalmente tomando en cuenta el grado de intención se clasifican como:

- Accidentales: descomposturas del *hardware*, fallas en el software, inundación, incendio, etc.
- Errores: de utilización, de explotación, de ejecución, etc.
- Acciones malintencionadas: robo, fraude, sabotaje, intrusiones, etc.

1.1.7 Procedimiento general de una intrusión

El procedimiento general de una intrusión consiste en una serie de pasos que permiten obtener información valiosa, vulnerando la seguridad de una organización, equipo o aplicación.

En caso de que el procedimiento se realice a manera de simulación de ataque, se le conoce como *Penetration Test* (PenTest) o prueba de penetración, donde puede

permitirse o no la explotación de vulnerabilidades con el fin de evitar dañar los activos.

Este tipo de prueba se realiza para conocer la situación actual de seguridad en una organización, red, aplicación o equipo. De acuerdo con (PTES, 2012), (Mieres , 2009) y (Pacheco & Jara, 2012) el procedimiento general de una intrusión consta de las siguientes fases:

- Reconocimiento
- Escaneo
- Explotación
- Aseguramiento del acceso
- Destrucción de evidencia
- Retroalimentación

1.1.7.1 Reconocimiento

La fase de reconocimiento es la que más tiempo consume en una intrusión, ya que busca identificar el objetivo, obtener la mayor cantidad de información para posteriormente utilizarla.

Algunas técnicas en esta fase serían; la recopilación de direcciones de correo, ubicación geográfica, obtener información del personal, organigramas, direcciones IP, proveedores de servicios, sitios de Internet, aplicaciones que utilizan, recolección de la basura, ingeniería social entre otras (Pacheco & Jara, 2012).

1.1.7.2 Escaneo

La fase de escaneo consiste en obtener todos los errores y vulnerabilidades posibles de acuerdo a la información recabada (usuarios, grupos, nombres de

equipos, recursos de red, recursos compartidos y servicios brindados), a fin de explotar estas vulnerabilidades posteriormente (Gómez Vieites, 2011), (Pacheco & Jara, 2012).

En esta fase se emplean técnicas como, la detección de sistemas operativos, escaneo de puertos, identificación de servicios y escaneo de vulnerabilidades.

1.1.7.3 Explotación

Si la fase de explotación se encuentra dentro de un PenTest puede tomarse la decisión de no intentar explotar las vulnerabilidades, sino simplemente llegar hasta el conocimiento de las mismas y sus posibles repercusiones. (Pacheco & Jara, 2012).

Esta fase proviene de la palabra *exploit*, que se refiere a aprovecharse de un error, falla o vulnerabilidad a fin de ocasionar un funcionamiento inadecuado del sistema o aplicación, permitiendo realizar cambios en el flujo de ejecución normal del programa a fin de controlar los resultados.

1.1.7.4 Aseguramiento del acceso

Se podría pensar que una vez que un atacante obtiene el control de un sistema y accede regularmente a este, su tarea está terminada y finalizará sus actividades (Pacheco & Jara, 2012). No obstante, el acceso es el primer paso. Puesto que un atacante desea mantener el control del sistema indefinidamente, procede a la instalación de puertas traseras o mecanismos que le permitan mantener el acceso.

En este paso se planea la conservación del acceso con los mayores privilegios posibles que permitan reingresos al sistema en el futuro; en algunos casos los

atacantes llegan incluso a parchar (solventar) la vulnerabilidad utilizada para el acceso con el fin de que otros intrusos no puedan utilizarla.

1.1.7.5 Destrucción de evidencia

Un aspecto muy importante y que es vital para los atacantes es el no dejar huella, ya sea en la red o en los equipos atacados y en el peor de los casos minimizarlos (Pacheco & Jara, 2012).

Un atacante con poca experiencia pensaría únicamente en tener éxito, mientras que uno más avanzado pensará minuciosamente en la forma de pasar desapercibido, borrar sus rastros como si nada hubiera pasado.

1.1.7.6 Retroalimentación

Esta fase poco conocida como retroalimentación o documentación nos permite evaluar los resultados de la intrusión realizada (PTES-Reporting, 2011).

En caso de que se tratara de un PenTest se realizaría un informe técnico y ejecutivo mostrando los resultados obtenidos, los alcances, además de proveer las posibles soluciones permitiendo así cubrir las vulnerabilidades encontradas.

En caso de ser un ataque real, el atacante evaluaría sus resultados y el valor de estos, permitiéndole desarrollar mecanismos más avanzados de ataque contra otros objetivos utilizando los recién adquiridos, o bien, pedir rescate por ellos.

Capítulo 2

Seguridad perimetral en redes

En este capítulo se mostrará que es la seguridad perimetral, cómo se implementa, ventajas y algunos de los equipos más representativos que nos ayudan a implementarla de manera lógica.

El término de seguridad perimetral es bastante amplio y ha tenido diversas atribuciones a lo largo del tiempo. En un principio este término se aplicaba al hecho de evitar accesos no autorizados a los sistemas de cómputo mediante seguridad física, evitando el acceso a personas ajenas o sin autorización a los equipos en las salas de cómputo.

Hoy en día la seguridad física no es suficiente, el acceso a los servicios se realiza desde Internet, permitiendo a cualquier usuario ingresar a los mismos.

De esta forma nace la necesidad de seguridad lógica, permitiendo mediante hardware y software controlar el acceso a los sistemas de cómputo.

Sin embargo, estos equipos especializados tienen un costo elevado para instituciones u organizaciones con recursos económicos limitados, por lo que estas optan por soluciones en software libre.

2.1 Seguridad perimetral

La seguridad perimetral es la infraestructura y elementos de red que proveen de seguridad dentro del perímetro, en una red interna, ante otra que generalmente es Internet, de tal forma que genera una “coraza” que protege a todos los elementos sensibles de ser atacados dentro del perímetro de seguridad.

Esto implica que cada paquete de red debe de ser inspeccionado, analizado, aceptado o rechazado de acuerdo a las políticas de seguridad en nuestra red. (INTECO, 2011), (Ramos Fraile, 2011), (Taboada Gómez, 2005).

En contraste con la **figura 2-1**, una arquitectura con seguridad perimetral sería muy parecida a la que se muestra en la **figura 2-2**.

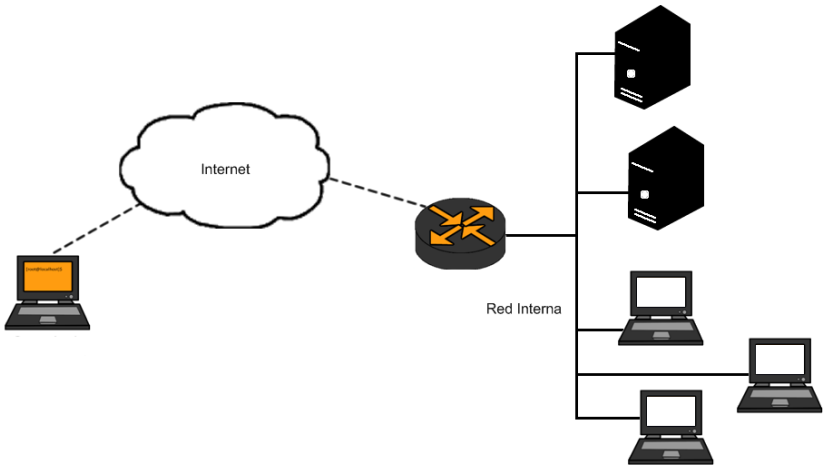


Figura 2-1 Arquitectura sin seguridad perimetral (Ramos Fraile, 2011).

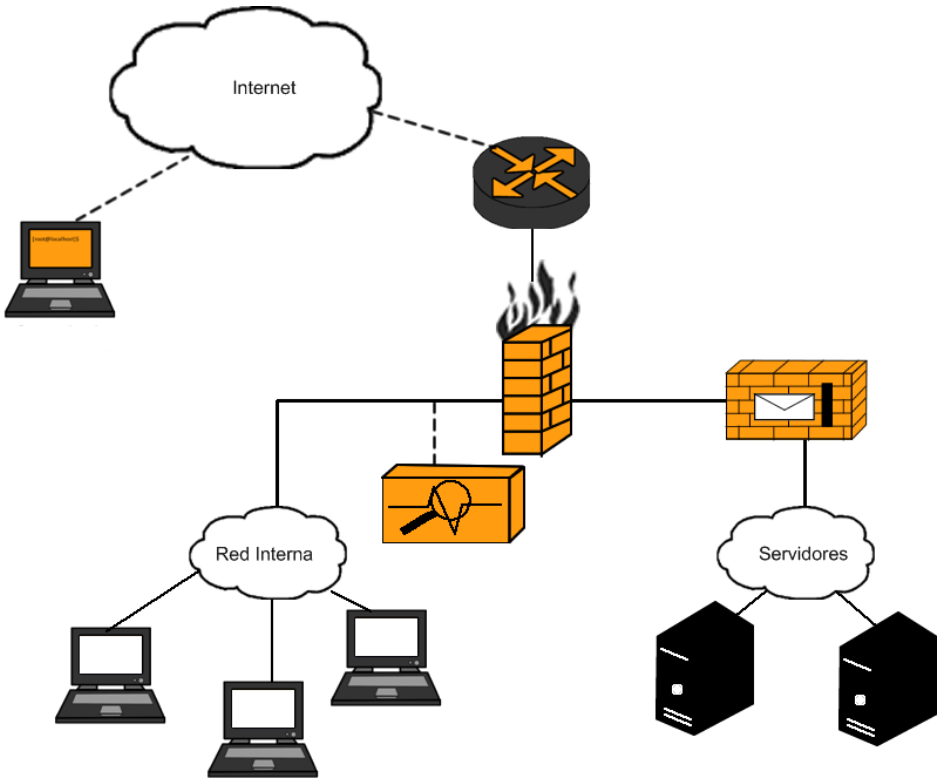


Figura 2-2 Arquitectura con seguridad perimetral (Ramos Fraile, 2011).

Para entender este concepto, es mucho más sencillo mostrar un ejemplo que no cuenta con seguridad perimetral, donde es posible observar:

- Red sin segmentar.
- Servidores internos.
- Falta de monitoreo.
- Falta filtro de tráfico de entrada o salida.
- Sin análisis de malware o spam.
- Un usuario remoto accede directamente a los servicios.

2.2 Equipos de seguridad perimetral

Existe una gran cantidad y variedad de equipos para seguridad perimetral, una buena fuente de análisis son las reseñas que se realizan en revistas especializadas así como los resultados de empresas dedicadas a la investigación y asesoramiento en temas relacionados con TI.

Cualquier implementación de seguridad perimetral basa su funcionamiento en la piedra angular llamada *Firewall*. Su implementación en conjunto con diferentes dispositivos permite implementar un perímetro de seguridad más robusto y con diferentes niveles de seguridad, dependiendo de los recursos monetarios, activos a proteger y recursos humanos con los que se cuenta. Algunos de los dispositivos para seguridad perimetral, más importantes son:

- *Firewall*
- IDS
- IPS
- Escáner de Vulnerabilidades
- Appliance contra malware
- Appliance contra SPAM
- *Honeynet*
- Herramientas de análisis
- Herramientas de integración
- Herramientas de evaluación

2.2.1 *Firewall*

Un *Firewall* es un dispositivo que realiza un filtrado de paquetes o contenido, a partir de unas reglas definidas por el administrador de la red, teniendo en cuenta las direcciones IP fuente y destino, tipo de contenido permitido y servicios de la red al que corresponden.

Cabe mencionar que un *Firewall* no inspecciona si el contenido es adecuado, o limpio, por esta razón, el malware pasa por los servicios permitidos y confiables como páginas Web o correos electrónicos.

Un *Firewall* reconoce únicamente lo que está en sus políticas, aquello que puede y aquello que no puede pasar. Este tipo de barrera evita muchos de los ataques contra

redes ya que se necesitan conocimientos de su funcionamiento y como ofuscar los ataques para evitarlos.

2.2.2 IDS

El objetivo de un IDS es la identificación de los ataques o las violaciones de seguridad a través del control de las actividades de la red y sus componentes.

Los IDS son de gran ayuda para registrar posibles intentos de acceso no autorizado, pero pueden generar muchos falsos positivos ocasionando falsas alarmas, haciendo que no se tomen precauciones con alertas muy parecidas. Es por esta razón que se debe de contar con un buen IDS y configurarlo adecuadamente para reducir estas fallas.

Una implementación de NIDS puede ser como la mostrada en la **figura 2-3**.

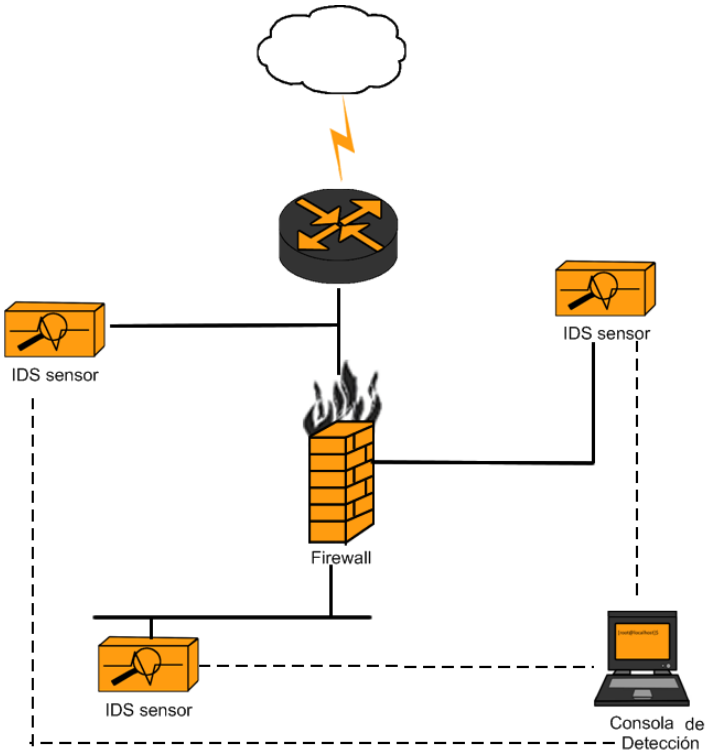


Figura 2-3 Sistema de detección de intrusiones, adaptada de (Merola, 2006)

Existen diferentes tipos de IDS, los instalados en los equipos de los usuarios (*host*) HIDS y los que se encuentran en la red (*network*) NIDS.

Un IDS puede ser comparado con una alarma contra ladrones, si se intenta algún tipo de actividad sospechosa, se activará algún tipo de respuesta. Cuantos más sensores se instalen mejor, porque cada sensor se especializa en la detección de un tipo concreto de actividad, pero, como cualquier otro sistema automatizado, un IDS puede quedar inoperativo, provocar una falsa alarma o ser eludido (Merola, 2006).

2.2.3 IPS

Los sistemas de prevención de intrusos son dispositivos o programas que se usan para detectar señales de intrusiones dentro de redes o sistemas y son capaces de realizar acciones. Estas acciones generan políticas de bloqueo de tráfico, previniendo intrusiones (Piper, 2011).

Para explicar mejor que son, podríamos utilizar la siguiente analogía. Imagine un edificio de oficinas con un guardia de seguridad en la entrada, quien permite al personal y al cartero entrar al edificio siempre y cuando no parezcan sospechosos, de lo contrario no les permitirá el acceso o generará una alerta.

Solemos utilizar un *Firewall* para protegernos de los ataques a nuestros sistemas informáticos, y emplear los sistemas de detección de intrusiones para monitorizar tales ataques. Sin embargo, en este momento la detección de intrusos no basta. ¿Qué más da que se detecte el ataque o no, si no somos capaces de resistirlo? La solución a esta interrogante son los sistemas de prevención de intrusos (Piotrowski, 2006).

Estos sistemas trabajan defendiendo contra fallas de seguridad en tiempo real, mientras mantienen una política de seguridad. De esta manera mantienen seguros los sistemas y en caso de detectar algo extraño actúan bloqueándolo.

Un IPS por lo general se encuentra en forma de *appliance*, los agentes de *software* corren en servidores o en entornos virtualizados, como se muestra en la **figura 2-4**.

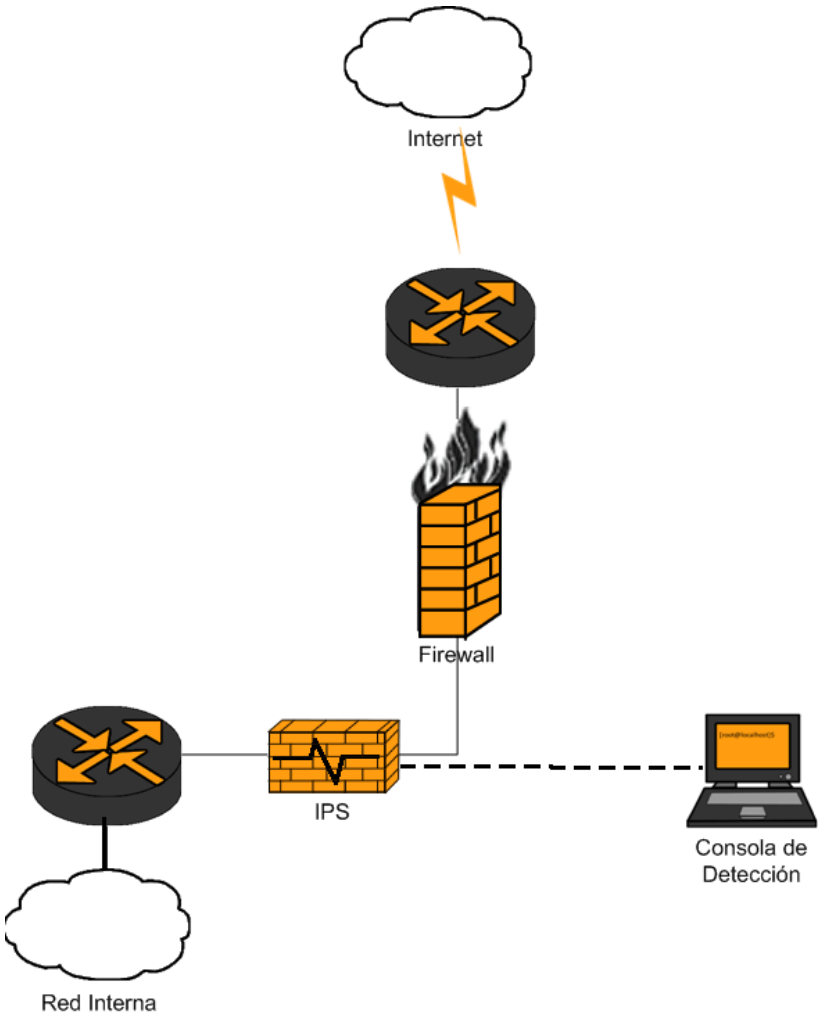


Figura 2-4 Sistema de prevención de intrusiones (Piotrowski, 2006).

2.2.4 Escáner de vulnerabilidades

Es una herramienta que permite realizar una verificación de seguridad en una red mediante el análisis de los puertos abiertos, servicios y versiones de software, entre otras cosas, en un equipo o en toda la red. Estas herramientas son de suma utilidad ya que permiten supervisar la seguridad de todos los equipos que están en la red, mostrando los riesgos y posibles soluciones a las mismas, **figura 2-5**.

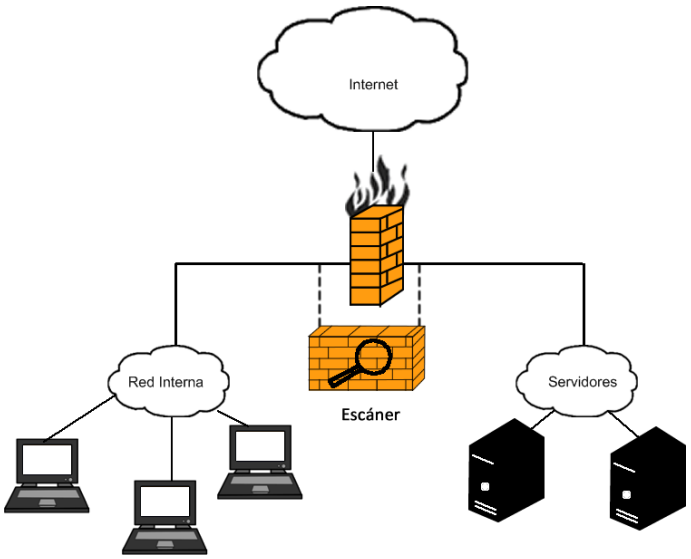


Figura 2-5 Escáner de Vulnerabilidades.

El proceso de análisis utiliza solicitudes que permiten determinar los servicios que se están ejecutando, además de analizar las respuestas, identificar el sistema operativo, las versiones de las aplicaciones, vulnerabilidades, entre otras cosas (Nidecki, 2006).

2.2.5 Appliance contra *malware*

Son dispositivos que sirven como defensa contra nuevas amenazas, cuentan con los últimos avances de análisis de firmas y detección de malware (SOPHOS, 2013), como se muestra en la **figura 2-6**.

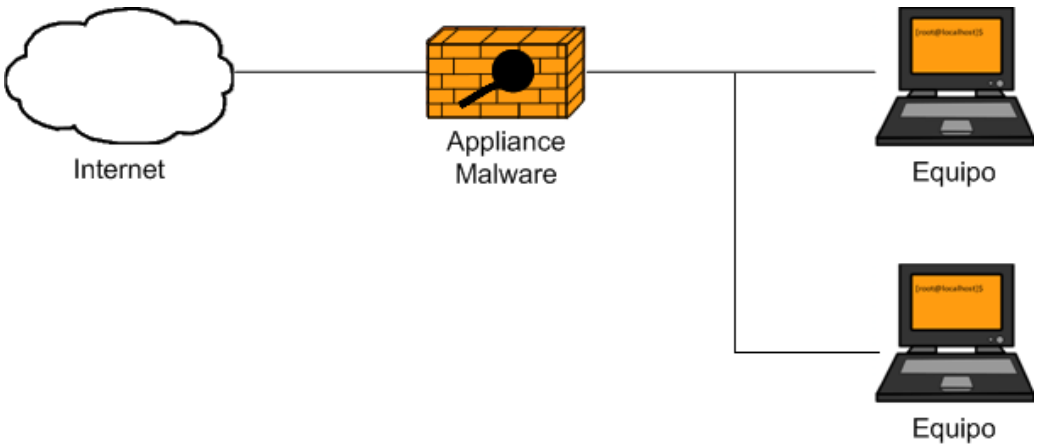


Figura 2-6 Appliance contra malware

Estos dispositivos escanean el contenido del tráfico de red en busca de firmas de malware y virus, eliminado una gran cantidad de amenazas nuevas y conocidas. La combinación con antivirus personales provee máxima seguridad sin comprometer la escalabilidad (CISCO, 2011).

2.2.6 Appliance contra spam

Estos dispositivos protegen la red contra los problemas del *spam* que se actualizan diariamente. Una arquitectura con un equipo dedicado al bloqueo de *spam* sería como la mostrada en la **figura 2-7**.

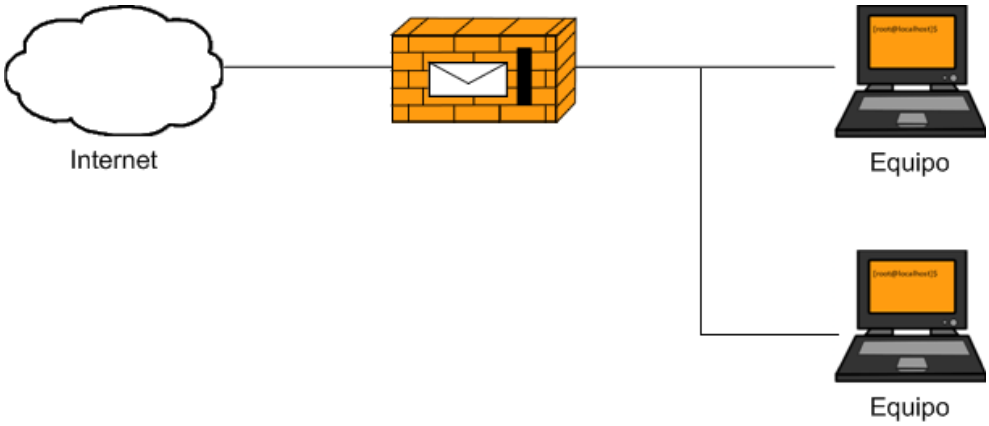


Figura 2-7 Appliance contra spam

Las medidas implementadas hace unos meses ya no son efectivas, debido a que el *spam* generado en una *botnet* es mejorado en tan solo unos minutos.

Los sistemas contra *spam* cuentan con las últimas firmas y técnicas, actualizadas día con día, previenen del uso de la red interna para el envío de spam e informan cuando la red está siendo usada para tales fines (SHOPOS, 2014).

2.2.7 *Honeynet*

Una *honeynet* o red señuelo ha sido configurada y conectada a otras redes para que pueda ser sondeada, atacada e incluso comprometida por intrusos, permitiendo aprender sobre las herramientas y técnicas utilizadas por los intrusos. Facilitan la captura de nuevos virus y códigos dañinos para su posterior análisis. A diferencia, una darknet (red oscura) es una porción de un espacio IP asignado donde no reside ningún servicio sensible (Oppleman, 2008).

Estas redes se clasifican en oscuras porque aparentemente no hay nada funcionando en ellas e incluyen por lo menos un servidor, diseñado para actuar como un colector de paquetes, como se muestra en la **figura 2-8**.

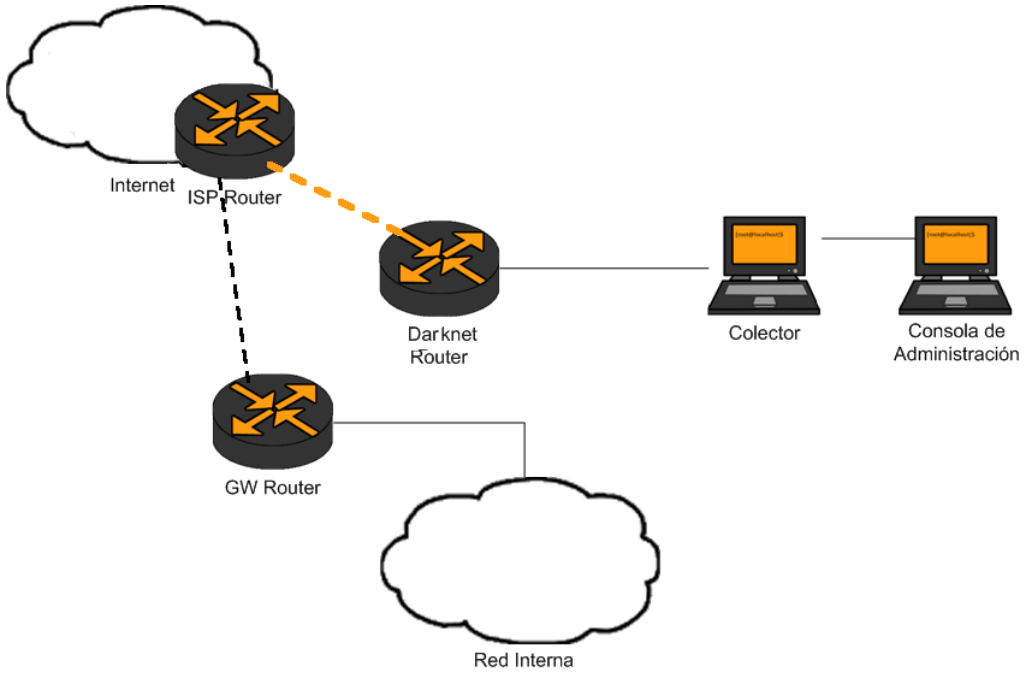


Figura 2-8 Ejemplo de una darknet (Oppleman, 2008)

Al igual que las *darknets*, una *honeynet* es en general una porción de un espacio IP asignado. Por otro lado, una *honeynet* proporciona un destino donde los paquetes van a morir, este destino imita un servicio real (o muchos servicios), y permite que

ocurra la conexión haciendo que se establezca un diálogo bidireccional completo, a diferencia de una *darknet*.

Una *honeynet* imitando un servicio real debe ser un recurso sostenido y constantemente monitoreado que tenga como objetivo atraer atacantes para sondearlos y/o infiltrarlos.

Una arquitectura donde se implementa una *honeynet* sería muy parecida a la mostrada en la **figura 2-9**.

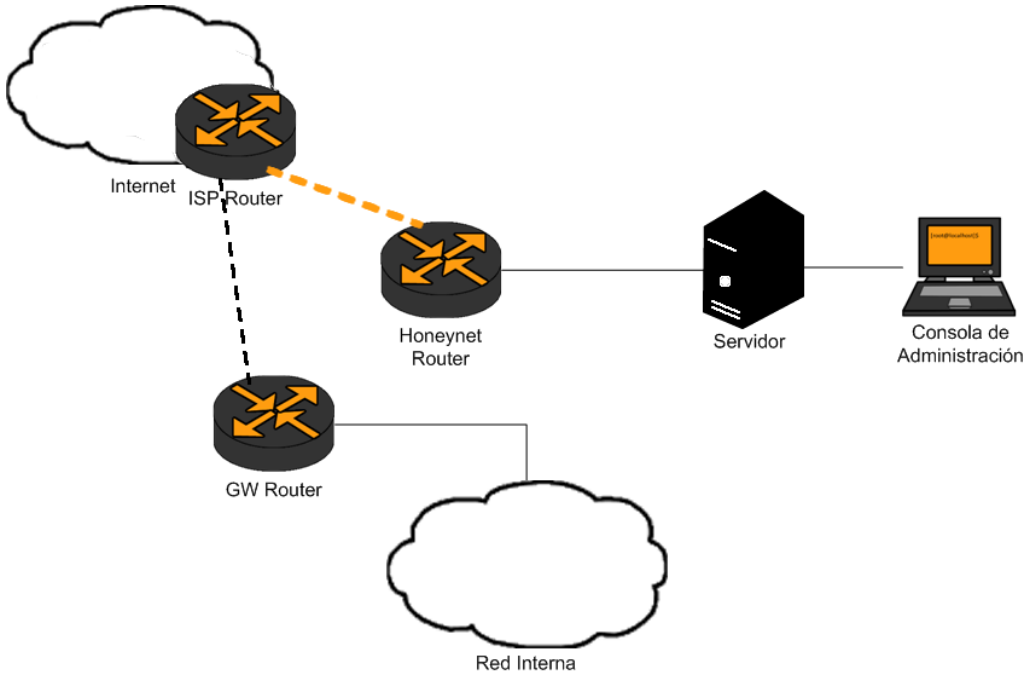


Figura 2-9 Ejemplo de una Honeynet

2.2.9 Herramientas de análisis

Las herramientas de análisis son utilizadas para el análisis forense de las vulnerabilidades, dirigiendo el análisis de tráfico de red y reportando las necesidades de seguridad.

Estas herramientas proporcionan la tecnología para identificar de manera automática los problemas, los reportes de otros dispositivos, junto con las capacidades de filtrado que van más allá de la captura, aplicación de filtros de

manera sencilla ya que posee capacidades de decodificación y análisis del tráfico de la red (Expresión Binaria, 2013). Aunado a esto, estas herramientas capturan el tráfico de red y de manera automática pueden volver a ensamblarlo a su formato nativo, haciéndolo mucho más fácil de analizar con los datos que van a través de la red.

2.2.10 Herramientas de integración

La detección de ataques en las redes corporativas es un problema difícil de solucionar, los sistemas de detección generan una gran cantidad de información por su alta sensibilidad, imposibilitando la gestión y análisis de información teniendo así una paradoja de información (Gómez Rodríguez, 2008), un ejemplo de su implementación sería como el mostrado en la **figura 2-10**.

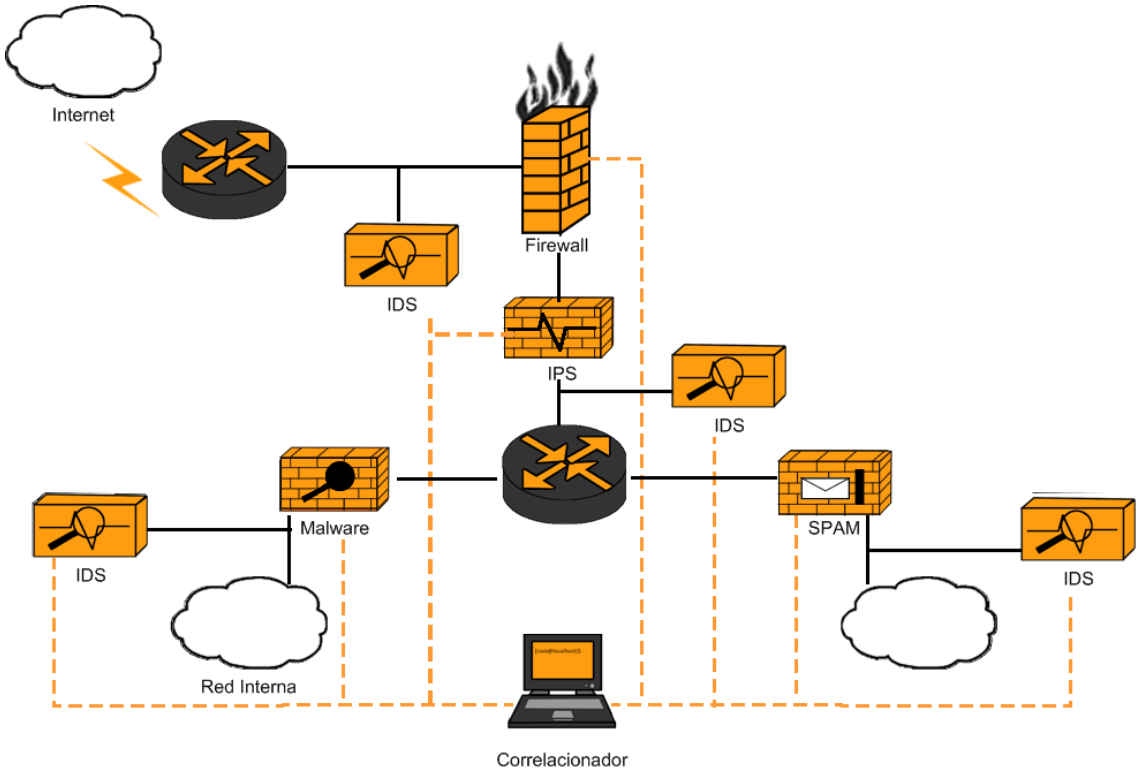


Figura 2- 10 Herramientas de Integración.

Además de que la información recabada por los sensores aparece de forma única y aislada, sin relación entre sí, dificultando la abstracción, es necesario contar con

un equipo que integre toda la información recabada, estos equipos son llamados sistemas SIM (Security Information Management).

Los sistemas SIM centralizan el proceso de análisis de eventos de seguridad, sus principales características podrían reducirse en:

- Análisis de eventos de seguridad de fuentes diversas
- Correlación de eventos para reducir falsos positivos
- Posibilitar la reacción activa, evitando daños mayores
- Análisis post-incidentes para obtener información que será usada para mejorar la seguridad de la red y la detección de nuevos ataques

La correlación es la capacidad de relacionar y procesar la información una vez que esta es homogénea, permitiendo llevar una administración centralizada de la seguridad, obteniendo información de los diferentes equipos y observando el comportamiento en la red.

2.2.11 Herramientas de evaluación

Las herramientas de evaluación están diseñadas para ayudar a las organizaciones a evaluar los puntos débiles de su entorno de seguridad de TI, presentan un listado de cuestiones ordenadas por prioridad así como orientación específica para minimizar esos riesgos, permitiendo fortalecer la seguridad de manera fácil y efectiva, realizando una serie de pruebas de manera remota a las redes y equipos, generando reportes en las que se sugieren soluciones para los problemas de seguridad (Microsoft, 2011), (insecure.org, 2003).

Estas herramientas son de vital importancia, puesto que arrojan información sobre problemas de seguridad de manera más actualizada, con la posibilidad de obtener información de cómo solucionarlas y de las consecuencias de omitirlas.

Capítulo 3

Firewall creado con "Packet Filter"

En este capítulo describiremos que es un *firewall*, cuáles son sus funciones y las ventajas y desventajas de los diferentes tipos que existen. Se mostrará la forma de implementar un firewall con Packet Filter en modo transparente y NAT.

Existe una gran cantidad de *Firewalls*, todos funcionan conceptualmente de la misma forma, regulando lo que puede pasar y que será bloqueado. Actualmente muchos equipos, antivirus o software adicional incluyen un *Firewall*, en algunos casos dentro de un modem DSL de acceso a Internet, que es un equipo provisto por un proveedor de servicios de Internet (ISP). De igual forma se incluyen en *software* de antivirus, dentro del sistema operativo, en routers inalámbricos y muchos otros.

Las variaciones y los precios de estos dependen de la cantidad de conexiones que son capaces de analizar, así como de la rapidez del análisis, mientras algunos son buenos en una red pequeña, para una empresa con cientos de usuarios estos no

son útiles, por lo que el precio se eleva exorbitantemente al comprar equipos dedicados para esta tarea.

Comparando una empresa con una escuela o universidad, el número de equipos en estas son mucho mayores, ya que cuentan con varios salones de clase con acceso a Internet, dispositivos móviles, así como servidores internos. Por esta razón, en el ámbito académico es común utilizar un *Firewall* basado en software libre dedicando un equipo diseñado para esta tarea en la sala de cómputo o cuarto de comunicaciones. Por lo general se recurren a *Firewalls* creados con Open BSD o GNU/Linux, debido a que un buen conocimiento, configuración y manejo de estos sistemas mantendrán regulado el tráfico de la red sin problemas.

3.1 Qué es un *Firewall*

Algunos autores señalan que un *Firewall* es aquel elemento en la red que se encarga de separar redes informáticas, efectuando un control del tráfico existente entre ellas (García Alfaro & Perramón Tornil, 2004). Este control permite o deniega el paso de la comunicación de una red a otra mediante el control de TCP/IP.

Para configurar un *Firewall* se debe tomar en cuenta los siguientes puntos:

- ¿Qué tipo de tráfico en la red se permite entrar y salir?
- ¿Sólo el tráfico establecido en las políticas será bloqueado?
- El equipo utilizado para esta tarea debe ser de confianza, ¿Qué equipo se utilizaría?

Existen diferentes tipos de *Firewalls*, cada uno se enfoca en diferentes elementos a permitir y bloquear, lo que ofrece ventajas y desventajas. Por ello, debemos elegir el más adecuado a nuestras necesidades o la combinación de varios para lograr un mejor desempeño en la red.

Este tipo de *Firewall* se encarga de encaminar los paquetes de una red a otra mediante una serie de reglas de filtrado que le indican de qué manera proceder, cuales son aceptadas y cuales descartadas, como se indica en la **figura 3-1**.

Figura 3-1 Firewall de filtrado de paquetes

Estas reglas de filtrado permiten determinar si a un paquete de red le está permitido o bloqueado el paso de un lado a otro de la red. Al trabajar sobre la capa de red adquiere información para su funcionamiento mediante *Internet Protocol* (IP), *User Datagram Protocol* (UDP) y *Transmission Control Protocol* (TCP) con lo que se obtiene:

- Dirección de origen y de destino.
- Protocolo e indicadores especiales.
- Puertos de origen y de destino o tipos de mensaje (según el protocolo).
- Contenido de los paquetes.
- Tamaño del paquete.

Cada paquete que llegue al *Firewall* será comparado con una serie de reglas, hasta encontrar alguna coincidencia, ejecutando la acción indicada en la regla, como denegar, aceptar, redirigir, etc. Si el paquete no encuentra coincidencia con alguna regla, se utilizará la política por defecto. Si la política por defecto es restrictiva, el paquete será descartado, mientras que una en una permisiva el paquete será aceptado.

Una política restrictiva es difícil de configurar debido a que el administrador debe de conocer muy bien su red, tipos de contenidos y direcciones. Si no se establece que

puedan pasar ciertos paquetes estos nunca pasarán ocasionando problemas en la red. Por otro lado, una política permisiva, es sencilla de configurar, pero incrementa los riesgos de seguridad al no bloquear aquello que se desconoce.

3.3 *Firewall* de aplicación (*Application Firewall*)

Un *Firewall* a nivel de aplicación, conocido también como servidor intermediario o *proxy*, no encamina los paquetes a nivel de red sino que actúa como retransmisor, como se muestra en la **figura 3-2**.

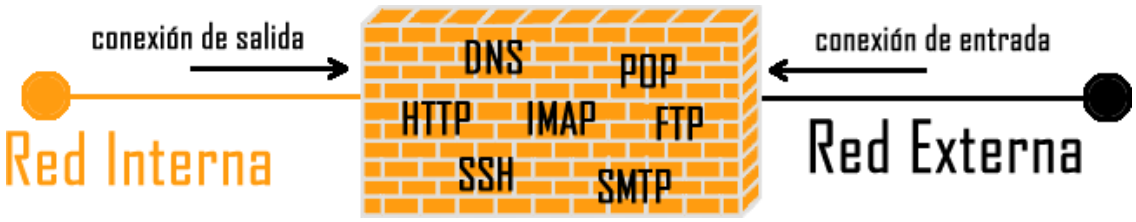


Figura 3-2 Firewall de aplicación

Cuando los usuarios de una red intentan conectarse con el servidor remoto, primero deben conectarse con el *proxy*, este abre un canal de comunicación con el servidor remoto, una vez que tiene respuesta del exterior, la retransmite al usuario que realizo la petición de manera transparente, como se muestra en la **figura 3-3**.

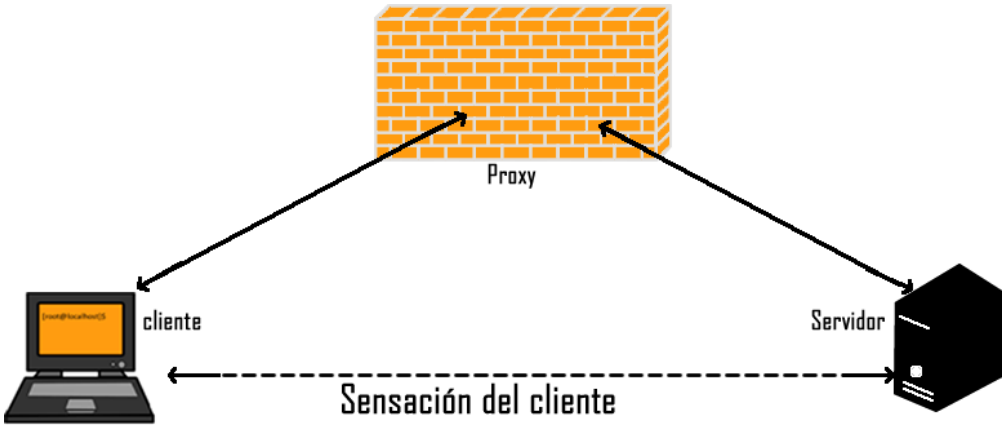


Figura 3-3 Funcionamiento de un Firewall de aplicación

Este tipo de *Firewall* puede analizar el contenido de los paquetes, dando una mayor seguridad sobre el contenido que es permitido y el rechazado, de forma más explícita que una simple dirección IP. Esto puede producir que en una red grande el rendimiento disminuya, porque el contenido será analizado antes de ser retransmitido.

De manera práctica, se recomienda utilizar ambos, primero uno filtro de paquetes para bloquear la mayor cantidad de paquetes de forma rápida y posteriormente filtrar el contenido aceptando sólo los permitidos.

3.4 *Firewall* por estados (*Stateful Packet Filtering*)

Un *Firewall* por estados permite tener un registro del estado, en las conexiones de red, mediante atributos de inicio, fin y errores.

Una vez que la conexión es permitida, ya no es analizada sino que se mantiene el registro y seguimiento, como se muestra en la **figura 3-4**.

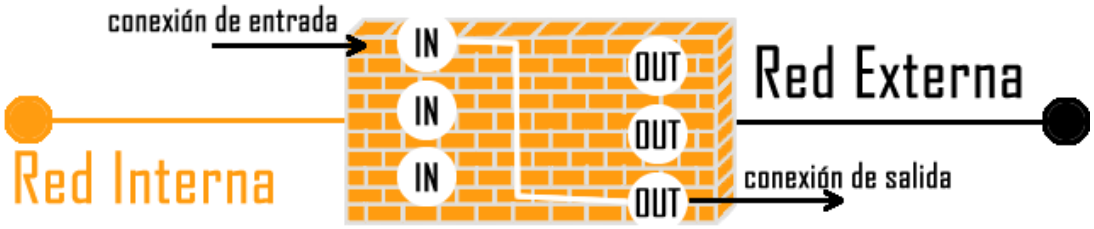


Figura 3-4 Funcionamiento de un Firewall *por estados*

Las características de seguridad que ofrecen estos dispositivos consisten en determinar que conexiones están permitidas, el número de conexiones y destino de éstas, permitiendo que el funcionamiento sea más eficiente, de manera más rápida, puesto que se analiza el comportamiento y no el contenido.

3.5 "*Packet Filter*"

Packet Filter es un *Firewall* para el sistema operativo OpenBSD, comúnmente nos referimos a este simplemente con la abreviatura "pf", fue desarrollado por Daniel Hartmeier y publicado en Diciembre del 2001 (Hansteen, The book of pf, 2008) se incluye dentro del cualquier sistema BSD, formando parte de sus herramientas básicas.

Algunas de las funciones de "pf" son el filtrado de direcciones, la creación de *Network Access Translator* (NAT), redirección de puertos y balanceo de carga.

3.6 Instalando OpenBSD

Para utilizar "pf", primero se debe obtener una fuente de instalación de OpenBSD, descargándola libremente desde cualquier repositorio oficial.

El proceso de instalación es un poco diferente a los sistemas GNU/Linux, por ello se mostrarán los pasos para su instalación.

En la **figura 3-5** y **figura 3-6** podemos ver los primeros pasos de su instalación.

```
Welcome to the OpenBSDxi386 5.5 installation program.
(I)nstall, (U)pgrade (A)utoinstall or (S)hell? I_

Choose your keyboard layout ('?' or 'L' for list) [default] es
System hostname? (short form, e.g. 'foo') ixkan_
Available network interfaces are: em0 eml vlan0.
IPv4 address for em0? (or ’dhcp’ or ’none’) [dhcp] none
IPv6 address for em0? (or ’rtsol’ or ’none’) [none] none
Available network interfaces are: eM0 vlan0.
Which one do you wish to configure? (or 'done') [em0] done
DNS domain name? (e.g. ’bar.com’) [my, domain] _
DNS nameservers? (IP address list or 'none') [none]
```

Figura 3-5 Configurando parámetros de red, idioma y hostname.

```
Password for root account? (will not echo)
Password for root account? (again) _
Start sshd(8) bydefault? [yes]
Start ntpd(8) bydefault? [no]
Do you expect torun the X WindowSystem? [yes]   no
Change the default console to com0? [no]
Setup a user? (enter a lower-case loginname or 'no') [no]
```

Figura 3-6 Asignando password de root, ssh y x-window.

A continuación, en la **figura 3-7** se indica el disco duro en que se instalará el sistema base.

```
Avaliable disks are: wd0.
Which one is the root disk? (or 'done') [wd0]
Use DUIDs rather than device names in fstab? [yes]
MDR has invalid signature; not showing it.
Use (W)hole disk or (E)dit the MDR? [whole]
```

Figura 3-7 Indicando el disco duro.

Una vez seleccionado el disco duro es necesario particionarlo. Es posible seleccionar el particionamiento automático, pero para nuestros fines es necesario utilizar uno manual, **figura 3-8**.

```
The auto-allocated layout for wd0 is:
#      size       offset    fstype  [fsize bsize cpg]
a:   1024.0M          64    4.2BSD    2048 16384   1 #  /
b:    127.1M     2097216      swap
c:  41042,0M           0    unused
d:   2851.9M     2357440    4.2BSD    2048 16384   1       #  /tmp
e:   4223.0M     8198144    4.2BSD    2048 16384   1       #  /var
f:   2048.0M    16846944    4.2BSD    2048 16384   1       #  /usr
g:   1024,0M    21041248    4.2BSD    2048 16384   1       #  /usr/X11R6
h:   5462,9M    23138400    4.2BSD    2048 16384   1       #  /usr/local
i:   2048.0M    34326400    4.2BSD    2048 16384   1       #  /usr/src
j:   2048.0M    38520704    4.2BSD    2048 16384   1       #  /usr/obj
k:  20184.1M    42715008    4.2BSD    2048 16384   1       #  /home
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] C_
```

Figura 3-8 Particionamiento manual (Custom)

Al seleccionar el particionamiento manual, se mostrará el editor de particiones, **figura 3-9**.

```
You will now create an OpenBSD disklabel inside the OpenBSD MBR
partition. The disklabel defines how OpenBSD splitsup the MBR partition
into OpenBSD partitions in which filesystems and swap space are created.
You must provide each filesystem's mountpoint in this program.

The offsets used in the disklabel are ABSOLUTE, i.e. relative to the
start of the disk, MOT the start of the OpenBSD MBRpartition.

Label editor (enter '?' for help at any prompt)
> _
```

Figura 3-9 Editor de particiones

La **figura 3-10** muestra la creación de particiones para el sistema base y el área de intercambio.

```
> a a
offset: [64]
size: [84052016] 2000M
Rounding to cylinder: 4096511
FS type: [5.3 BSD]
mount point: [none] /
> a b
offset: [4096544]
size: [79955536] 2000M
Rounding to cylinder: 4080541
FS type: [5.3 BSD]
mount point: [swap]
> a d
offset: [8177085]
size: [75874995] 6000M
Rounding to cylinder: 12289725
FS type: [5.3 BSD]
mount point: [none] /usr
```

Figura 3-10 Agregando particiones

Para la instalación, se debe considerar la mayor cantidad de espacio disponible en la partición /var, **figura 3-11**.

```
>a e
offset: [20466784]
size: [63585296]
FS type: [5.3 BSD]
mount point: [none] /var
```

Figura 3-11 Agregando partición (/var)

La **figura 3-12** muestra la finalización del particionamiento escribiendo la información en el disco duro.

```
> w
> q
```

Figura 3-12 Finalizando y escribiendo datos en el disco

De esta forma el sistema se encuentra configurado, pero aún le faltan paquetes, la **figura 3-13**, muestra la selección de los orígenes de instalación de paquetes.

```
Let's install the sets!

Location of sets? (cd disk ftp http or 'done') [cd]
Available CD-ROMs are: cd0.
Which one contains the install media? (or 'done') [cd0]
Location of sets? (cd disk ftp http or 'done') [cd]

Pathname to the sets? (or 'done') [5.5/i386]
```

Figura 3-13 Orígenes de instalación

El origen de instalación es el mismo CD de instalación, en la **figura 3-14**, se muestran los paquetes básicos que se pueden instalar.

```
Select sets by entering a set name, a file name pattern or 'all'.
De-selectsets by prepending a to the set name, file name pattern or 'all*.
Selected sets are labelled '[X]'.
[X] bsd        [X] etc55.tgz  [X] xbase55.tgz  [X] xserv55.tgz
[X] bsd.rc     [X] comp55.tgz [X] xetc55.tgz
[ ] bsd.mp     [X] Man55.tgz  [X] xshare55.tgz
[X] base55.tgz [X] game55.tgz [X] xfont55.tgz
Set names(s)? (or 'abort' or 'done') [done]
Directory does not contain SHA256.sig. Continue without verification? [no] yes
```

Figura 3-14 Seleccionando los paquetes a instalar

La **figura 3-15** muestra la instalación de los paquetes.

```
Bsd          100% |***************************************| 8809 KB 00:06
bsd.rd       100% |***************************************| 6227 KB 00:06
base55.tgz   100% |***************************************|53505 KB 00:59
etc55.tgz    100% |***************************************|  507 KB 00:00
comp55.tgz   100% |***************************************|56767 KB 00:59
man55.tgz    100% |***************************************| 9121 KB 00:11
game55.tgz   100% |***************************************| 2568 KB 00:01
xbase55.tgz  100% |***************************************|11581 KB 00:12
xetc55.tgz   100% |***************************************|71655 KB 00:00
xshare55.tgz 100% |***************************************| 2969 KB 00:04
xfont55.tgz  100% |***************************************|38484 KB 00:37
xserv55.tgz  100% |***************************************|20720 KB 00:20
Location of SETS? (cd disk ftp http or 'done') [done]
```

Figura 3-15 Paquetes instalados

Posteriormente se solicita la información de zona horaria, en el caso de México se utiliza “Mexico/General”, **figura 3-16**.

```
What timezone are you in? ('?' For list) [Canada/Mountain] Mexico/General
```

Figura 3-16 Seleccionando la zona horaria

Finalmente, se mostrará el mensaje de felicitación por haber concluido la instalación de OpenBSD, **figura 3-17**.

```
Saving configuration fi les...done.
Generating initial host.random file...done.
Making all device nodes...done.

CONGRATULATIONS! Your OpenBSD install has been successfully completed!
To boot the new system, enter 'reboot' at the command prompt.
When you login to your new system the first time, please read your mail
using the 'mail' command.
```

Figura 3-17 Instalación finalizada

Una vez terminado, es necesario reiniciar y arrancar el sistema OpenBSD.

3.7 Habilitando “*Packet Filter”*

Para el funcionamiento de “pf” como *firewall* se requiere que el sistema cuente con al menos dos interfaces de red, **figura 3-18**, así como determinar si el funcionamiento será en modo Transparente o NAT.

```
em0: f1ags=8802<BR0ADCAST, SIMPLEX, MULTICAST> mtu 1500
       lladdr 08:00:27:4d:70:8c
       priority: 0
       media: Ethernet autoselect (1000baseT full-duplex)
       status: active
em1: f1ags=8802<BR0ADCAST, SIMPLEX, MULTICAST> mtu 1500
       lladdr 08:00:27:d6:3f:b8
       priority: 0
       media: Ethernet autoselect (1000baseT full-duplex)
       status: active
```

Figura 3-18 Interfaces de red

Como primer paso, se prepararan las interfaces para su funcionamiento, para hacerlo se debe de escribir "*up"* en el archivo de configuración de las interfaces de red, por ejemplo "/etc/hostname.em0" y "/etc/hostname.em1", **figura 3-19**.

```
# echo up > /etc/hostname.em0
# echo up > /etc/hostname.em1
# sh /etc/netstart
```

Figura 3- 19 Habilitando interfaces

Para que el sistema inicie con "pf", desde el arranque se debe establecer "*YES*" en el archivo de arranque /etc/rc.conf, **figura 3-20**. De esta manera, se cargarán las reglas de /etc/pf.conf cuando el sistema operativo arranque.

```
pf = YES            #Packet filter / NAT
```

Figura 3-20 Habilitando "pf"

3.8 *Firewall* en modo transparente

Para habilitar el modo transparente se deben de configurar las interfaces y generar un puente entre ambas de tal forma que los paquetes de red fluyan sin modificaciones, para lograr esto se genera el archivo "/etc/hostname.bridge0", con la información **figura 3-21**.

```
# cat /etc/hostname.bridge0
add em0
add em1
up
```

Figura 3-21 Generando puente de interfaces

Reiniciando los servicios de red se verifica el funcionamiento, **figura 3-22**.

```
Bridge0: flags=41<UP, RUNNING>
```

Figura 3-22.Interfáz puente

Un *Firewall* en modo transparente permite implementarlo en cualquier punto de la red sin tener que realizar alguna modificación en el direccionamiento de los equipos en la red. Esto permite filtrar paquetes de red sin modificar la topología de la misma.

3.9 *Firewall* en modo NAT

Para habilitar el modo NAT se deben de configurar las interfaces para permitir el reenvío de paquetes en el Sistema Operativo.

Para ello se debe modificar el archivo "/etc/sysctl.conf" y descomentar la línea que contiene "#net.inet.ip.forwarding=1", **figura 3-23**. De igual forma se debe eliminar el puente entre interfaces, en caso de existir.

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

Figura 3- 23.Habilitando el reenvío de paquetes

Cabe recalcar que para que el reenvío de paquetes funcione, es necesario reiniciar el sistema.

La traducción de direcciones de red (NAT) es el proceso de modificar la información del encabezado de los paquetes IP para que puedan ser enrutados hacia el destino requerido. Se utiliza en los routers para permitir a varios equipos conectarse a otra red a través de la IP externa del equipo.

Un Firewall en modo NAT permite filtrar el contenido mientras se realiza el proceso de reenvió y traducción de direcciones.

Capítulo 4

Desarrollo del sistema gráfico para la gestión de un *Firewall* del tipo "Packet Filter"

Este capítulo es el tema principal de este trabajo, se describe la arquitectura, el funcionamiento, la estructura y características del sistema gráfico, junto con las pruebas que se realizaron durante su desarrollo.

El *Firewall* se considera la primera medida de seguridad que debe tener cualquier infraestructura de red, ya que se implementa con reglas simples que indican qué se puede y qué no se puede hacer en ella.

Existen una gran cantidad de *Firewalls*, los comerciales se suelen utilizar con mucha facilidad, son efectivos, cuentan con una interfaz administrativa sin necesidad de

comandos, tienen respaldos de configuración, están adaptados a múltiples idiomas, funcionan en forma Transparente y NAT.

Los que están basados en software libre son tan buenos como los comerciales, pero tienen un inconveniente, sólo pueden ser utilizadas por un usuario con conocimientos de sistemas operativos tipo Unix, línea de comandos, redes, protocolos y conocimiento del idioma inglés.

Como se ha mencionado anteriormente *Packet Filter*, es uno de los mejores *Firewalls* desarrollado en software libre, pero es complicado administrarlo mediante un archivo de configuración, ya que al tener cientos de reglas en un mismo archivo, es difícil mantenerlas legibles y modificarlas fácilmente sin estropear otras.

Existen algunas herramientas que permiten mediante una GUI *(Graphical User Interface)*, crear y editar reglas para el archivo de configuración de “pf”, por ejemplo, “*FirewallBuilder”*, que permite generar archivos de configuración para “pf”.

Estas herramientas son bastante buenas para generar reglas de filtrado, pero el problema es que su interfaz no es tan amigable y sencilla haciendo que su mantenimiento sea un poco menos complicado que el archivo de texto de configuración, pero no soluciona esta desventaja.

Los *Firewalls* comerciales cuentan con una interfaz agradable e intuitiva que permite administrarlos de manera sencilla y eficiente. Esta es una gran diferencia en el momento de decidir que *Firewall* utilizar.

Además de implementar las políticas, los sistemas comerciales permiten obtener información estadística para reportes y optimización del tráfico. Estas características son accesibles cuando se compran los dispositivos adicionales. Siempre y cuando se cuente con una licencia anual para mantener en funcionamiento el *Firewall* comercial.

La propuesta *Desarrollo de un sistema gráfico para la gestión de un Firewall del tipo Packet Filter* permitirá generar las políticas de filtrado de forma más amigable para el usuario final, de forma muy parecida a como se realiza en los sistemas comerciales.

Esta implementación integra varias tecnologías con el fin de obtener un *Firewall* parecido a los de uso comercial, pero basado en software libre.

Una de las ventajas en comparación con otros basados en software libre, es que se encuentra desarrollado totalmente en PHP, el lenguaje de programación para Web más utilizado en el mundo. Esto permitirá mejoras adicionales por terceros aumentando su funcionalidad.

Como resultado se obtendrá un *appliance* desarrollado en software libre, que permita realizar las mismas tareas de los dispositivos comerciales, permitiendo a usuarios sin acceso a estos equipos, practicar en ellos y optar por soluciones de código abierto.

Con el desarrollo de esta propuesta se obtendrá un proyecto de software libre llamado Ixkan, que podrá ser utilizado como *Firewall* basado en *Packet Filter*.

4.1 Arquitectura

La arquitectura de Ixkan es del tipo Cliente – Servidor, las reglas son generadas vía Web y son almacenadas en una base de datos, después de esto se dispara una bandera de cambios, donde un script de consola se encuentra a la espera de estas modificaciones y realizar cambios en la configuración del sistema operativo y en Packet Filter.

Todo el desarrollo se basa en herramientas basadas en software libre, además de incorporar *scripts* de consola para la ejecución de comandos en el sistema, como se observa en la **figura 4-1**.

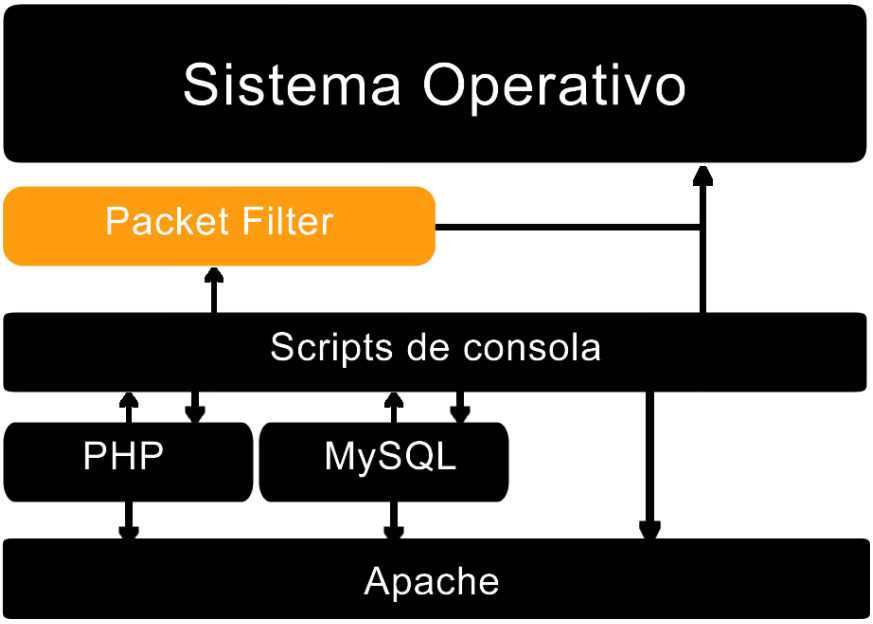


Figura 4-1 Arquitectura para el funcionamiento de Ixkan

Toda la información que será utilizada se encuentra almacenada en la base de datos, MySQL. Ixkan permite incorporar nuevos manejadores de bases de datos agregando los *scripts* en PHP para su funcionamiento.

La interfaz del usuario se encuentra desarrollada totalmente para ser utilizada con múltiples dispositivos y computadoras en diferentes navegadores. Esto se logra mediante la utilización de hojas de estilo CSS y HTML junto con JavaScript y Ajax permitiendo acceder de manera intuitiva a la información.

La parte medular del desarrollo se encuentra en PHP la cual se encarga de la construcción, validación de políticas, generación de formularios de captura y despliegue de información que son desarrollados independientemente del idioma, con la finalidad de agregar nuevos idiomas a futuro.

Finalmente PHP genera un archivo de cambios que es interpretado por un *script* de consola para realizar cambios en el sistema. Este archivo es invocado al arranque y se encuentra en espera de modificaciones para actuar. Sin este archivo las configuraciones y cambios en las reglas de “pf” no se llevarían a cabo.

Una de las ventajas de que su desarrollo sea en PHP es que muchos programadores podrán modificar las funciones e incorporar mejoras y módulos adicionales de acuerdo a sus necesidades.

4.1.1 Funcionamiento

Para que Ixkan funcione adecuadamente cuenta con dos partes fundamentales. La parte *Web* y el *script* de consola Ixkan.sh, **figura 4-2**.

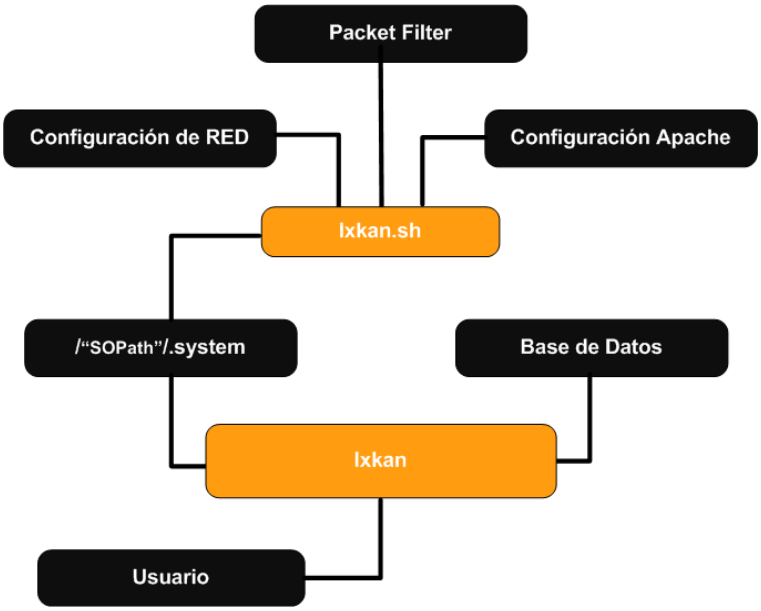


Figura 4-2 Funcionamiento de Ixkan

El usuario interacciona en la parte W*eb* con un navegador de Internet que permita el uso de HTTPS.

El sistema Ixkan almacena la información en la base de datos y las configuraciones del sistema son almacenadas en el archivo "$SOPath/.system".

El *script* de consola Ixkan.sh obtiene los parámetros de modificación del archivo "$SOPath/.system", aplicándolos al sistema operativo, la configuración de Apache, a la interfaces de red, al servidor DNS, generando así reglas de filtrado para "pf" o modificando la forma de funcionamiento del *Firewall* en NAT o transparente.

En las primeras pruebas se intentó que PHP escribiera directamente en los archivos de configuración. El resultado con este mecanismo genero problemas con los permisos de escritura ya que el usuario de Apache no contaba con ellos. Al modificar los permisos de los archivos, este mecanismo funcionaba una sola vez, después OpenBSD encontraba este error o problema de seguridad y lo corregía poniéndole otros permisos a los archivos de configuración.

Una vez estudiado el problema, la solución más óptima fue utilizar un archivo temporal donde se indicaran las modificaciones, esta es la función de la carpeta "/SOPath", la cual puede ser nombrada como se elija siempre que se indique en el archivo "config.php" de Ixkan, lo cual permite que usuarios no deseados puedan obtener información sobre el funcionamiento de Ixkan, al no conocer la ubicación de dicha carpeta. La carpeta contiene información sobre la configuración de "pf", los DNS, *Gateway* e interfaces de red.

4.1.2 Estructura

La estructura de Ixkan es modular, lo que permite hacer modificaciones en las secciones necesarias sin alterar las demás funciones. La **figura 4-3** muestra la estructura de los archivos de Ixkan.

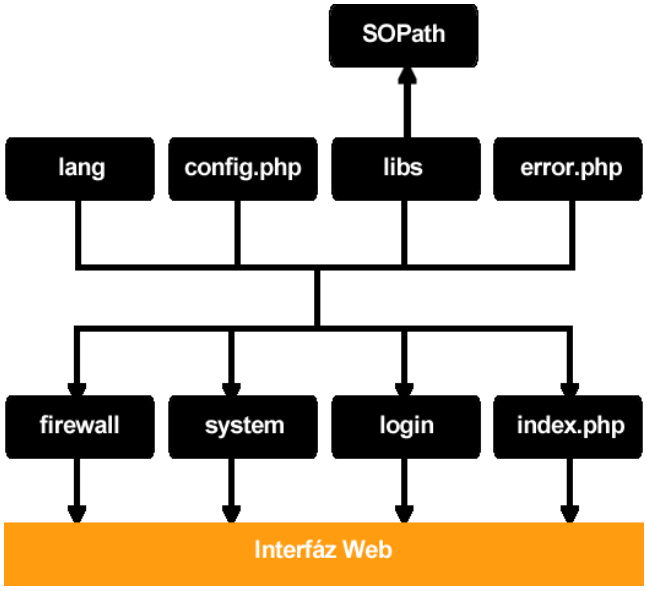


Figura 4-3 Estructura Web de Ixkan

La parte grafica se encuentra en la carpeta “/css” conteniendo las imágenes, logos y hojas de estilo para vestir la aplicación.

El funcionamiento del sistema como las direcciones IP asignadas, el modo de funcionamiento, los usuarios, interfaces, DNS y el acceso al sistema se encuentra en la carpeta “/system”.

Para las acciones que se refieren al funcionamiento del *Firewall* para “pf” se encuentra la carpeta “/firewall”, así como las direcciones, servicios, grupos y redireccionamiento.

Los *scripts* para su funcionamiento en PHP y JavaScript se encuentran en “/libs” lo que permiten buscar y modificar las funciones de manera fácil y sencilla.

Los idiomas son almacenados en la carpeta “/lang” y para su uso se debe modificar la configuración de Ixkan en la variable “$lang”;

La carpeta “/SOPath” contiene los archivos temporales para modificar acciones del sistema operativo, Apache, DNS, interfaces de red y “pf”.

El acceso y cierre de sesión se realiza en “/login” de manera que el acceso a cualquier archivo es primero redirigido a esta sección, también permite prohibir a los usuarios el acceso después de ciertos intentos fallidos en tiempo determinado en el archivo de configuración de Ixkan.

Para evitar el funcionamiento inadecuado al deshabilitar JavaScript en el navegador, se cuenta con “error.php”, que bloquea el acceso cuando JavaScript esta deshabilitado.

La página principal o “index.php” muestra la información del panel de control conformada por los módulos adicionales instalados.

Los módulos adicionales son agregados en la carpeta “/mods” permitiendo agregar y eliminar módulos sin modificar las funciones de Ixkan.

Finalmente el archivo más importante, se trata de la configuración, ”config.php”, donde se indican los datos de conexión a la base, los directorios de archivos necesarios para su funcionamiento, así como el idioma a utilizar. Junto con esto, en

caso de fallas se debe contar con “config.php.tpm” que permite tener un archivo de configuración en caso de que este falle al ser modificado por Ixkan.

La **tabla 4-1** muestra las secciones y su función dentro de Ixkan

Tabla 4-1. Secciones del sistema

Sección	Función
css	Diseño gráfico y hojas de estilos
system	Páginas de despliegue para configuración del sistema
firewall	Páginas de despliegue para configuración del *Firewall*
libs	Scripts que contienen las funciones, formularios para la escritura, obtención y despliegue de información.
lang	Archivos con las cadenas de texto utilizadas en la interfaz
SOPath	Archivos temporales para configuración del sistema operativo y Packet Filter
login	Apertura y cierre de sesión
Index.php	Panel de control, página de inicio
config.php	Archivo de configuración de Ixkan

4.1.3 Módulos adicionales

Los módulos adicionales permiten obtener información y funciones adicionales, como lo hacen los sistemas comerciales, por ejemplo, podemos generar uno que obtenga información y estadísticas del sistema, otro permitiría generar estadísticas de red para monitoreo.

Otras funciones que podrían ser incorporadas serían un generador de informes en HTML y PDF, análisis de datos, balanceo de carga, IDS, IPS, entre otros.

La **figura 4-4** muestra los módulos adicionales con archivos, permitiendo ofrecer características independientes del sistema base Ixkan.

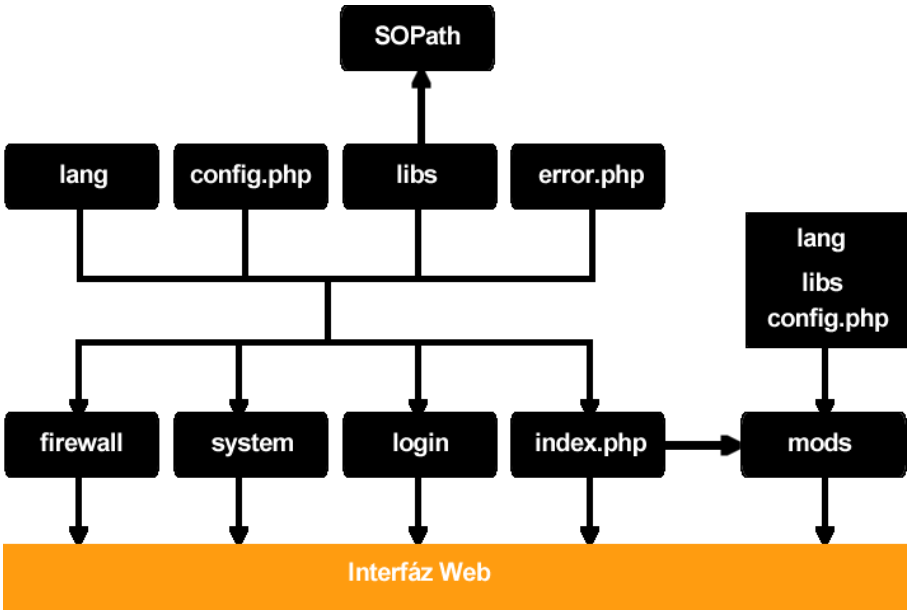


Figura 4-4 Estructura Web de Ixkan con módulos adicionales

4.2 Instalación del Sistema gráfico

Ixkan funciona con *script*s de consola y la interacción con el usuario mediante un sistema *Web*. En esta sección se mostrarán los pasos para la instalación del sistema gráfico y los archivos de configuración y ejecución de Ixkan.

4.2.1 Instalando OAMP

La instalación de un sistema OAMP se refiere al sistema operativo OpenBSD en conjunto con Apache, MySQL y PHP. Por defecto OpenBSD incluye un servidor W*eb* Apache y las librerías para ofrecer un servicio de transmisión segura mediante SSL (OpenBSD, 2014). Este servicio de transmisión se realiza mediante HTTPS, el cual genera un canal de comunicación cifrado entre el usuario y el servidor web,

permitiendo que información sensible como nombres de usuario y contraseñas sean protegidos contra la captura de paquetes de red (Microsoft, 2014).

Para asegurar la comunicación primero se debe generar un certificado, que se almacena en el directorio /etc/ssl/ con la clave correspondiente en /etc/ssl/private/, **figura 4-5**.

```
# openssl genrsa -out /etc/ss1/private/server.key 1024
Generating RSA private key, 1024 bit long Modulus
.......++++++
.....................................................................
....... ........ ++++++
e is 65537 (0x10001)
```

Figura 4-5 server.key

La **figura 4-6** muestra cómo generar un "*certificate signing request*".

```
# openssl req -new -key /etc/ss1/private/server.key -out
/etc/ssl/private/server.csr
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) []:mx
State or Province Name (full name) []:Distrito Federal
Locality Nar-ie (eg, city) []:DF
Organization Name (eg, company) []:ixkan
Organizational Unit Name (eg, section) []:firewall
Common Name (eg, fully qualified host name) []:ixkan
Email Address [ ]: humberto_keymur@ixkan. Org
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:passwodParaFirmarCertificado
An optional company name []:ixkan.org
```

Figura 4-6. Generando server.csr

Para que la conexión sea cifrada Apache debe de contar con un certificado "crt" generado a partir del "crs" y la llave del servidor, **figura 4-7**.

```
#openssl x509 -req -days 365 -in /etc/ssl/private/server.csr -signkey
/etc/ssl/private/server.key -out /etc/ssl/server.crt

Signature ok
subject =/C=mx/ST=Distrito Federal/L=DF/0=ixkan/0U=firewall/CN=ixkan/
emai1Address=humberto_keymur@ixkan.org
Getting Private key
```

Figura 4-7 Obteniendo el server.crt

Para que Apache funcione con HTTPS basta con configurar “/etc/rc.conf” e indicarle que inicie en modo seguro, **figura 4-8**.

```
# use -u to disable chroot, see httpd(8)
#httpd_flags=N0 # for normal use: "" (or "-DSSL" after reading ssl(8))
httpd_f lags = "-DSSL"
```

Figura 4-8 Habilitando apache con HTTPS

Una vez reiniciado el sistema podemos comprobar su funcionamiento en el navegador escribiendo la IP asignada, anteponiendo HTTPS, **figura 4-9** y **figura 4-10**.

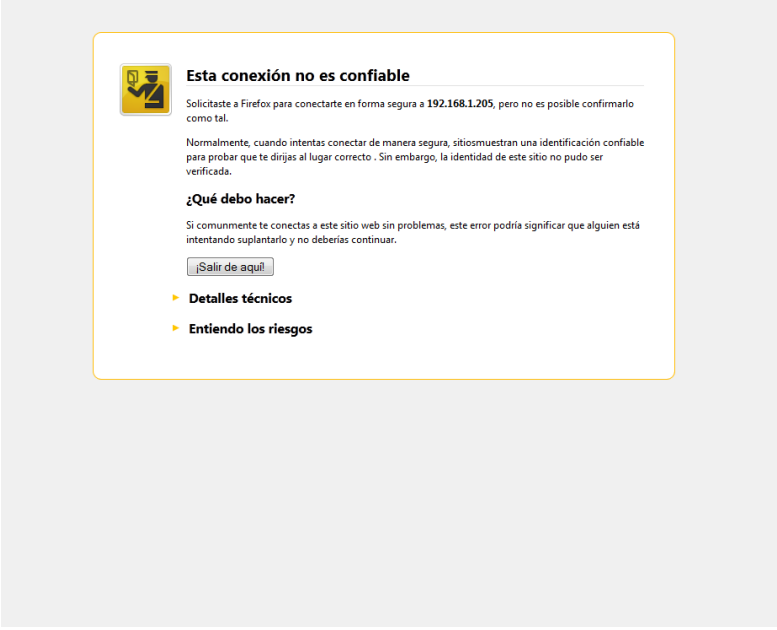


Figura 4-9 Comprobando el acceso

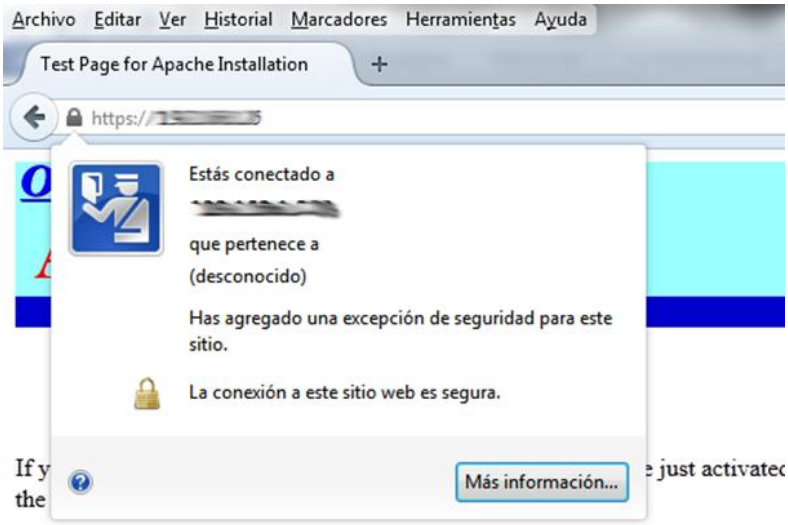


Figura 4-10 Comprobando la conexión segura

Para tener acceso desde Internet es necesario configurar la pasarela y el DNS, **figura 4-11** y **figura 4-12**.

```
#cat /etc/mygate
192.168.1.254
```

Figura 4-11 Agregando la pasarela

```
# cat etc/resolv.conf
Nameserver 8.8.8.8
```

Figura 4-12 Asignando los DNS

En la **figura 4-13** se muestra como al hacer *ping* a un sitio en Internet comprobamos la conexión.

```
# ping google.com
PING google.com (173.194.115.160): 56 data bytes
64 bytes from 173.194.115.160: icmp_seq=0 ttl=58 time=49.740 ms
64 bytes from 173.194.115.160: icmp_seq=1 ttl=58 time=49.154 ms
64 bytes from 173.194.115.160: icmp_seq=2 ttl=58 time=49.444 ms
64 bytes from 173.194.115.160: icmp_seq=3 ttl=58 time=49.132 ms
--- google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 49.132/49.367/49.740/0.332 ms
```

Figura 4-13 Ping a google.com

Una vez configurado Apache, se debe instalar PHP y la base de datos, para realizar esta tarea se tienen que descargar los paquetes desde su página oficial o indicar los orígenes de *software*, **figura 4-14**.

```
# export PKG_PATH=ftp://mirror.esc7.net/pub/OpenBSD/5.5/packages/i386/
```

Figura 4-14 Estableciendo el origen de instalación

Para ejecutar script en PHP, se deben instalar los paquetes y realizar las ligas, como se indican en la **figura 4-15**.

```
# pkg_add libiconv-1.14p1.tgz
# pkg_add libxml-2.9.1.tgz
# pkg_add gettext-0.18.2p4.tgz
# pkg_add png-1.6.8.tgz
# pkg_add jpeg-9p0.tgz
# pkg_add freetype-1.3.1p3.tgz
# pkg_add php-5.3.28p2.tgz
php-5.3.28p2:femail-0.98: ok
php-5.3.28p2:femail-chroot-0.98p2: ok
php-5.3.28p2: ok
Look in /usr/local/share/doc/pkg-readmes for extra documentation.
--- +php-5.3.28p2 -------------------
To enable the php-5.3 module please create a symbolic link from
/var/www/conf/modules.sample/php-5.3.conf to
/var/www/conf/modules/php.conf. As root:

    ln -sf /var/www/conf/modules.sample/php-5.3.conf
/var/www/conf/modules/php.conf

The recommended php configuration has been installed to:
    /etc/php-5.3.ini.
```

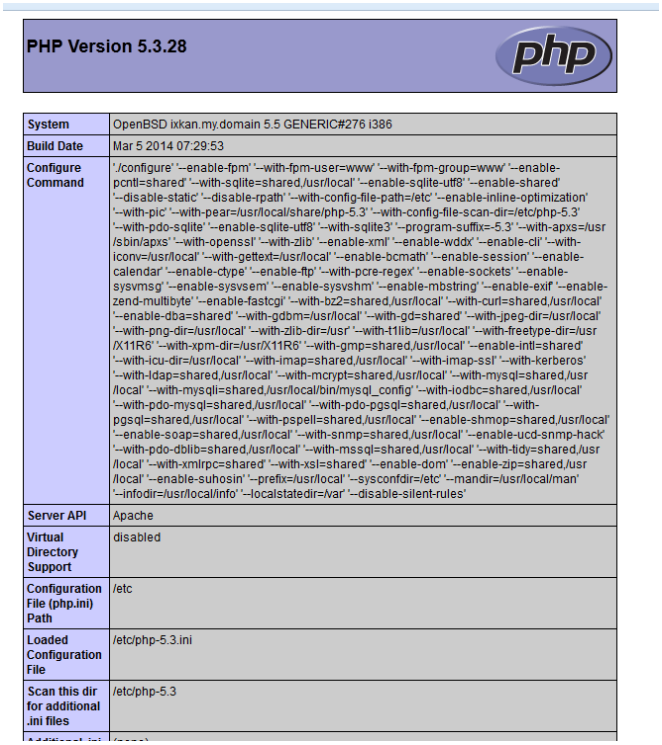
Figura 4-15 Instalando PHP

Para que Apache pueda desplegar contenido PHP hay que indicarlo en su archivo de configuración y reiniciar el servicio, **figura 4-16**.

```
DirectoryIndex index.php index.html
```

Figura 4-16 Agregando el despliegue de contenido PHP

Finalmente para comprobar su funcionamiento, se agregó un archivo de prueba en “/var/www/htdocs/index.php”, **figura 4-17**.



PHP Version 5.3.28

System	OpenBSD ixkan.my.domain 5.5 GENERIC#276 i386
Build Date	Mar 5 2014 07:29:53
Configure Command	'./configure' '--enable-fpm' '--with-fpm-user=www' '--with-fpm-group=www' '--enable-pcntl=shared' '--with-sqlite=shared,/usr/local' '--enable-sqlite-utf8' '--enable-shared' '--disable-static' '--disable-rpath' '--with-config-file-path=/etc' '--enable-inline-optimization' '--with-pic' '--with-pear=/usr/local/share/php-5.3' '--with-config-file-scan-dir=/etc/php-5.3' '--with-pdo-sqlite' '--enable-sqlite-utf8' '--with-sqlite3' '--program-suffix=-5.3' '--with-apxs=/usr/sbin/apxs' '--with-openssl' '--with-zlib' '--enable-xml' '--enable-wddx' '--enable-cli' '--with-iconv=/usr/local' '--with-gettext=/usr/local' '--enable-bcmath' '--enable-session' '--enable-calendar' '--enable-ctype' '--enable-ftp' '--with-pcre-regex' '--enable-sockets' '--enable-sysvmsg' '--enable-sysvsem' '--enable-sysvshm' '--enable-mbstring' '--enable-exif' '--enable-zend-multibyte' '--enable-fastcgi' '--with-bz2=shared,/usr/local' '--with-curl=shared,/usr/local' '--enable-dba=shared' '--with-gdbm=/usr/local' '--with-gd=shared' '--with-jpeg-dir=/usr/local' '--with-png-dir=/usr/local' '--with-zlib-dir=/usr' '--with-t1lib=/usr/local' '--with-freetype-dir=/usr/X11R6' '--with-xpm-dir=/usr/X11R6' '--with-gmp=shared,/usr/local' '--enable-intl=shared' '--with-icu-dir=/usr/local' '--with-imap=shared,/usr/local' '--with-imap-ssl' '--with-kerberos' '--with-ldap=shared,/usr/local' '--with-mcrypt=shared,/usr/local' '--with-mysql=shared,/usr/local' '--with-mysqli=shared,/usr/local/bin/mysql_config' '--with-iodbc=shared,/usr/local' '--with-pdo-mysql=shared,/usr/local' '--with-pdo-pgsql=shared,/usr/local' '--with-pgsql=shared,/usr/local' '--with-pspell=shared,/usr/local' '--enable-shmop=shared,/usr/local' '--enable-soap=shared,/usr/local' '--with-snmp=shared,/usr/local' '--enable-ucd-snmp-hack' '--with-pdo-dblib=shared,/usr/local' '--with-mssql=shared,/usr/local' '--with-tidy=shared,/usr/local' '--with-xmlrpc=shared' '--with-xsl=shared' '--enable-dom' '--enable-zip=shared,/usr/local' '--enable-suhosin' '--prefix=/usr/local' '--sysconfdir=/etc' '--mandir=/usr/local/man' '--infodir=/usr/local/info' '--localstatedir=/var' '--disable-silent-rules'
Server API	Apache
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php-5.3.ini
Scan this dir for additional .ini files	/etc/php-5.3

Figura 4-17 Desplegando phpinfo()

En la base de datos MySQL es necesario instalar los paquetes y realizar las ligas, como se indican en la **figura 4-18**.

```
# pkg_add mysql-client-5.1.73v0.tgz
mysql-client-5.1.73v0: ok
# pkg_add p5-DBD-mysql-4.023p0.tgz
p5-DBD-mysql-4.023p0:p5-Net-Daemon-0.48: ok
p5-DBD-mysql-4.023p0:p5-PlRPC-0.2018p1: ok
p5-DBD-mysql-4.023p0:p5-FreezeThaw-0.5001: ok
p5-DBD-mysql-4.023p0:p5-MLDBM-2.05: ok
p5-DBD-mysql-4.023p0:p5-Params-Util-1.07: ok
p5-DBD-mysql-4.023p0:p5-Clone-0.34: ok
p5-DBD-mysql-4.023p0:p5-SQL-Statement-1.33: ok
p5-DBD-mysql-4.023p0:p5-DBI-1.622: ok
p5-DBD-mysql-4.023p0: ok
# pkg_add mysql-server-5.1.73v0.tgz
mysql-server-5.1.73v0: ok
The following new rcscripts were installed: /etc/rc.d/mysqld
See rc.d(8) for details.
Look in /usr/local/share/doc/pkg-readmes for extra documentation.
```

Figura 4-18 Instalando MySQL

La base de datos MySQL no se inicia por defecto, por lo que hay que iniciarla manualmente y cambiar la contraseña de administrador de la misma, **figura 4-19**.

```
# /usr/local/bin/mysql_install_db
# /usr/local/bin/mysqld_safe &
[1] 17117
# 140511 11:09:04 mysqld_safe Logging to '/var/mysql/ixkan.my.domain.err'.
140511 11:09:04 mysqld_safe Starting mysqld daemon with databases from
/var/mysql
# /usr/local/bin/mysqladmin –u root –p <password>
```

Figura 4-19 Arrancando MySQL

Para que MySQL inicie de manera automática, se debe indicar en el *script* de arranque en /etc/rc.local, **figura 4-20**.

```
# MySQL startup
/usr/local/bin/mysqld_safe --user=_mysql &
```

Figura 4- 20 Agregando script de arranque

Finalmente, para que PHP y MySQL puedan trabajar en conjunto, se debe instalar el conector, **figura 4-21**.

```
# pkg_add php-mysql-5.3.28p0.tgz
php-mysql-5.3.28p0: ok
--- +php-mysql-5.3.28p0 -------------------
You can enable this module by creating a symbolic link from
/etc/php-5.3.sample/mysql.ini to
/etc/php-5.3/mysql.ini. As root:

    ln -sf /etc/php-5.3.sample/mysql.ini /etc/php-5.3/mysql.ini
#  ln -sf /etc/php-5.3.sample/mysql.ini /etc/php-5.3/mysql.ini
# apachectl restart
/usr/sbin/apachectl restart: httpd restarted
```

Figura 4-21 Instalando el conector

Hasta este punto, el sistema operativo cuenta con los elementos necesarios para el funcionamiento de Ixkan.

4.2.2 Instalando y configurando el Sistema Base

Una vez que se cuenta con los elementos necesarios para el funcionamiento de Ixkan, sólo falta seguir los pasos que se indican en el instalador que se proporciona, install.sh, en el paquete Ixkan.tar.gz. El instalador solicita parámetros para la instalación como nombre y contraseña de la base de datos, carpeta para alojar los archivos /SOPath, la carpeta y dirección IP para el acceso *Web* entre otras cosas.

4.2.2.1 Archivo de configuración

Una vez terminado el instalador, se genera un archivo llamado config.php. Este *script* contiene:

- Los parámetros para la conexión a la base de datos
- Las rutas de acceso *Web* y el "SOPath"
- Las variables de versión e idioma
- Parámetros para mitigar acceso denegados
- El número de políticas por página para ser visualizadas

En la **tabla 4-1** de la **sección 4.1.2**, se muestran las secciones y archivos generados junto con el archivo de configuración

El archivo puede ser modificado por el usuario, para indicarle el puerto de acceso en HTTPS, como se describe en la **sección 4.5.2** (configuración de red),

4.2.2.2 Archivos para el SO

La ruta SOPath contiene los archivos temporales que permiten al *script* de ejecución tomar los valores y aplicarlos al sistema. Posteriormente, se limpian los valores temporales.

Algunos de los elementos temporales que se almacenan son:

- La acción a realizar .system
- Las reglas de filtrado en pf.conf
- El DNS para Web, en resolv.conf
- La Gateway en mygate
- Las interfaces de red en nics.conf

4.2.3 *Scripts* de ejecución

El *script* de ejecución Ixkan.sh realiza todas las modificaciones y ajustes necesarios, realizando lo que se le pide en la parte *Web,* ejecutando acciones para llevar a cabo estas tareas utilizando comandos del sistema.

Es necesario agregar este *script* al arranque del sistema, al igual que se hace con MySQL.

4.3 Entrar y salir del sistema

El acceso a Ixkan se realiza haciendo una petición HTTPS a la dirección IP o dominio asignado durante el proceso de instalación. Es recomendable configurar Ixkan con una IP local, esto nos permitirá modificar la configuración y políticas de filtrado de paquetes desde la red local sin exponerlo a una red pública, **figura 4-22** y la **figura 4-23**.

Figura 4-22 Acceso desde una computadora

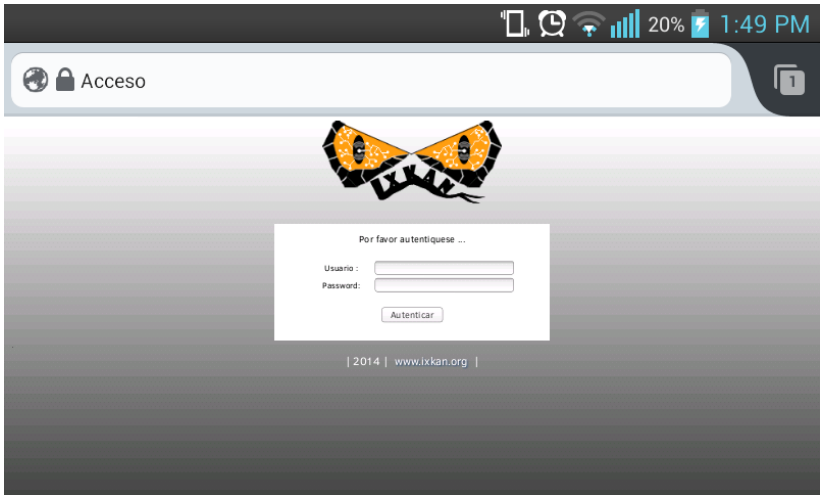


Figura 4-23 Acceso desde un dispositivo móvil

La configuración puede ser modificada directamente en Ixkan en la sección de acceso, como se describe en la **sección 4.5.2**.

4.4 Panel de control

Una vez autenticado en el sistema, se redirecciona al panel de control, donde se muestra de manera rápida los detalles del sistema y la información de los módulos adicionales instalados. La **figura 4-24** muestra el mapa del sitio, con los archivos que podemos visualizar en la interfaz Web.

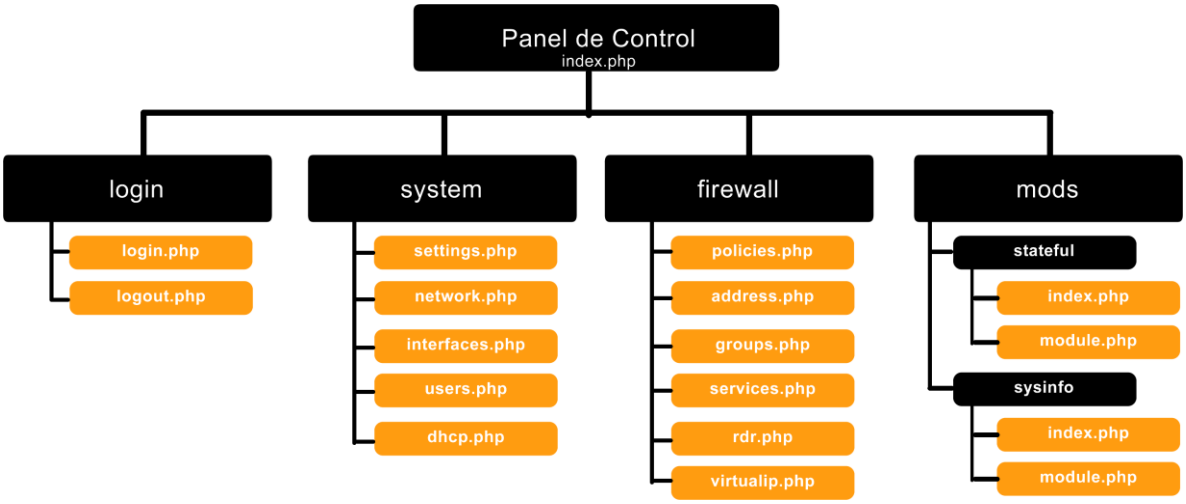


Figura 4-24 Mapa de sitio

4.5 Sistema

En la sección de Sistema se configuran los parámetros que hacen relación con el sistema operativo, como son las interfaces de red, direcciones IP, usuarios, DNS, DHCP y el modo de operación.

Brevemente, esta sección no realiza nada que tenga relación con acciones que deban realizarse en un *Firewall*.

La **figura 4-25** muestra los archivos que podemos visualizar en la interfaz Web para la sección *system*.

```
./ixkan/system/
./ixkan/system/settings.php
./ixkan/system/dhcp.php
./ixkan/system/interfaces.php
./ixkan/system/network.php
./ixkan/system/users.php
```

Figura 4-25 Sección system

4.5.1 Modo de operación

La **figura 4-26** despliega la pantalla de selección de modo de operación, indicando la descripción del modo actual que puede ser transparente o NAT.

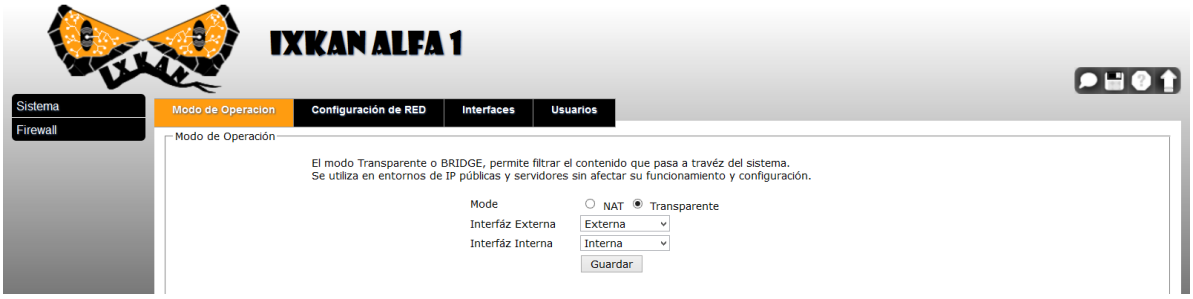


Figura 4-26 Modo de operación

El modo de operación transparente utiliza dos interfaces de red a manera de puente, de tal forma que “pf” puede bloquear o permitir el paso de tráfico, sin modificar la configuración de los demás equipos.

El modo NAT funciona como un *router,* sirviendo de pasarela de paquetes de red, donde “pf” puede permitir o bloquear el tráfico, normalmente esta función se utiliza para filtrar una red interna.

4.5.2 Configuración de red

La configuración de red, **figura 4-27**, establece los parámetros para el acceso, DNS asignados y en caso de que el modo de operación sea NAT se establece la pasarela de red LAN.

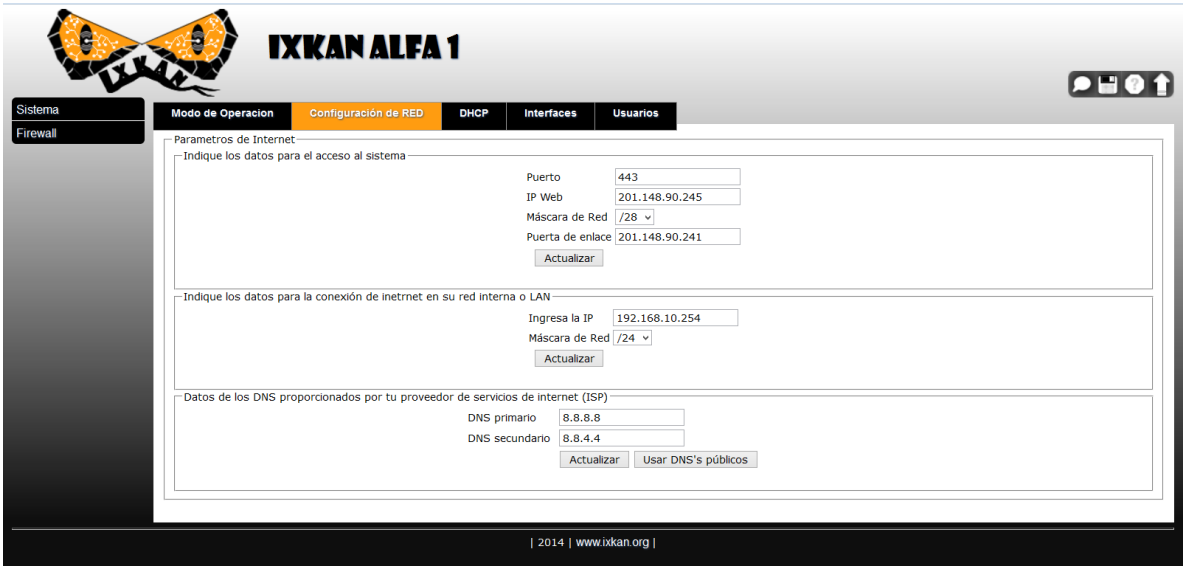


Figura 4-27 Configuración de red

4.5.3 DHCP

La **figura 4-28** muestra la pantalla donde se indican los parámetros para una conexión de red de manera automática o DHCP. Es posible indicar si se desea habilitar o deshabilitar esta opción.

Figura 4-28 DHCP

4.5.4 Interfaces

La **figura 4-29** muestra la pantalla en donde se indican todas las interfaces de red disponibles para el funcionamiento de Ixkan. Si una interfaz no se encuentra en la lista es posible agregar nuevas presionando el botón Agregar.

Figura 4-29 Interfaces de red

4.5.5 Usuarios

La **figura 4-30**, muestra la pantalla con la lista de usuarios y permisos de acceso al sistema.

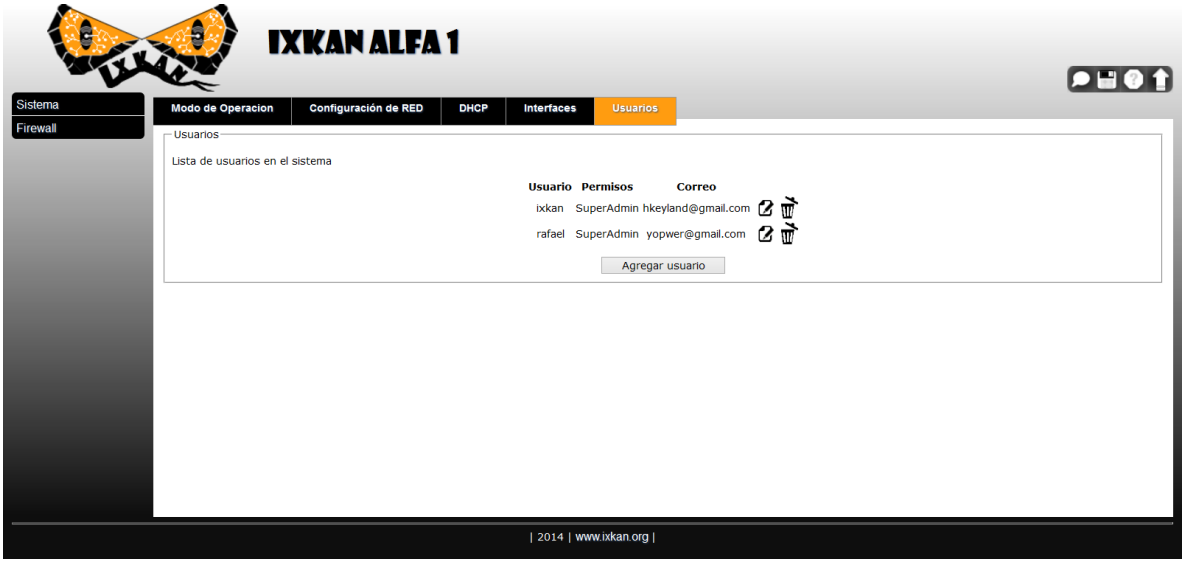


Figura 4-30 Usuarios en el sistema

El rol del usuario Admin únicamente es capaz de visualizar la información, no puede realizar cambios en el sistema, mientras que SuperAdmin puede realizar cambios en el sistema.

4.6 *Firewall*

En la sección de *Firewall*, se configuran los parámetros de direcciones de red, servicios o protocolos, redirecciones, asignaciones direccionales de IP a puertos y las políticas de filtrado.

La **figura 4-31** muestra los archivos que podemos visualizar en la interfaz Web para la sección *Firewall*.

```
./ixkan/firewall/
./ixkan/firewall/policies.php
./ixkan/firewall/address.php
./ixkan/firewall/groups.php
./ixkan/firewall/rdr.php
./ixkan/firewall/services.php
./ixkan/firewall/virtualip.php
```

Figura 4-31 Sección firewall

4.6.1 Políticas

La **figura 4-32** muestra la lista de políticas de filtrado de manera resumida, mostrando el flujo de la conexión y si esta se bloquea o se permite.

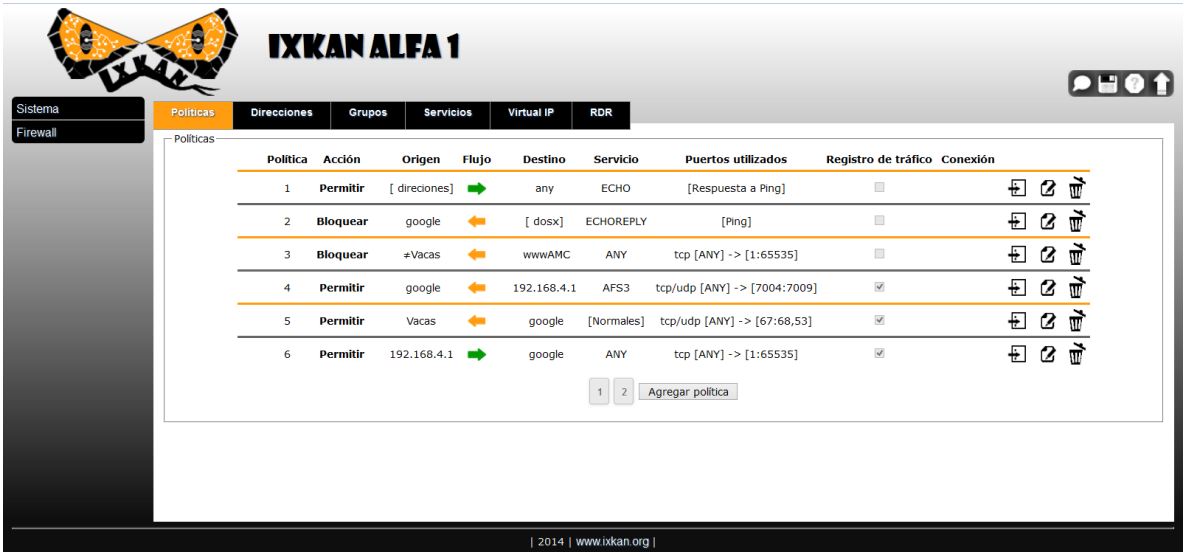


Figura 4- 32 Políticas de filtrado

Las políticas son ordenadas de acuerdo a la prioridad y su orden de aplicación. Es posible modificarlas, mover su orden y eliminarlas.

En la parte final se muestra una lista con las páginas de políticas y el botón para agregar nuevas.

4.6.2 Direcciones

La **figura 4-33** muestra la lista de direcciones IP o dominio, las cuales permiten tener una dirección de origen y destino en el formulario de generación de políticas.

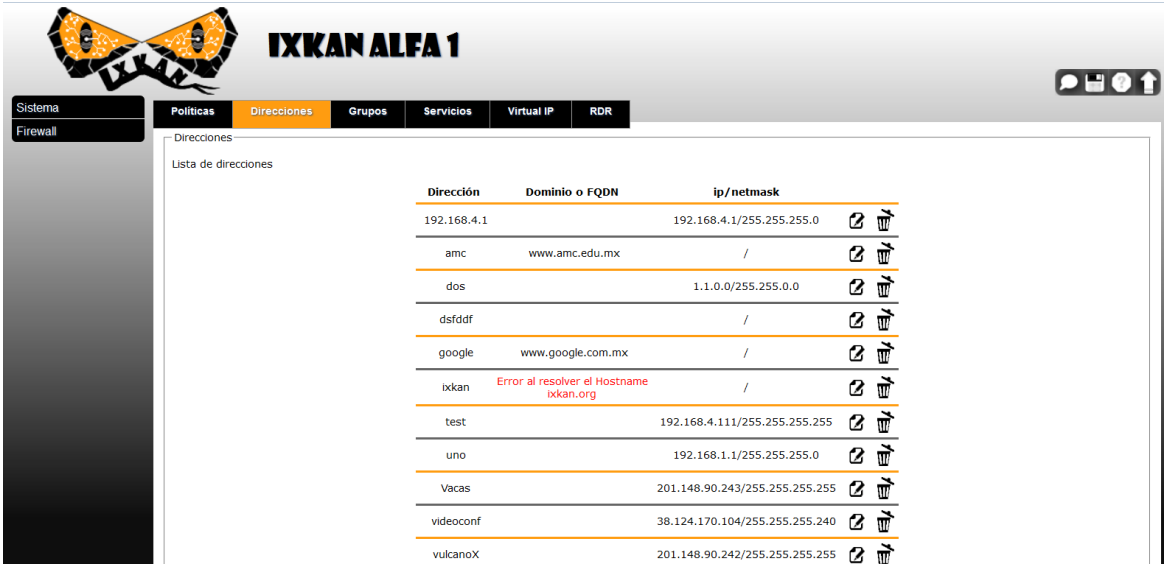


Figura 4-33 Direcciones

Si alguna de ellas lanza un error, se debe revisar la esta dirección, ya que no es posible obtener la dirección IP del dominio.

Las direcciones son ordenadas alfabéticamente y es posible modificarlas y eliminarlas. Al final de estas se encuentra el botón para agregar nuevas.

4.6.3 Grupos

La **figura 4-34** muestra la lista de grupos formados para una mejor administración del *Firewall*. Esto permite tener una lista de direcciones IP o de servicios y protocolos para hacer más fácil la creación de reglas y aumentar la velocidad de búsqueda y aplicación por parte de “pf”.

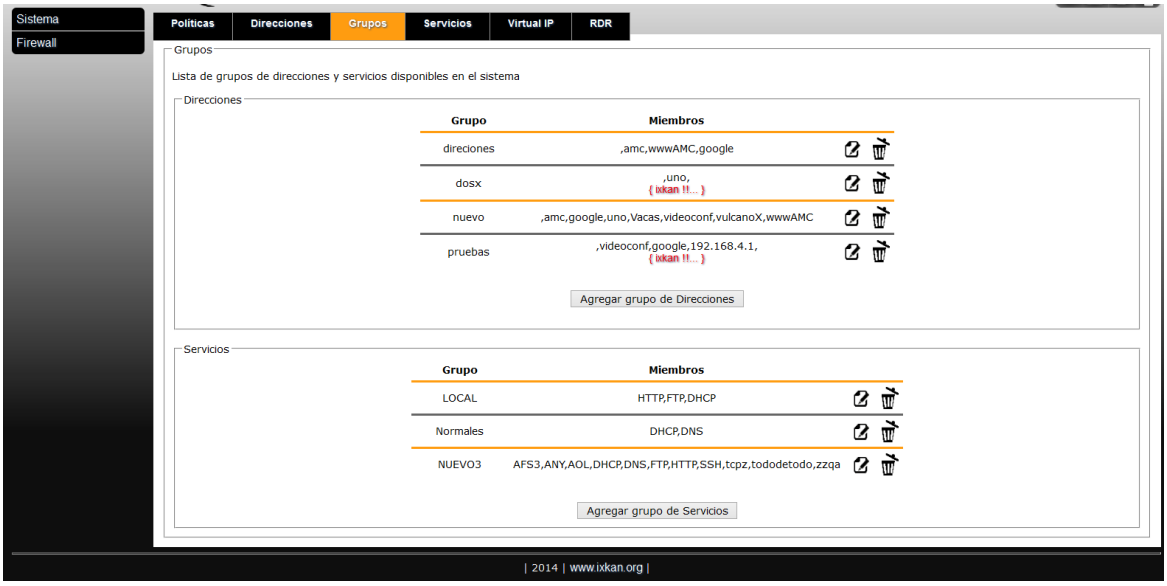


Figura 4- 34 Grupos

La creación de grupos se realiza mediante un formulario, **figura 4-35**, de manera intuitiva y fácil.

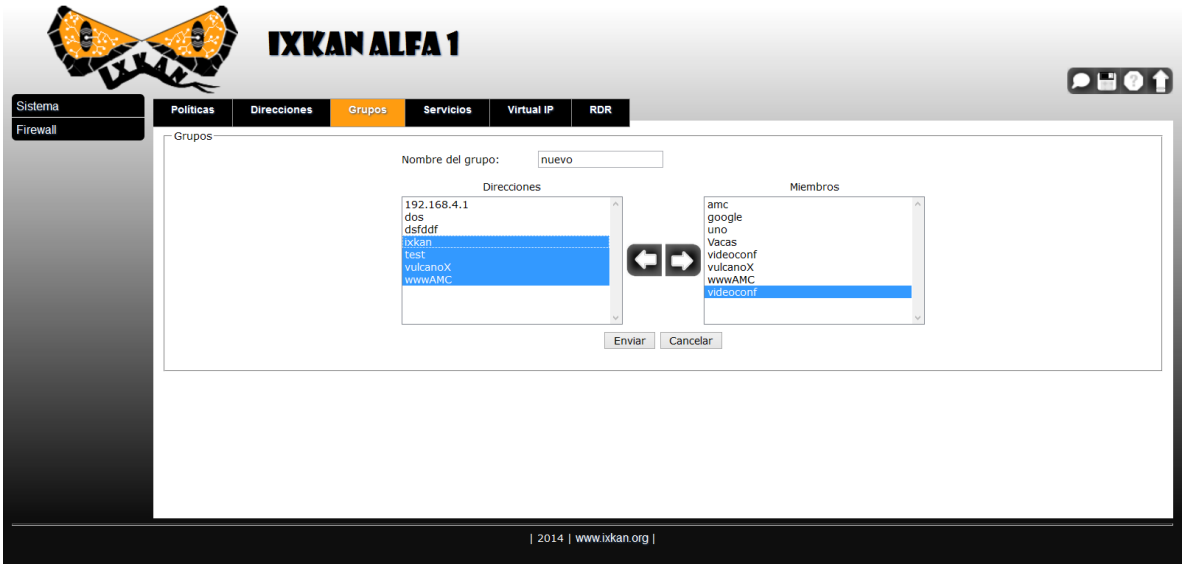


Figura 4-35 Creación de grupos

4.6.4 Servicios

La **figura 4-36** muestra la lista de servicios y protocolos ya sean TCP, UDP o ICMP, indicando el puerto origen, destino o un rango.

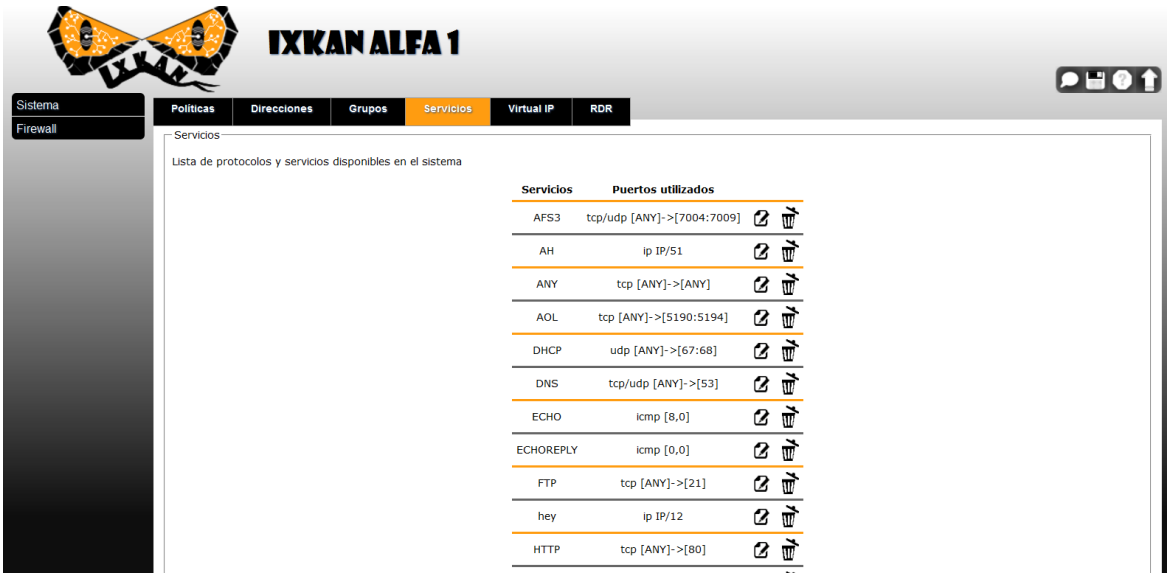


Figura 4-36 Servicios en el sistema

La creación de servicios permite elegir los parámetros, dependiendo del tipo de protocolo a utilizar, **figura 4-37.**

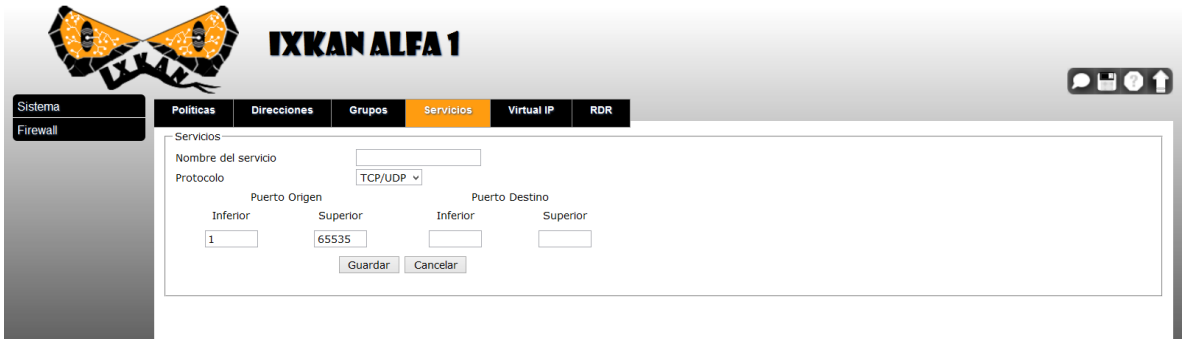


Figura 4-37 Nuevo servicio TCP/UDP

4.6.5 Virtual IP

Algunos equipos comerciales incluyen esta función que permite asignar una IP pública a una IP interna de la LAN, únicamente en modo NAT.

La **figura 4-38** indica la asignación de IP pública a IP interna, de forma que los usuarios desde Internet pueden acceder a este equipo, aunque se encuentre en una red LAN.

Figura 4-38 Virtual IP

Esta función es útil en entornos de pruebas o cuando el servicio no es dedicado, permitiendo acceder a la red interna mediante esta IP virtual.

4.6.6 Redireccionamiento

En algunas ocasiones es necesario redirigir el tráfico de un puerto a un equipo en especial. La **figura 4-39** indica el redireccionamiento de un puerto a una IP local, siempre que el sistema se encuentre en modo NAT.

Figura 4-39 Redireccionamiento

Las pruebas realizadas al proyecto se realizaron para verificar su funcionamiento, eliminar la presencia de errores en la implementación así como los problemas de sintaxis que resultaran al generar las reglas de filtrado para Packet Filter y la configuración del sistema operativo.

En la mayoría de desarrollos es común encontrar errores, por lo que las pruebas de funcionamiento y usabilidad son muy importantes para cumplir con las características y especificaciones que se describieron desde el inicio del proyecto.

Durante el desarrollo del proyecto se realizaron las pruebas de las diferentes secciones y módulos del mismo. Obteniendo la retroalimentación, posibles errores y solventándolos desde su comienzo.

Una vez terminado el desarrollo se solicitó a usuarios que conocían de manera superficial el uso y administración de equipos de redes, que realizaran pruebas del mismo con el fin de encontrar errores y realizar recomendaciones sobre la navegación, funcionalidad y de la información mostrada por la interfaz.

Las pruebas tanto del desarrollo como las hechas por los usuarios mostraron problemas de resolución de DNS, enlaces a sitios inexistentes, errores en las cadenas de lenguaje utilizadas, al insertar datos inexistentes o mal formados en los formularios, falta de retroalimentación al completar formularios, duplicidad de valores, mal funcionamiento al generar reglas, acceso no autorizado al deshabilitar JavaScript, entre otros.

Las pruebas de rendimiento con el equipo donde se instaló Ixkan, mostraron que el uso de MySQL como base de datos agrega una carga de alrededor de 43 megabytes y que Apache agrega por cada usuario que utiliza la interfaz un aproximado de 3 megabytes. Los datos de la RAM en uso en conjunto con el sistema operativo fueron realizadas durante varios días mostrando que la carga real del sistema es de alrededor de 500 MB. Este valor nos indica el mínimo de RAM necesario para utilizar Ixkan en OpenBSD.

Ya que Ixkan no requiere el uso de CPU de manera excesiva, los datos arrojados del uso del procesador nos indicaban que se utilizaba del 3 al 5% de manera aproximada durante toda la prueba.

Cabe mencionar que las pruebas realizadas se denominan de caja negra (Rodriguez Tello, 2011), ya que si bien se verifico el código fuente al ser desarrollado se obtuvo más información de errores y modificaciones por parte de los usuarios sin acceso al código fuente.

Debido a que esta es una primera versión es posible que se encuentren errores que serán solventados en versiones posteriores.

4.7.1 Pruebas de integración

Las pruebas en la integración de los elementos mostraron durante las pruebas que todas las librerías son necesarias para el funcionamiento del sistema. La parte más esencial se encuentra en la base de datos ya que sin esta las modificaciones no pueden ser almacenadas y no se permite el acceso a los usuarios.

Durante la integración se realizaron pruebas sobre la redundancia del sistema agregando elementos que almacenan la configuración antes de realizar modificaciones y si estas no son equivocadas se restablecen los valores anteriores.

Un ejemplo de esta prueba es la asignación de una IP fuera de nuestro segmento y deseamos configurársela para el acceso en Ixkan. Al detectar que esta IP no puede ser establecida restablece la IP anterior.

4.7.2 Pruebas de usabilidad

Las pruebas realizadas por usuarios se llevaron a cabo en diferentes equipos y dispositivos con resoluciones de pantalla distintos. Esta prueba mostró que es posible utilizar la interfaz desde dispositivos móviles y que la interacción no se ve afectada en pantallas táctiles.

Las pruebas de navegación mostraron que existían varias secciones pequeñas y comunes que se podían juntar y mostrarlas todas en una sola pantalla, permitiendo un mejor entendimiento de la información de red.

Otro punto a mencionar se refiere a la validación de formularios que no se vio alterado al utilizar los diferentes equipos y dispositivos.

4.7.2 Pruebas de funcionamiento

El equipo fue probado para funcionar en modo NAT y transparente, revisando que la configuración del sistema y los archivos de configuración se generaran de manera exitosa. Este punto requiero solventar el problema de permisos al entre los usuarios del sistema operativo, ya que Apache no puede escribir archivos que son propiedad de root. Para realizar esta tarea se realizó un shellscript ejecutado por root, que toma la información de apache y realiza las modificaciones con todos los permisos.

Los equipos utilizados comercialmente no permiten elegir la política por defecto y al realizar las pruebas con archivos de configuración de Packet Filter, la política por defecto era permisiva. Mucha de la documentación consultada no mencionaba la utilización con una política restrictiva, lo que requirió probar las reglas, revisar los registros del sistema, la tabla de conexiones y estados arrojada por Packet Filter. Finalmente se realizaron pequeños ajustes al código que permite modificar el resultado del archivo de configuración de Packet Filter, obteniendo políticas restrictivas y permisivas.

Conclusiones

Se desarrolló un sistema de gestión gráfico para un *Firewall* del tipo *Packet Filter* denominado Ixkan.

Esta herramienta gráfica vía Web permite gestionar la construcción de políticas y reglas de filtrado de paquetes en un *Firewall* de modo transparente o NAT en el sistema operativo OpenBSD con Packet Filter.

Ixkan es una herramienta desarrollada por completo con software libre, cuenta con características como son la necesidad imperante de seguridad en todo aquel sitio u organización que esté conectada a Internet y la de no invertir cantidades de dinero onerosas para su adquisición; estas características muy particularmente le permitirán competir con sistemas comerciales, siempre y cuando el código se actualice y fortalezca a través de la participación de aquellos entusiastas que decidan unirse al proyecto.

Al ser liberado como “software libre” significa que cualquier usuario tiene la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

- La libertad de ejecutar el programa para cualquier propósito.
- La libertad de estudiar cómo funciona el programa y cambiarlo para que haga lo que se desee.
- La libertad de redistribuir copias para ayudar a otros.

- La libertad de distribuir copias de sus versiones modificadas a terceros

Cualquier institución u organización, podrá utilizar esta alternativa, sin necesidad de conocer a detalle las implicaciones que intervienen en su implementación y administración ya que lo intuitivo y amigable del sistema lo asiste en todo momento. De igual forma, los administradores de sistemas, especialistas en seguridad, administradores de red, los estudiantes y en general cualquier persona interesada, podrá acceder a un sistema con las características esenciales para habilitar un mecanismo robusto de seguridad como es un Firewall.

Aquellas organizaciones que tengan implementado Packet Filter como *Firewall* podrán gestionar sus reglas de manera más cómoda, entendible y eficiente al tener una interfaz gráfica sencilla e intuitiva.

De manera particular quiero enfatizar que este desarrollo puede apoyar las actividades académicas desarrolladas en los laboratorios de las materias de redes y seguridad implementadas actualmente o que vengan a futuro, ya que durante su desarrollo y posterior implementación me ha permitido comprender de manera sistematizada y práctica algunos de los temas vistos en las materias ya señaladas.

Po otro lado la utilización de Ixkan en entornos de producción ha permitido mitigar ataques de denegación de servicio y de fuerza bruta contra los servidores y demás equipos dentro del perímetro, en los lugares donde se implementó se pudo bloquear el acceso a sitios y direcciones no permitidas durante horarios laborales y escolares. Algunos ejemplos de estos sitios son: redes sociales, sitios de descargas, chats, videos, entre otros, como lo solicitaba la política organizacional donde se probó.

El sistema desarrollado al ubicarlo correctamente provee del primer mecanismo de seguridad para la organización, tiene la capacidad de sustituir paulatinamente siempre que se haga de manera controlada y previa capacitación, equipos comerciales que tengan funciones similares. Los resultados positivos de su aplicación serán inmediatos, refiriéndonos a costos de implementación, mantenimiento y actualización y por supuesto al nivel de seguridad ofrecido ya que se tendrá protección en las capas 2, 3, 4 y excepcionalmente la capa 7 del modelo OSI.

Recomendaciones

El trabajo aquí no ha concluido, las funcionalidades y características que se pueden incorporar son muchísimas. Lo aprendido sobre el desarrollo de aplicaciones, configuración y el funcionamiento de OpenBSD con “Packet Filter” es enorme y depende de las necesidades de cada administrador.

Este desarrollo liberado como software libre tendrá actualizaciones futuras en las que me gustaría implementar mecanismos de IPS, con SNORT, análisis de tráfico mediante graficas de uso de red, balanceo de carga y priorización de tráfico, acceso WIFI, entre otras muchas.

Ya que el desarrollo permite incorporar módulos, en esta primera versión se incluyen los módulos de filtrado por estados e información del sistema, que permite a otros desarrolladores observar cómo crear nuevos módulos e incorporarlos al sistema base o encapsularlos de manera independiente sin modificar el código fuente.

Apéndice A

En el apéndice se mencionan algunos de los archivos más importantes para el funcionamiento de Ixkan. El código fuente y la documentación completa se encuentra en http://www.ixkan.org.mx .

Archivo de configuración: config.php

```
<?php
/**
*Ixkan, the eye that everything looks
* configuration file.
*
* @package Ixkan.config
* @author Humberto Keymur Landeros <hkeyland@gmail.com> @hkeymur
* @since Versión 1.0, revisión 2014-05
* @version beta 1.0
*/

/*Database*/
$dbName='ixkan';
$dbType='mysql';
//Some host need this
//$dbHost='localhost';
$dbHost='127.0.0.1';
$dbUser='ixkan';
$dbPassword='ixkanpssword';

/*Paths*/
$SOPath='/var/www/htdocs/ixkan/'so_Path /';
$SystemPath='so_Path/';
```

```
$www = "https://IP_adm:443/ixkan";

/*Vars*/
$systemVersion='Ixkan FW';
$homepage='www.ixkan.org';
$systemBuild='5-10-2014';
$license='GPL';
$site = "Ixkan";
$lang = 'es';

/*Login banned*/
$numErrors='3';
$minutesBanned='5';

/*Paginator*/
$elementsPerPages='6';

/*Default policy*/
/*block or pass*/
$configDefaultPolicy='block';

?>
```

Base de datos: mysql.php

```
<?php
/**
* Ixkan, the eye that everything looks
* All functions to manage Mysql.
*
*
* @package Ixkan.Mysql
* @author Humberto Keymur Landeros <hkeyland@gmail.com> @hkeymur
* @since Versión 1.0, revisión 2014-01
* @version alfa 1.0
*/
conect();
function conect(){
	global $dbName;
	global $dbType;
	global $dbHost;
	global $dbName;
	global $dbUser;
	global $dbPassword;
	$conect=mysql_connect($dbHost,$dbUser,$dbPassword) or die ("Error
conectando a la base de datos");
	mysql_select_db($dbName,$conect);
	return $conect;
}
...
...

?>
```

Paquete de Idioma : lang/es.php

```
<?php
/*This file contains all strings used at the interface*/
/*Este archivo contiene todas las cadenas utilizadas en la interfáz*/
$title_index='Ixkan, el ojo que todo lo ve';
$title_login='Acceso';
$title_normal='Ixkan';
$strRDR='RDR';
$strRDRLong='Reenvío de puertos';
$strRDRAdd='Agregar RDR';
$strGroupFormText='Lista de grupos de direcciones y servicios disponibles en el
sistema';
$strHelp='Ayuda';
$strSupport='Soporte';
$strSaveConfig='Guardar configuración';
$strExit='Salir';
...
...

?>
```

Habilitar JavaScript : error.php

```
<?php
require_once('config.php');
require_once("./libs/$dbType.php");
require_once("./lang/$lang.php");
require_once("./libs/php.php");
require_once("./libs/forms.php");
?>
<html>
<head>
<title><?php echo $title_index;?></title>
<?php
	meta();
	JQuery();
	icon();
	css('style');
?>
	<script type="text/javascript" charset="utf-8">
	$(function(){
		<?php
		//menuJQ();
		?>
	});
	</script>
</head>
<body>
<?php
headerDiv();
titleDiv();
systemDiv();
//left_Menu();
?>
<div id="columna_central">
<div id="main">
<fieldset>
```

```
<legend><?php echo $legendError ?></legend>
<?php

echo $activateJavascript;

jsButton('index.php','Ok','    '.$srtReturn.'    ');
?>
</fieldset>
</div>
  </div>
<?php footer(); ?>
</body>
</html>
```

Hoja de estylos : css/style.css

```
body{
	background-color: #0e0e0e;
	background-image:url(bg.png);
	background-repeat:no-repeat;
	background-repeat:repeat-x;
	margin:10px;
	margin-top:0px;
}
div{
	margin:0px;
}
img{
	border:none;
}
#header{
	height:100px;
}
.JSbutton{
	margin-top:10px;
}
#logo{
	margin-top:5px;
	margin-left:40px;
	}
#system{
	position:absolute;
	height:50px;
	left:300px;
	top: 25px;
	/*font-family:Croobie;*/
	font-family:Showcard Gothic;
	color:#000;
	font-size:35px;
}
#system a{
	font-family:Showcard Gothic;
	color:#000;
	font-size:35px;
}
```

Lista de tablas y figuras

Bibliografía y Mesografía

Microsoft. (20 de 10 de 2014). Securing Web Communications: Certificates, SSL, and https://. Obtenido de Microsoft /Web: http://www.microsoft.com/web/post/securing-web-communications-certificates-ssl-and-https

Carvajal, A. (2008). Introducción a la inseguridad de la información. *Revista Hakin9*.

Carvajal, A. (2008). Introducción a la inseguridad de la información. *Revista Hakin9*, 31-35.

Chapin, D., & Akridge, S. (2005). How Can Security Be Measured? *Information Systems Control Journal, ISACA*. Obtenido de http://www.isaca.org/Journal/Past-Issues/2005/Volume-2/Pages/How-Can-Security-Be-Measured.aspx

CISCO. (2011). *http://www.cisco.com.* Recuperado el 4 de 5 de 2014, de http://www.cisco.com: http://www.cisco.com/c/dam/en/us/products/collateral/security/content-security-management-appliance/ironport_anti_malware_datasheet.pdf

Cole, E., Krutz, R., & Conley, J. (2005). *Network Security Bible.* Indianapolis, Indiana: Wiley Publishing, Inc.

Expresión Binaria. (22 de 08 de 2013). *http://www.expresionbinaria.com/*. Obtenido de Xplico framework forense para el análisis de tráfico de red: http://www.expresionbinaria.com/xplico-framework-forense-para-el-analisis-de-trafico-de-red/

García Alfaro, J., & Perramón Tornil, X. (2004). *Aspectos avanzados en seguridad en redes.* Barcelona, España: Eureca Media, SL.

Gómez Rodríguez, F. (2008). Open Source Security Information Management. *Hakin9*, 18-19.

Gómez Vieites, Á. (2011). *Enciclopedia de la seguridad informática, 2ª edición actualizada.* México D.F.: Alfaomega Grupo Editor.

Hansteen, P. (2008). *The book of pf.* San Francisco, CA: No Starch Press, Inc.

insecure.org. (6 de 9 de 2003). *http://insecure.org*. Obtenido de Las 75 Herramientas de Seguridad Más Usadas: http://insecure.org/tools/tools-es.html

INTECO. (2011). *Instituto Nacional de Tecnologías de la Comunicación .* Obtenido de http://www.inteco.es: http://www.inteco.es/extfrontinteco/img/File/demostrador/monografico_catalogo_seguridad_perimetral.pdf

ISECOM. (2010). *OSSTMM 3 – The Open Source Security Testing Methodology Manual.* New York, USA: ISECOM.

iso27000.es. (25 de 4 de 2013). *El portal de ISO 27001 en Español*. Obtenido de El portal de ISO 27001 en Español: http://www.iso27000.es/sgsi.html#section2d

Ixkan. (10 de 6 de 2014). *ixkan.org*. Obtenido de ixkan.org: http://www.ixkan.org

López Barrientos, M., & Quezada Reyes, C. (2006). *Fundamentos de seguridad informática.* México D.F.: Facultad de Ingeniería, Universidad Nacional Autónoma de México.

Mañas, J. (20013). *Guía de seguridad (CCN-STIC-401) Glosario y abreviaturas.* Madrid, España: Centro Criptológico Nacional.

Merola, A. (2006). Funcionamiento de los Sistemas de Detección de Intrusiones. *Hakin9*, 26-28.

Microsoft. (17 de 3 de 2011). *http://www.microsoft.com*. Obtenido de TechNet Security: Herramienta de Evaluación de Seguridad de Microsoft (MSAT): http://technet.microsoft.com/es-es/security/cc185712

Mieres , J. (1 de 2009). *www.evilfingers.com.* Recuperado el 15 de 11 de 2013, de EvilFingers: https://www.evilfingers.com/publications/white_AR/01_Ataques_informaticos.pdf

Nidecki, T. (2006). GFI LANguard Network Security Scanner. *Hakin9*, 13.

OpenBSD. (2014). *OpenBSD.org*. Recuperado el 10 de 5 de 2014, de http://www.openbsd.org: http://www.openbsd.org/faq/faq10.html#HTTPS

Oppleman, V. (2008). Network Defense. *Hakin9*, 14-25.

OWASP. (5 de 4 de 2014). *www.owasp.org*. Obtenido de www.owasp.org: https://www.owasp.org/index.php/Vulnerability

Pacheco, F., & Jara, H. (2012). *Ethical Hacking 2.0.* Buenos Aires, Argentina: Fox Andina S.A.

Phoha, V. V. (2002). *Internet Security Dictionary.* New York, NY, USA: Springer-Verlag New York, Inc.

Piotrowski, M. (2006). Un sistema IPS a base de Snort. *Hakin9*, 48-53.

Piper, S. (2011). *Intrusion Prevention Systems FOR DUMMIES.* Indianapolis, Indiana: Wiley Publishing, Inc.

PTES. (30 de 4 de 2012). *Penetration Testing Execution Standard, PTES*. Recuperado el 26 de 4 de 2014, de Penetration Testing Execution Standard, PTES: http://www.pentest-standard.org/index.php/Main_Page

PTES-Reporting. (30 de 9 de 2011). *Penetration Testing Execution Standard*. Recuperado el 1 de 5 de 2014, de Penetration Testing Execution Standard: http://www.pentest-standard.org/index.php/Reporting

Ramos Fraile, A. (2 de 2011). *http://www.intypedia.com/.* Obtenido de Enciclopedia de la Seguridad de la Información: http://www.criptored.upm.es/intypedia/docs/es/video5/DiapositivasIntypedia005.pdf

Real Academia Española. (11 de 2 de 2014). *Real Academia Española*. Recuperado el 11 de 2 de 2014, de Real Academia Española: http://lema.rae.es/drae/srv/search?id=wwcs1Hw7LDXX28FVsWTl

Rodriguez Tello, E. (11 de 3 de 2011). *Laboratorio de Tecnologías de Información del Cinvestav Tamaulipas.* Obtenido de Laboratorio de Tecnologías de Información del Cinvestav Tamaulipas: http://www.tamps.cinvestav.mx/~ertello/swe/swTestingTecZacatecas.pdf

SHOPOS. (1 de 2014). *http://www.sophos.com*. Recuperado el 4 de 5 de 2014, de http://www.sophos.com: http://www.sophos.com/en-us/products/your-needs/technology-case-studies/stop-spam.aspx

Snort. (12 de 5 de 2014). *snort.org*. Obtenido de snort.org: http://www.snort.org/

SOPHOS. (30 de 10 de 2013). *SOPHOS*. Recuperado el 4 de 5 de 2014, de SOPHOS: http://www.sophos.com/en-us/products.aspx

Taboada Gómez, E. (2005). *http://aui.es/.* Obtenido de Asociación de Usuarios de Internet: http://aui.es/IMG/pdf/spam.pdf